Pengamanan File Video dengan Enkripsi dan Deskripsi Menggunakan Algoritma Salsa20 dan RSA

I Putu Herdy Juniawan[a1], I Komang Ari Mogi[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]juniawan.2208561033@student.unud.ac.id
[2]arimogi@unud.ac.id

Abstract

In the face of escalating security challenges in the advanced digital era, the exchange and storage of video files have become integral to daily life, impacting everything from business communication to personal entertainment. While the internet facilitates easy access to multimedia content, it also brings heightened risks of data theft and privacy breaches. Cryptography offers a solution by encrypting sensitive data. This research employs two encryption algorithms, Salsa20 and RSA, to protect video files. Salsa20 ensures fast and secure encryption, maintaining data integrity and confidentiality. RSA facilitates secure key exchange, limiting access to authorized recipients. The combined use of these algorithms effectively secures video files, guarding against data theft and privacy breaches. This offline encryption system offers flexible accessibility without compromising security. Additionally, it preserves the original file extension, adding an extra layer of protection, while decryption enables seamless access and playback of video content.

Keywords: *Video Encryption, Salsa20, RSA, Data Security, Cryptography.*

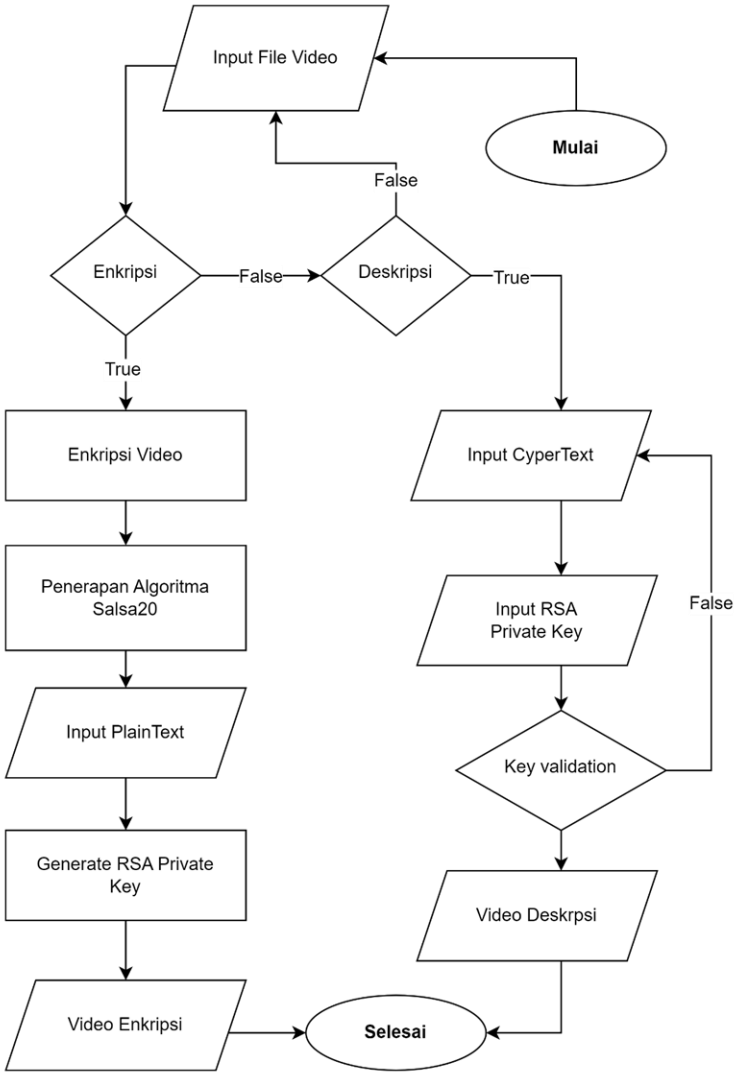
1. Pendahuluan

Dalam menghadapi tantangan keamanan dalam era digital yang semakin maju, Penting untuk menyadari bahwa berbagi dan menyimpan file video kini menjadi bagian tak terpisahkan dari kehidupan sehari-hari, mempengaruhi berbagai aspek dari komunikasi bisnis hingga hiburan pribadi. Seiring dengan kemajuan teknologi, internet telah memfasilitasi akses yang lebih cepat dan mudah terhadap konten multimedia, namun, di sisi lain, meningkatnya konektivitas ini juga menimbulkan risiko yang tak terhindarkan terkait dengan pencurian data dan pelanggaran privasi[1]. Kriptografi telah muncul sebagai solusi yang ampuh untuk menjaga data sensitif agar tidak diakses oleh pihak yang tidak berwenang. Dengan menggunakan teknik-teknik kriptografi yang canggih, informasi yang disimpan dalam format video dapat dienkripsi, membuatnya sulit untuk diakses oleh pihak yang tidak berwenang[2]. Enkripsi mengacu pada proses mengubah