3rd Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
30 Agustus 2023 – Jakarta, Indonesia
Volume 2, Nomor 2, September 2023 - ISSN 2962-8628 (*online*)

PENERAPAN ALGORITME KRIPTOGRAFI SHA-256 DAN AES-256 UNTUK PENGAMANAN FILE PADA PT PELANGI SENTRAL KREASI

Arief Dharmawan[1], Haris Munandar[2]

[1,2]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budiluhur, Jakarta, Indonesia
Email: [1]ariefdharmawan2014@gmail.com, [2]haris.munandar@budiluhur.ac.id

Abstrak- PT Pelangi Sentral Kreasi adalah perusahaan Indonesia yang bergerak di bidang industri kreatif. Didirikan pada tahun 2005 yang berfokus pada pengembangan dan produksi produk-produk kreatif yang inovatif dan berkualitas tinggi. Saat ini penanganan data penawaran *project* yang dilakukan oleh PT Pelangi Sentral Kreasi tergolong belum aman karena data masih dalam bentuk *cipherfile* yang dimana masih bisa dibaca oleh semua orang yang memiliki file. Hal ini dapat menyebabkan kebocoran informasi mengenai penawaran harga *project* yang dilakukan oleh PT Pelangi Sentral Kreasi, sehingga *project* yang ditawarkan menjadi bisa tidak kompetitif dibandingkan dengan vendor lain selain PT Pelangi Sentral Kreasi. Demi menjaga keamanan dan privasi data, algoritme kriptografi dapat digunakan sebagai salah satu solusinya. Algoritme kriptografi adalah metode matematika untuk mengamankan komunikasi dan data dari akses oleh pihak yang tidak berwenang. Hasil akhir yang didapat dari penelitian, sistem mampu melakukan enkripsi dan dekripsi file secara efisien dan nyaman bagi pengguna. Dengan menerapkan algoritme SHA-256 dan AES-256 serta menambah sedikit modifikasi pada hasil enkripsi rata-rata dibutuhkan waktu yang sangat singkat untuk mengekripsi file dengan ukuran 266 *kilobyte*, yakni hanya membutuhkan waktu 29.7 milidetik. dan 31.4 milidetik saja untuk proses dekripsi file.

Kata Kunci: Kriptografi, Enkripsi, AES-256, SHA-256, Keamanan Data

IMPLEMENTATTION OF SHA-256 AND AES-256 CRYPTOGRAPHY ALGORITHMS FOR SECURING FILE AT PT PELANGI SENTRAL KREASI

Abstract- *PT Pelangi Sentral Kreasi is an Indonesian company operating in the creative industry sector. Established in 2005, it focuses on the development and production of innovative and high-quality creative products. Currently, the handling of project proposal data by PT Pelangi Sentral Kreasi is considered insecure, as the data is still in the form of cipher files that can be read by anyone who has the file. This can lead to information leaks regarding project pricing proposals by PT Pelangi Sentral Kreasi, potentially making their offered projects less competitive compared to other vendors apart from PT Pelangi Sentral Kreasi. To maintain data security and privacy, cryptographic algorithms can be utilized as one of the solutions. Cryptographic algorithms are mathematical methods used to secure communication and data from unauthorized access. The research outcome indicates that the system is capable of efficiently performing file encryption and decryption, providing user convenience. By implementing SHA-256 and AES-256 algorithms along with slight modifications to the encryption process, it takes a very short amount of time to encrypt a 266-kilobyte file, specifically only 29.7 milliseconds. The decryption process for the file takes only 31.4 milliseconds.*

Keywords*: Cryptography, Encryption, AES-256, SHA-256, Data Security*

1. PENDAHULUAN

Keamanan dan kerahasiaan dokumen merupakan aspek yang sangat penting dalam dunia informasi sekarang ini, apalagi informasi yang disimpan dan dikirim bersifat penting dan rahasia. Kriptografi merupakan salah satu solusi atau metode pengamanan dokumen yang tepat untuk menjaga kerahasiaan dan keaslian dokumen, serta dapat meningkatkan aspek keamanan suatu dokumen atau informasi [1].

Di dalam dunia bisnis dan organisasi, keamanan komputer sangatlah penting untuk menjaga keamanan data dan informasi. Kebocoran data atau serangan terhadap sistem komputer dapat mengakibatkan kerugian material maupun immaterial. Keamanan data menjadi sangat penting dalam menghindari terjadinya pencurian data, kebocoran data, dan penggunaan data secara tidak sah oleh pihak yang tidak bertanggung jawab. Oleh karena itu, perlu adanya pengamanan data yang dapat memberikan privasi dan keamanan yang optimal.

Kriptografi adalah teknik untuk memastikan kerahasiaan dan keamanan pesan saat dikirim ke tujuan, mencegah penyadapan. Enkripsi, dekripsi, dan kunci adalah tiga dasar kriptografi. Fungsi utama kriptografi adalah merahasiakan kunci dan mengubah plaintext menjadi chiphertext, sehingga plaintext menjadi password yang tidak

diketahui orang lain, tanpa harus merahasiakan algoritme yang digunakan. Jika kata sandi digunakan untuk kerahasiaan dan keamanan dapat dibaca oleh orang lain, maka kerahasiaannya terancam.Alhasil dapat dikatakan aman digunakan sebagai aplikasi berupa pesan teks, gambar, suara, video, dan media lainnya. Algoritme kriptografi adalah metode matematika untuk mengamankan komunikasi dan data dari akses oleh pihak yang tidak berwenang dengan harapan dapat meningkatkan keamanan data dan menghindari kebocoran informasi kepada vendor lain ketika tahap *Offering Project*. Algoritme kriptografi digunakan untuk mengenkripsi data, yaitu merubah data asli menjadi data yang tidak dapat dibaca oleh pihak yang tidak memiliki akses [2].

Kata kriptografi terdiri dari dua bagian yang berasal dari bahasa yunani, yaitu kripto dan graphia dimana kripto dapat diartikan sebagai rahasia (*secret*) dan graphia sebagai tulisan (*writing*) [3]. Berdasarkan istilahnya kriptografi merupakan ilmu dan seni pengamanan pesan saat pesan dipindahkan pada suatu tempat ketempat lainnya. Kriptografi adalah suatu ilmu menganalisis teknik matematika yang berkaitan dengan pengamanan informasi seperti penyembunyian data, kesahan data, integritas data, serta keaslian data. Kriptografi yaitu ilmu pengetahuan dan seni melindungi pesan supaya terjaga. Sasaran penggunaan kriptografi yaitu membentuk sesuatu yang samar, berupa pesan rahasia seperti teks, suara, gambar dan video [4]

Kriptografi mempelajari bagaimana menjaga keamananan suatu pesan (*plaintext*). Tugas utama kriptografi adalah melakukan penjagaan pesan atau kunci maupun keduanya agar tetap terjaga kerahasiaannya dari penyadap (*attacker*). Penyadap pesan diasumsikan mempunyai akses yang lengkap dalam saluran komunikasi antara pengirim pesan dan penerima pesan [5]

PT Pelangi Sentral Kreasi adalah perusahaan Indonesia yang bergerak di bidang industri kreatif. Didirikan pada tahun 2005, perusahaan ini memiliki kantor pusatnya di Jakarta. Sejak awal berdirinya, PT Pelangi Sentral Kreasi telah fokus pada pengembangan dan produksi produk-produk kreatif yang inovatif dan berkualitas tinggi. Dengan pengalaman dan keahlian yang dimiliki, PT Pelangi Sentral Kreasi terus berupaya untuk tetap menjadi pemimpin dalam industri kreatif Indonesia. Mereka terus mengikuti perkembangan tren dan teknologi terkini serta berinvestasi dalam sumber daya manusia yang berkualitas dan infrastruktur yang modern guna menjaga keunggulan kompetitifnya dengan vendor lain saat melakukan penawaran *project*. Saat ini penanganan data penawaran *project* yang dilakukan oleh PT Pelangi Sentral Kreasi tergolong belum aman karena data masih dalam bentuk *cipherfile* yang dimana masih bisa dibaca oleh semua orang yang memiliki file. Hal ini dapat menyebabkan kebocoran informasi mengenai penawaran harga project yang dilakukan oleh PT Pelangi Sentral Kreasi, sehingga *project* yang ditawarkan menjadi bisa tidak kompetitif dibandingkan dengan *vendor* lain selain PT Pelangi Sentral Kreasi.