data ke dalam format yang tidak dapat dibaca tanpa memiliki kunci enkripsi yang cocok., sedangkan deskripsi adalah proses mengembalikan data ke keadaan aslinya menggunakan kunci dekripsi yang tepat. Selain itu, dengan meningkatnya kesadaran akan keamanan informasi, implementasi kriptografi dalam pertukaran dan penyimpanan file video telah menjadi semakin penting. Dalam jurnal ini peneliti menerapkan dua algoritma enkripsi yaitu Salsa20 dan RSA. Algoritma Salsa20 adalah algoritma enkripsi arus yang terkenal karena kecepatan dan keamanannya. Dengan menggunakan Salsa20, peneliti dapat mengenkripsi file video secara efisien dan efektif, menjaga integritas dan kerahasiaan data. Namun, untuk mengamankan proses pertukaran kunci enkripsi Salsa20, peneliti juga menerapkan algoritma kunci publik RSA sebagai enkripsi tambahan. RSA menyediakan mekanisme yang aman untuk pertukaran kunci, memastikan bahwa hanya penerima yang dimaksud yang dapat mendapatkan akses ke kunci enkripsi yang diperlukan[3].

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

2. Metode Penelitian

2.1. Desain Sistem



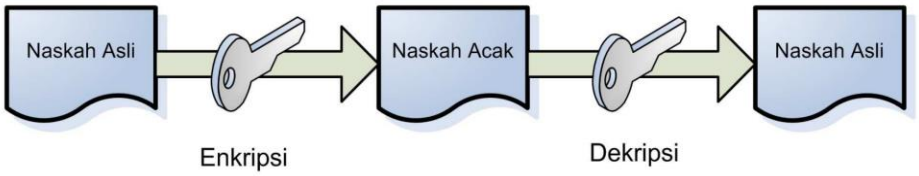
Gambar 1. *Flowchart* Sistem

Untuk alur cara kerja *Flowchart* Sistem tahapan dimulai dengan memasukkan file video yang akan dienkripsi atau didekripsi. Pengguna kemudian memilih apakah ingin melakukan enkripsi atau dekripsi. Jika opsi enkripsi dipilih, proses dilanjutkan dengan penggunaan algoritma *Salsa20* untuk mengenkripsi video tersebut. Selain itu, pengguna diminta untuk memasukkan *plaintext* yang nantinya akan dijadikan sebagai kunci deskripsi juga, untuk memastikan keamanan tambahan. Setelah itu, kunci privat RSA dihasilkan untuk keamanan tambahan. Hasil akhir dari proses ini adalah video yang terenkripsi dengan aman, sehingga terhindar dari akses tanpa izin. Di sisi lain, jika opsi dekripsi dipilih, pengguna diminta memasukkan *Ciphertext* (teks terenkripsi) dan kunci privat RSA. Validasi kunci dilakukan, dan jika benar, proses lanjut ke tahap dekripsi. Video kemudian didekripsi, dan hasilnya adalah video yang terdekripsi sepenuhnya, siap untuk ditonton kembali. Dalam kesimpulan, *flowchart* ini memberikan gambaran yang jelas mengenai langkah-langkah yang harus diikuti untuk mengamankan sebuah video melalui enkripsi serta mengembalikan video tersebut ke bentuk semula melalui dekripsi.