Untuk itu, demi menjaga keamanan dan privasi data, algoritme kriptografi dapat digunakan sebagai salah satu solusinya. Dalam penelitian ini, penulis akan mencoba membahas mengenai penerapan algoritma kriptografi menggunakan algoritma SHA-256 dan AES-256 dengan judul "Penerapan Algoritma Kriptografi SHA-256 Dan AES-256 Untuk Pengamanan File Pada PT Pelangi Sentral Kreasi". SHA-256 adalah algoritme *hash* yang digunakan untuk menghasilkan nilai *hash* unik dari suatu input data, *Secure Hash Algorithm* (SHA-256) [8] merupakan fungsi hash yang umum digunakan, sampai saat ini belum ada yang dapat memecahkan algoritme fungsi hash SHA-256 [4][9]. Sedangkan, AES-256 (*Advanced Encryption Standard* - 256) adalah algoritme simetris yang digunakan sebagai algoritme enkripsi dan dekripsi. AES-256 merupakan algoritme kriptografi simetris dengan panjang blok cipher 256 [6][7]. Dalam penerapannya, algoritme kriptografi AES-256 (*Advanced Encryption Standard* - 256) [10] dapat digunakan pada berbagai jenis data, seperti data teks, data gambar, data *audio*, dan data *video*. Dengan menggunakan algoritma kriptografi, data akan lebih aman dari akses oleh pihak yang tidak memiliki akses. Selain itu, penggunaan algoritme kriptografi juga dapat memberikan kepercayaan bagi pengguna bahwa data yang mereka kirim atau simpan di sistem telah terlindungi dengan baik. [2].

2. METODE PENELITIAN

2.1 Algoritme SHA-256

Secure Hash Algorithm (SHA-256) merupakan fungsi hash yang umum digunakan, sampai saat ini belum ada yang dapat memecahkan algoritme fungsi hash SHA-256. Algoritme SHA-256 memiliki 8 langkah pengerjaan yaitu sebagai berikut :

a. Tambahkan *bit Padding*
   Pesan diisi sehingga panjangnya kongruen dengan 448, modulus 512. *Padding* 1 *bit* ditambahkan di akhir pesan, diikuti oleh banyaknya nol yang diperlukan sehingga panjang bit sama dengan 448 modulus 512.
b. Panjang *Append*
   Representasi panjang pesan 64 bit ditambahkan pada hasil akhirnya, langkah ini untuk membuat panjang pesan kelipatan 512 *bit*.

c. *Parsing* Pesan
Pesan *padding* diuraikan menjadi N blok pesan 512 bit, M(1), M(2),...M(N), dengan menambahkan blok 64 bit..

d. Inisialisasi Nilai Hash
Nilai hash awal, H(0) diatur, terdiri dari delapan kata 32 bit, dalam bentuk heksadesimal.

Tabel 1 *Initial Hash Value*

Variabel	Hash Value	Variabel	Hash Value	Variabel	Hash Value	Variabel	Hash Value
$H_0^{(0)}$	=6A09E667	$H_1^{(0)}$	=BB67EA85	$H_2^{(0)}$	=3C6EF372	$H_3^{(0)}$	=A54FF53A
$H_4^{(0)}$	=510E527F	$H_5^{(0)}$	=9B05688C	$H_6^{(0)}$	=1F83D9AB	$H_7^{(0)}$	=5BE0CD19

e. Mempersiapkan jadwal pesan
SHA-256 menggunakan jadwal pesan enam puluh empat kata 32 bit, kata-kata dari jadwal pesan diberi label W0, W1, ..., W63.

$$Wt = \int_{\sigma_1^{(256)}}^{M_t^{(t)}}(W_{i-2}) + W_{i-7} + \sigma_0^{(256)}(W_{i-15}) + W_{i-16} \quad \frac{0 \le t \le 15}{16 \le t \le 63} \tag{1}$$

Dimana :

$$\sigma_1^{(256)}(W_{i-2}) = \left((W_{i-2})rotr\ 17\right) \oplus ((W_{i-2})rotr\ 19) \oplus ((W_{i-2})shr\ 10) \tag{2}$$

$$\sigma_0^{(256)}(W_{i-15}) = \left((W_{i-15})rotr\ 7\right) \oplus ((W_{i-15})rotr\ 18) \oplus ((W_{i-2})shr\ 3) \tag{3}$$

Keterangan:
Wt = Blok pesan yang baru
Mt = Blok pesan yang lama
W_{i-2} = Blok pesan dari W ke i-2
W_{i-15} = Blok pesan dari W ke i-15
rotr = *Rotate Right*
shr = *Shift Right*
⊕ = Operator XOR

f. Mempersiapkan jadwal pesan
Inisialisasi delapan variabel kerja a, b, c, d, e, f, g, dan h dengan nilai hash (i-1) For t= 0 to 63.

T_1 = h + h + $\sum_1^{(256)}(e) + Ch(e,f,g)K_1^{(256)} + W_t$ (4)
T_2 = $\sum_0^{(256)}(\alpha)$ + Maj(a,b,c) (5)
$h = g$ (6)
$g = f$ (7)
$f = e$ (8)
$e = d + T_1$ (9)
$d = c$ (10)
$c = b$ (11)
$b = a$ (12)
$\alpha = T_1 + T_2$ (13)

Dimana :

$$\sum_1^{(256)}(e) = (\text{e}\ rotr\ 6) \oplus (\text{e}\ rotr\ 11) \oplus (\text{e}\ rotr\ 25)$$

$$\sum_0^{(256)}(\alpha) = (\text{e}\ rotr\ 2) \oplus (\text{e}\ rotr\ 13) \oplus (\text{e}\ rotr\ 22)$$

$Ch(e, f, g) = (e \wedge f) \oplus (\sim e \wedge g)$
$Maj(a, b, c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$
a, b, c, d, e, f, g, h = Variabel yang berisi pesan *hexadecimal*
$K_1^{(256)}$ = Konstansta SHA-256
rotr = *Rotate Right*
⊕ = Operator XOR
∧ = Operator *AND*

Tabel 2 Konstanta SHA-256

428A2F98	71374491	B5C0FBCF	E9B5DBA5	3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3	72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC	2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7	C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13	650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3	D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5	391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208	90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

g. Menjumlahkan hasil akhir *a, b, c, d, e, f, g, h* dengan inisial *hash value* $H^{(i)}$

Tabel 3 Table Penjumlahan *Hash*

H_0 (i)	$=a+H_0$ (i)	H_1 (i)	$=a+H_1$ (i)	H_2 (i)	$=a+H_2$ (i)	H_3 (i)	$=a+H_3$ (i)
H_4 (i)	$=a+H_4$ (i)	H_5 (i)	$=a+H_5$ (i)	H_6 (i)	$=a+H_6$ (i)	H_7 (i)	$=a+H_7$ (i)

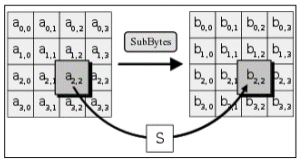
h. Output

Setelah mengulangi langkah 1 hingga 4 sebanyak N kali, fungsi *hash* yang dihasilkan adalah sebagai berikut :

H_0 (N) || H_1 (N) || H_2 (N) || H_3 (N) || H_4 (N) || H_5 (N) || H_6 (N) || H_7 (N)

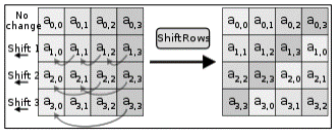
2.2 Algoritme AES-256

Algoritme AES didesain dengan menggunakan 4 transformasi yaitu Subbyte *Transformation*, *Shift Row Transform*, *Mix Column Transform*, dan *Add Round Key*.



Gambar 1 Transformasi *SubBytes*

Pada gambar 1 menunjukan bahwa pada tiap bit ai,j "status" ditukar dengan *SubBytes* S (ai,j) dengan kotak substitusi 8 bit. Langkah ini memberi sifat *nonlinea* dalam penyandian ini.

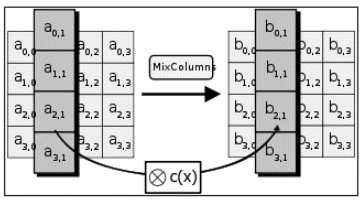


Gambar 2 Transformasi *ShiftRows*

Pada gambar 2 menunjukan transformasi *ShiftRows* yang mengubah tiap baris dengan uraian sebagai berikut :

1. Bagian AES, baris pertama dibiarkan.
2. Tiap bit pada baris kedua digeser sekali.
3. Pada baris ketiga dan keempat pun digeser sekali.
4. Baris ketiga digeser dua kali dan baris keempat digeser tiga kali.