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

2.2. Kriptografi

Kriptografi adalah teknik-teknik pengamanan data, Informasi diubah menggunakan suatu kunci enkripsi sehingga sulit untuk dibaca oleh pihak yang tidak memiliki kunci dekripsi yang sesuai. Proses dekripsi kemudian menggunakan kunci tersebut untuk mengembalikan data ke bentuk aslinya[4]. Sementara itu, asal-usul kata "kriptografi" berasal dari bahasa Yunani, menggabungkan kata "cryptos" yang berarti "secret" (tersembunyi) dan "graphein" yang berarti "writing" (penelitian). Oleh karena itu, kriptografi dapat diartikan sebagai "secret writing" (penelitian rahasia) [4].



Gambar 2. *Kriptografi*

Gambar diatas menggambarkan hasil dari proses enkripsi dan dekripsi. Secara umum, proses enkripsi melibatkan pengacakan teks asli (plaintext) menjadi teks teracak (ciphertext) yang sulit dibaca bagi siapa pun yang tidak memiliki kunci dekripsi [5].

2.3. Algoritma Salsa20

Algoritma Salsa20 adalah sebuah algoritma kriptografi simetris yang dikembangkan oleh Daniel J. Bernstein pada tahun 2005. Algoritma ini digunakan untuk melakukan enkripsi dan dekripsi data dengan kecepatan tinggi serta tingkat keamanan yang tinggi. Langkah-langkah utama dalam algoritma Salsa20 melibatkan inisialisasi, pembangkitan kunci, XOR, iterasi, dan output.

Tabel 1. *Initial State Salsa20*

Key	Cons	Nonce	Nonce
Pos	Pos	Cons	Key
Key	Key	Key	Cons

$$b \oplus= (a + d) <<< 7; \quad (1)$$

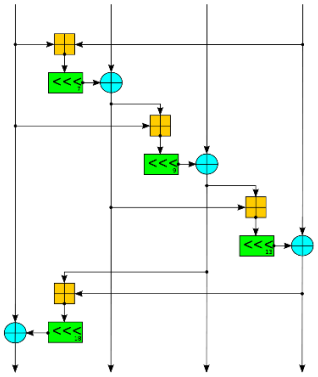
$$c \oplus= (b + a) <<< 9; \quad (2)$$

$$d \oplus= (c + b) <<< 13; \quad (3)$$

$$a \oplus= (d + c) <<< 18; \quad (4)$$

Algoritma Salsa20 merupakan salah satu kandidat dalam proyek eSTREAM. Algoritma ini menerima empat masukan, yaitu 8-byte block counter, 8-byte nonce (nomor pesan unik) yang biasanya berupa angka acak atau angka pseudorandom, dan 32 byte kunci. Keempat masukan untuk Algoritma Salsa20 diatur seperti yang ditunjukkan pada Tabel 1. Setiap kotak matriks dalam Gambar 1 memiliki ukuran 4 byte. Struktur ini dikenal sebagai initial state. Setiap nilai indeks dari matriks dalam initial state diubah menjadi bentuk little endian[6].