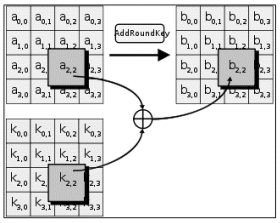
Langkah-langkah yang telah diuraikan di atas dapat dijalankan agar tiap kolom tidak dienkripsi masing-masing.



Gambar 3 Transformasi *MixColumns*

Pada Gambar 3 menunjukan bahwa empat bit dalam tiap kolom "status" digabung dengan transformasi linear yang ditukar. Fungsi *MixColumns* menerima empat bit masukan dan mengeluarkan empat bit yang tiap bit

masukannya saling memengaruhi. Fungsi ini bersama dengan *ShiftRows* memberikan penghamburan dalam penyandian.



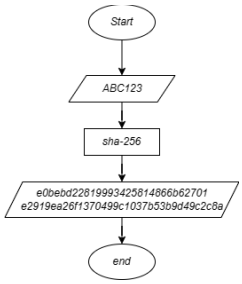
Gambar 4 Transformasi *AddRoundKey*

Pada gambar 4 menunjukan bahwa sub kunci digabung dengan "status". Tiap ronde sebuah sub kunci dibuat dari kunci utama dengan penjadwalan kunci AES. Tiap sub kunci berukuran sama dengan "status". Sub kunci ditambahkan menggunakan penggabungan tiap bit "status" dan bit yang letaknya sama pada subkunci dengan operasi XOR.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Algoritme *Hash* SHA-256

Pada tahap ini akan dilakukan proses *hash* pada kunci enkripsi agar tidak bisa dibaca dan jauh lebih aman. Berikut ini pada gambar 5 adalah contoh dari hasil kunci enkripsi yang di *hash* dari *plaintext* menjadi *ciphertext* :

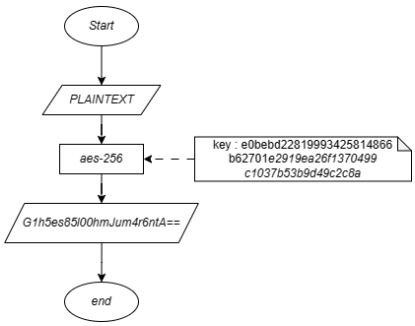


Gambar 5 Proses *Hash* Kunci Enkripsi

Nantinya yang akan digunakan sebagai kunci enkripsi adalah nilai *hash* dari “ABC123” yakni “e0bebd22819 993425814866b62701e2919ea26f1370499c1037b53b9d49c2c8a”

3.2 Implementasi Algoritme Enkripsi AES-256

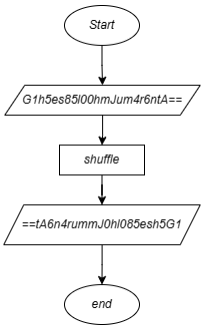
Setelah kita mendapkan kunci enkripsi yang telah didapatkan dari proses *hash* SHA-256 sebelumnya langkah berikutnya adalah melakukan enkripsi menggunakan algoritme AES-256 terhadap data yang akan dienkripsi. Contoh hasil dari enkripsi terdapat pada gambar 6 berikut :



Gambar 6 Proses Enkripsi AES-256

3.3 Modifikasi Hasil Enkripsi

Untuk memperkuat hasil enkripsi, disini akan dilakukan lagi modifikasi terhadap hasil enkripsi yang telah dilakukan dengan melakukan shuffle terhadap hasil enkripsi. Proses shuffle hasil enkripsi dapat dilihat pada gambar 7 berikut :



Gambar 7 Proses *Shuffle* Hasil Enkripsi AES-256

Shuffle dilakukan dengan mencacah hasil enkripsi menjadi *array* dengan besar 2 bit, kemudian posisi *array* ditukar posisinya dari yang *array* pertama menjadi *array* terakhir dan seterusnya sampai *array* habis.