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

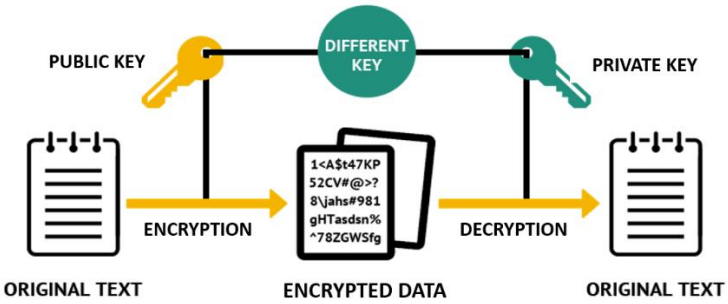


Gambar 3. Fungsi *quarter rounds*

Langkah selanjutnya setelah proses konversi ke bentuk little endian adalah pembentukan keystream. Dalam proses ini, algoritma Salsa20 melakukan total 20 putaran, di mana setiap putaran melibatkan fungsi quarter round. Persamaan 1-4 diterapkan pada setiap putaran dengan perbedaan: pada putaran ganjil, fungsi quarter round hanya memproses bagian kolom dari struktur, sementara pada putaran genap, fungsi quarter round memproses bagian baris. Hasil dari putaran terakhir ditambahkan dengan keadaan awal dalam bentuk little endian. Kemudian, hasil operasi sebelumnya disimpan dalam bentuk little endian sebagai keystream [6]. Keystream adalah komponen kunci krusial dalam proses enkripsi dan dekripsi algoritma Salsa20, diperlukan untuk operasi XOR dengan teks plaintext atau ciphertext. Setelah inisialisasi dengan kunci dan vektor inisialisasi, algoritma menghasilkan aliran kunci yang sesuai dengan panjang data yang akan dienkripsi atau didekripsi. Data dienkripsi dengan cara di-XOR-kan dengan aliran kunci yang dihasilkan [6]. Algoritma Salsa20 melakukan serangkaian iterasi menggunakan fungsi operasi dan permutasi untuk menghasilkan keystream. Setelah iterasi selesai, aliran kunci yang dihasilkan digunakan untuk melakukan enkripsi atau dekripsi data. Keamanan algoritma Salsa20 bergantung pada kekuatan kunci yang digunakan dan ketangguhan fungsi operasi dan permutasi di setiap putaran.

2.4. Algoritma RSA (Rivest Shamir Adleman)

Algoritma RSA adalah sebuah algoritma kriptografi kunci publik yang dikembangkan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1978. Dalam algoritma ini, pesan dienkripsi menggunakan kunci publik, dan hanya dapat didekripsi dengan kunci privat yang sesuai.. RSA didasarkan pada kesulitan dalam memecahkan faktorisasi dari bilangan-bilangan besar. Algoritma ini memiliki berbagai aplikasi, termasuk dalam pengamanan komunikasi digital dan penandatanganan digital[7].



Gambar 4. Algoritma RSA

Proses pembuatan sepasang kunci RSA (Rivest Shamir Adleman) melibatkan langkah-langkah sebagai berikut:

a. Inisialisasi:
- Pilih dua bilangan prima acak, sebagai contoh P = 11 dan q = 19.
- Hitung nilai n: $n = p \times q$n = p × q, sehingga 11 × 19 = 20911 × 19 = 209.
- Hitung nilai phi (n): $\phi(n) = (p-1) \times (q-1)$ ϕ (n) = (p−1) × (q−1), sehingga 10 × 18 = 18010 × 18 = 180.

b. Kunci Publik:
Pilih bilangan acak e yang relatif prima dengan phi(n), misalnya e = 7.

c. Kunci Pribadi:
Temukan bilangan d yang memenuhi syarat $d \times e \equiv 1 \pmod{\phi(n)}$.
Gunakan algoritma extended Euclidean untuk mencari d:

1. Inisialisasi: phi(n) = 180, e = 7.
2. Langkah 1: 180 = 7 \times 25 + 5.
3. Langkah 2: 7 = 5 \times 1 + 2.
4. Langkah 3: 5 = 2 \times 2 + 1.
5. Langkah 4: Terapkan substitusi balik: 1 = 5 – 2 × 2 = (180 – 7 × 25) × 2 = 175 × 7 − 180 × 251 = 5 – 2 × 2 = (180 – 7 × 25) × 2 = 175 × 7 – 180 × 25.
6. Langkah 5: Hasilkan d dengan memodulo phi(n): d = 175 mod 180 = 175d = 175 mod 180 = 175.