3.4 Pengujian Program

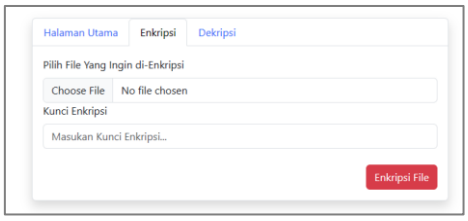
Pertama-tama untuk melakukan enkripsi data harus melakukan *login* terlebih dahulu ke dalam aplikasi. Tampilan halaman login dapat dilihat pada gambar 8 berikut :

Gambar 8 Tampilan Halaman *Login*

Untuk melakukan enkripsi data dilakukan dengan cara masuk ke menu enkripsi dari halaman utama aplikasi. Tampilan utama aplikasi dapat dilihat pada gambar 9 berikut :

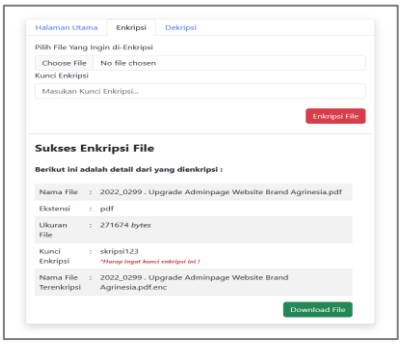
Gambar 9 Tampilan Halaman Utama Aplikasi

Setelah mengklik menu enkripsi maka akan tampil halaman enkripsi seperti pada gambar 10 berikut :



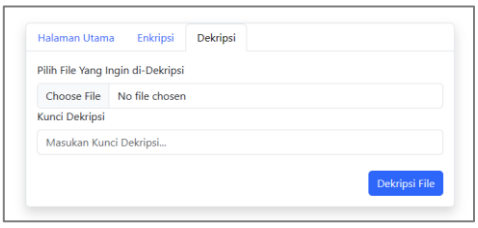
Gambar 10 Tampilan Halaman Enkripsi

Selanjutnya kita tinggal memilih file yang akan dienkripsi serta kunci enkripsi yang akan digunakan lalu mengklik tombol enkripsi file. Setelah mengklik tombol enkripsi file, akan muncul tombol download file dan detail dari file yang telah dienkripsi.



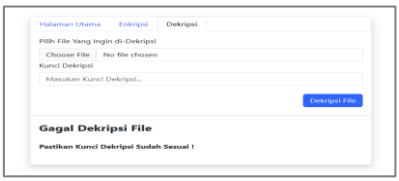
Gambar 11 Tampilan Detail Setelah Melakukan Enkripsi File

Setelah melakukan download pada file yang telah dienkripsi. Untuk mengembalikan file kedalam bentuk asalnya bisa dilakukan dengan mengklik menu dekripsi. Tampilan menu dekripsi tidak jauh berbeda dengan yang ada pada menu enkripsi. Hanya saja fungsi menu dekripsi ini berbeda dengan menu enkripsi. Menu dekripsi berfungsi untuk mengembalikan data yang telah dienkripsi menjadi bentuk sebelum enkripsi dengan memasukan kunci enkripsi yang dibuat sebelumnya pada menu enkripsi. Tampilan menu dekripsi dapat dilihat pada gambar 12 berikut ini :



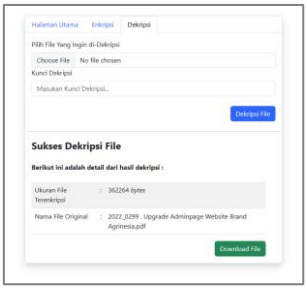
Gambar 12 Tampilan Halaman Dekripsi

Selanjutnya untuk mengembalikan data yang telah dienkripsi sebelumnya, sangat mirip dengan proses enkripsi file sebelumnya. Namun dekripsi file harus dilakukan dengan memasukan kunci dekripsi yang tepat. Kunci dekripsi adalah kunci yang sebelumnya dimasukan pada menu enkripsi sebelumnya. Jika salah dalam memasukan kunci dekripsi maka akan muncul pesan gagal melakukan dekripsi file seperti pada gambar 13 berikut ini :