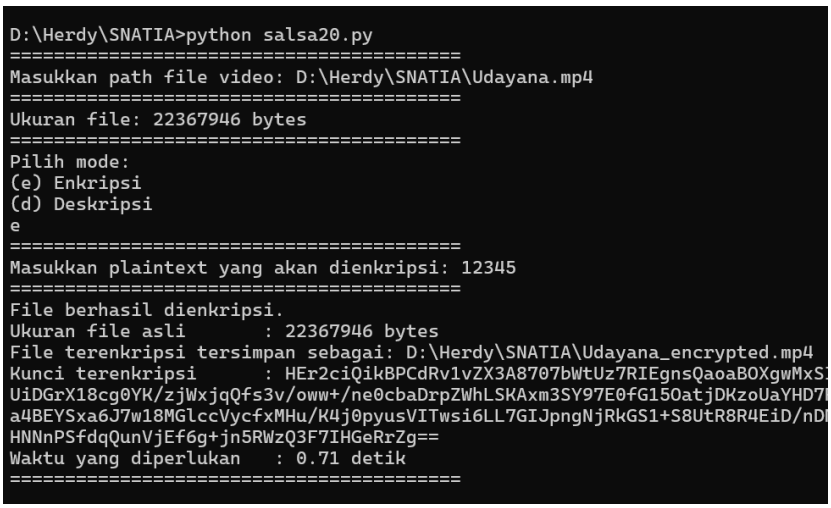
Hasil dari proses ini adalah kunci publik (n, e) = (209, 7) dan kunci pribadi (n, d) = (209, 175). Ini menjelaskan langkah-langkah awal dalam pembuatan sepasang kunci RSA, termasuk inisialisasi, pemilihan kunci publik, dan perhitungan kunci pribadi[2].

3. Hasil dan Diskusi

3.1. Uji Coba Sistem

a. Proses Enkripsi Video

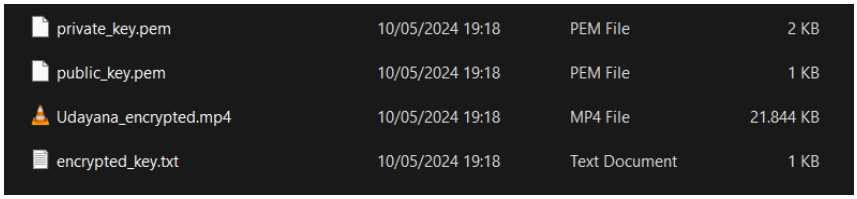
Proses enkripsi video ini bertujuan untuk mengevaluasi kemampuan sistem dalam mengenkripsi file video menggunakan algoritma Salsa20, dengan tambahan enkripsi RSA. Tahapan-tahapan yang dilakukan adalah sebagai berikut:



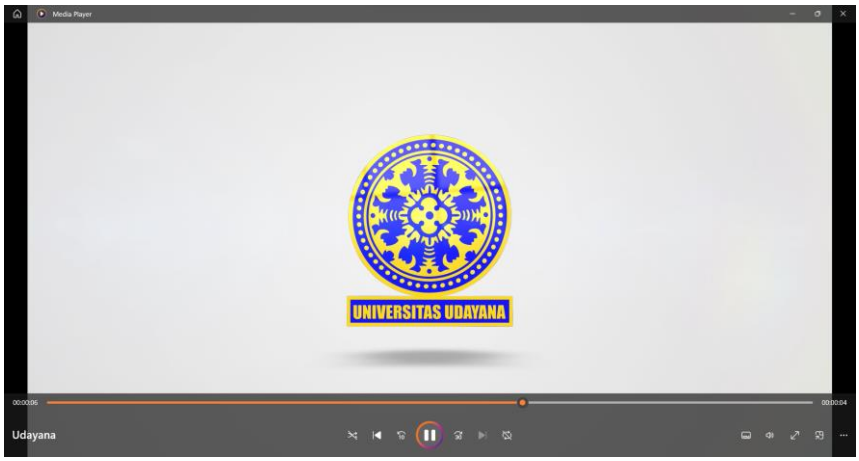
Gambar 5. Proses Enkripsi Video

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

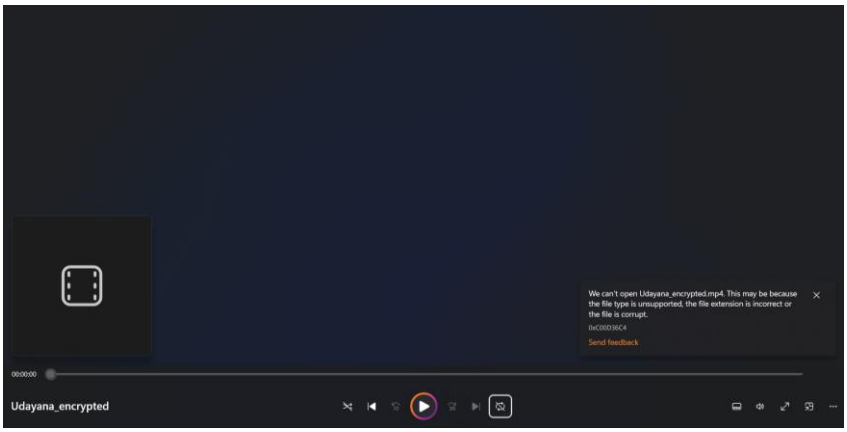
- **Persiapan File Video:** File video dengan format .mp4 dipersiapkan untuk proses enkripsi.
- **Input Path Video:** Pengguna diminta untuk memasukkan lokasi file video yang akan dienkripsi ke dalam program.
- **Enkripsi dengan Salsa20:** Program akan menginisiasi proses enkripsi video menggunakan algoritma Salsa20.
- **Pemasukan Plaintext:** Pengguna diminta untuk memasukkan plaintext yang akan digunakan untuk mendekripsi file video nanti.
- **Generasi RSA Key:** Program akan menghasilkan kunci RSA dalam bentuk public_key.pem dan private_key.pem, yang dapat digunakan untuk proses dekripsi video.
- **Penyelesaian Proses Enkripsi:** Setelah proses enkripsi selesai, program akan secara otomatis membuat file video dengan tambahan awalan 'encrypted_' pada nama file aslinya.



Gambar 6. Hasil Enkripsi Video



Gambar 7. Video Sebelum Enkripsi



Gambar 7. Video Sesusah Enkripsi

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

b. Proses Enkripsi Video

Berikut adalah deskripsi untuk setiap tahapan dalam proses deskripsi file video:

- **Persiapan File Video:** Tahap ini melibatkan pemilihan file video yang akan dideskripsi. File video dengan format .mp4 dipersiapkan untuk proses deskripsi.
- **Input Path Video:** Pengguna diminta untuk menyediakan path atau lokasi dari file video yang akan dideskripsi. Path ini akan digunakan oleh program untuk mengakses file video tersebut.
- **Input CyperText:** Pengguna akan diminta untuk menginputkan ciphertext yang diperoleh dari proses enkripsi sebelumnya. Ciphertext ini adalah hasil dari proses enkripsi menggunakan algoritma Salsa20 dan RSA.
- **Input Path RSA Private Key:** Pengguna juga diminta untuk memasukkan path atau lokasi dari file Private Key yang digunakan dalam proses enkripsi sebelumnya. Private Key ini diperlukan untuk mendekripsi ciphertext menjadi plaintext.
- **Validasi Kunci:** Program akan melakukan validasi terhadap kedua kunci yang dimasukkan oleh pengguna, yaitu ciphertext dan RSA Private Key. Jika keduanya sesuai, program akan melanjutkan ke tahap selanjutnya.
- **Penyelesaian Proses Deskripsi:** Setelah proses validasi kunci selesai, program akan melakukan proses deskripsi terhadap file video menggunakan kunci yang telah divalidasi. Hasil deskripsi akan disimpan dengan nama file baru yang memiliki awalan "decrypted_".

Gambar 8. Hasil Deskripsi Video

Gambar 9. Video Sesudah Deskripsi

3.2. Hasil Uji Coba Sistem

Tabel 1. Hasil Pengujian Deskripsi Video

No	Bulir Pengujian	Output yang diinginkan	Output yang keluar	Keterangan
1	Membaca data video	Sistem dapat membaca data video dari file explorer.	Sistem berhasil membaca data video dari file explorer.	Sesuai
2	*Plaintext* ke Ciphertext	Sistem dapat membaca *plaintext* dan mengubahnya dalam bentuk *ciphertext*.	Sistem berhasil membaca *plaintext* dan mengubahnya dalam bentuk *ciphertext.*	Sesuai