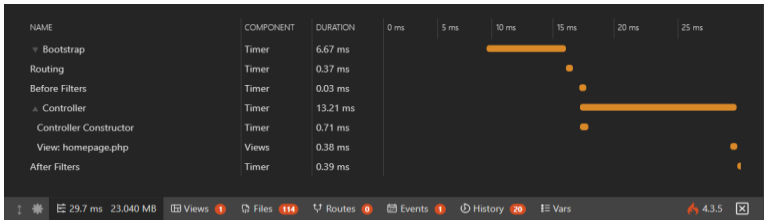
Gambar 13 Tampilan Gagal Melakukan Dekripsi

Oleh karena itu penting untuk mengingat kunci yang telah dibuat sebelumnya. Jika tidak maka file akan selamanya tidak bisa dikembalikan ke bentuk aslinya. Sebaliknya jika proses dekripsi sukses maka akan tampil detail dari file asli yang sebelumnya dienkripsi dan juga tombol download. Detail dekripsi file yang berhasil akan tampil seperti gambar 14 berikut ini :



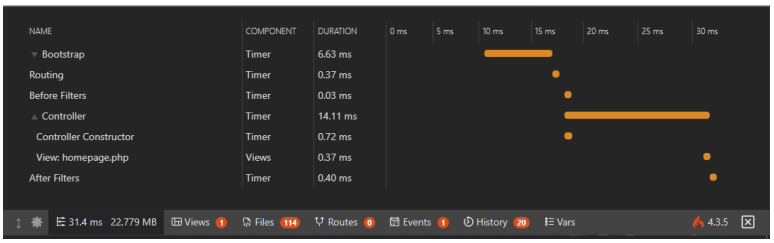
Gambar 14 Tampilan Detail Dekripsi

Setelah dilakukan pengujian terhadap proses enkripsi file, tahapan enkripsi file hanya memakan waktu 29.7ms terhadap enkripsi file pdf dengan ukuran 266Kb.



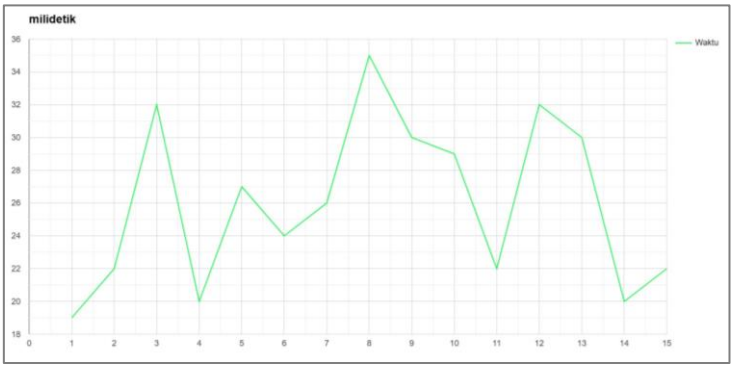
Gambar 15 Hasil Test Performa Enkripsi *File*

Selain itu untuk performa dekripsi file dengan ukuran yang sama hanya memakan waktu 31.4ms.

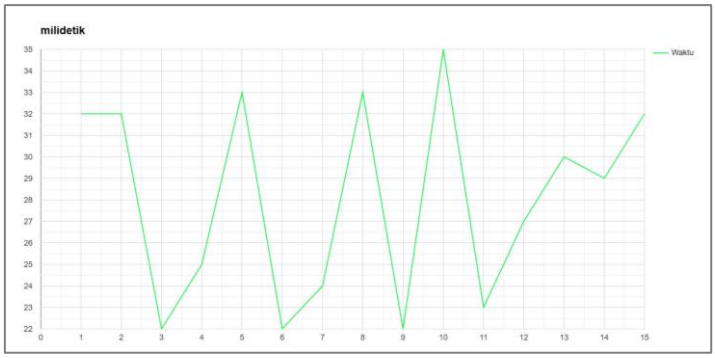


Gambar 16 Hasil Test Performa Dekripsi *File*

Dari pengujian sebanyak 15 kali, untuk mengenkripsi-dekripsi *file* dengan ukuran 266 *kilobyte* membutuhkan waktu antara 19 milidetik sampai 35 milidetik dengan detail sebagai berikut :