No	Bulir Pengujian	Output yang diinginkan	Output yang keluar	Keterangan
3	Generate key	Sistem mampu menghasilkan sepasang kunci *RSA* yaitu public key dan *private key* .	Sistem berhasil mendapatkan sepasang kunci *RSA* yaitu public key dan *private key*.	Sesuai
4	Enkripsi data video	Sistem mampu mengenkripsi data video sesuai algoritma *RSA(Rivest Shamir Adleman)* dan *Salsa20*	Sistem berhasil mengenkripsi data video sesuai algoritma *RSA(Rivest Shamir Adleman)* dan *Salsa20*.	Sesuai
5	Dekripsi data video	Sistem dapat mendeteksi data video yang telah terenkripsi untuk mendekripsikan kembali menggunakan *private key* dan *ciphertext*	Sistem berhasil mendeteksi data video yang telah terenkripsi untuk mendekripsikan kembali menggunakan *private key* dan *ciphertext*	Sesuai
6	Deteksi kesalahan sistem	Sistem dapat mendeteksi kesalahan pengguna seperti pengguna tidak memasukkan file video dan kesalahan pada kunci deskripsi	Sistem berhasil mendeteksi kesalahan pengguna seperti pengguna tidak memasukkan file video dan kesalahan pada kunci deskripsi	Sesuai

4. Kesimpulan

Berdasarkan pada penelitian dan pembahasan diatas, peneliti menyimpulkan bahwa penerapan Enkripsi dan Deskripsi pada File Video menggunakan Algoritma Salsa20 dan RSA adalah solusi yang efektif dalam menjaga kerahasiaan dan integritas data video, serta melindunginya dari ancaman pencurian data dan pelanggaran privasi. Dalam program ini, kombinasi metode enkripsi Salsa20 dan RSA telah terbukti efisien, menghasilkan lapisan keamanan yang kuat bagi data video. Metode RSA memberikan tambahan keamanan tingkat tinggi sementara Salsa20 mempercepat proses enkripsi secara signifikan. Sistem Enkripsi dan Dekripsi ini mampu beroperasi secara offline, memastikan aksesibilitas yang fleksibel tanpa mengorbankan keamanan. Proses enkripsi dalam sistem ini mampu mengamankan konten data video tanpa mengubah ekstensi aslinya, sehingga menghasilkan lapisan perlindungan yang lebih dalam. Sedangkan, proses dekripsi mempertahankan ekstensi asli dan mengembalikan isi data video ke bentuk semula, memungkinkan data video tersebut dapat diakses dan diputar kembali dengan lancar.

Daftar Pustaka

[1] F. Zuli and A. Irawan, “Implementasi Kriptografi Dengan Algoritma Blowfish Dan Riverst Shamir Adleman (RSA) Untuk Proteksi File,” vol. 9, no. 1, pp. 5–13, 2017, doi: 10.22441/fifo.v9i1.2568.

[2] Yusmaifany, Tommy, and R. Siregar, “Aplikasi Enkripsi Data Video Menggunakan Metode Rsa Dan Blowfish Berbasis Web,” vol. 2, no. 3, pp. 2024–535, 2024.

[3] A. Ramadhan, A. Kusyanti, and P. H. Trisnawan, “Implementasi Algoritme Enkripsi Salsa20 untuk Pengamanan Data Video Surveilans secara Real-Time,” vol. 5, no. 2, pp. 477–484, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[4] M. Qamal, “Kriptografi File Citra Menggunakan Algoritma Tea (Tiny Encryption Algorithm),” vol. 5, no. 2, pp. 11–43, 2014.

[5] S. Kromodimoeljo, *Teori dan Aplikasi Kriptografi*. 2009.

[6] M. Thareq, P. Beyri, A. Kusyanti, and F. A. Bakhtiar, “Implementasi Algoritme Salsa20 untuk Pengamanan Search Keyword Dokumen Terenkripsi,” vol. 4, no. 10, pp. 3531–3541, 2020, [Online]. Available: http://j-ptiik.ub.ac.id

[7] U. Indriani, O. Alfina, and N. Syahputri, “Journal of Machine Learning and Data Analytics (MALDA) Penerapan Algoritma RSA Dalam Keamanan File Ms Word,” vol. 1, no. 2, pp. 95–100, 2022.

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

Halaman ini sengaja dibiarkan kosong