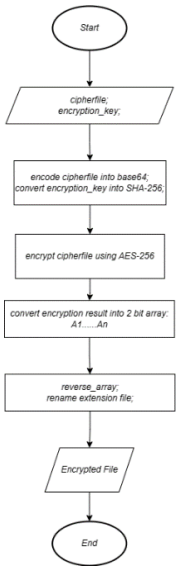
Gambar 17 Rata-rata performa enkripsi *file* sebanyak 15 x percobaan



Gambar 18 Rata-rata performa dekrispi *file* sebanyak 15 x percobaan

3.5 *Flowchart* Enkripsi File

Berikut ini adalah flowchart dari proses enkripsi yang terdapat pada aplikasi pada gambar 17 dibawah ini :

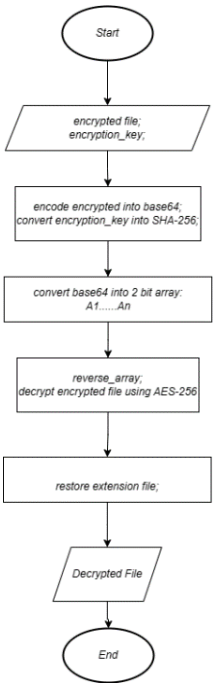


Gambar 19 *Flowchart* Enkripsi *File*

Sebelum dilakukan enkripsi pada file pada awalnya dilakukan *convert* terlebih dahulu pada *file* menjadi *text* menggunakan *base64 decode* dan juga *hash* kepada kunci enkripsi menggunakan SHA-256. Setelah mendapatkan kunci enkripsi yang telah di *hash* menggunakan SHA-256 barulah bisa dilakukan enkripsi file menggunakan algoritme AES-256. Setelah hasil dari enkripsi telah selesai, dilakukan modifikasi dengan mengubah terlebih dahulu hasil enkripsi menjadi array sebesar 2bit, dan dilakukan *reverse array* dengan menukar *array* pertama jadi terakhir, *array* kedua jadi kedua terakhir dst. Setelah selesai barulah dilakukan perubahan ekstensi file menjadi .enc dan pengguna bisa melakukan download pada file yang telah terenkripsi.

3.6 *Flowchart* Dekripsi File

Berikut ini adalah flowchart dari proses dekrispi yang terdapat pada aplikasi pada gambar 18 dibawah ini :



Gambar 20 *Flowchart* Dekripsi *File*

Sebelum dilakukan dekrispi pada file pada awalnya dilakukan *convert* terlebih dahulu pada *file* menjadi *text* menggunakan *base64 decode* dan juga *hash* kepada kunci enkripsi menggunakan SHA-256. Setelah mendapatkan kunci enkripsi yang telah di *hash* menggunakan SHA-256 barulah dilakukan proses dekripsi. Pertama dilakukan *convert base64 text* ke *array* sebelum dilakukan decrypt menggunakan kunci enkripsi yang telah kita hash sebelumnya. Setelah proses decrypt selesai dilakukan pengembalian nama ekstensi ke nama ekstensi awal. Dan pengguna bisa mendownload file yang telah dikembalikan ke bentuk awalnya.

4. KESIMPULAN

Berdasarkan hasil penelitian terhadap penerapan Algoritme AES-256 dan SHA-256 terhadap enkripsi file dapat ditarik kesimpulan bahwa, dari hasil akhir dari pengujian sistem, ditemukan bahwa sistem mampu melakukan enkripsi dan dekripsi file secara efisien dan nyaman bagi pengguna. Dengan menerapkan algoritme SHA-256 dan AES-256 serta menambah sedikit modifikasi pada hasil enkripsi ternyata hanya dibutuhkan waktu yang sangat singkat untuk mengekripsi file dengan ukuran 266 *kilobyte*, yakni hanya membutuhkan waktu 29.7 milidetik. dan 31.4 milidetik saja untuk proses dekripsi file.

DAFTAR PUSTAKA

[1] Widodo, B.E. and Purnomo, A.S. (2020) 'Implementasi Advanced Encryption Standard Pada Enkripsi Dan Dekripsi Dokumen Rahasia Ditintelkam Polda Diy', *Jurnal Teknik Informatika (JUTIF)*, 1(2), pp. 69–77.

[2] Riadi, I., Fadlil, A. and Tsani, F.A. (2022) 'Pengamanan Citra Digital Berbasis Kriptografi Menggunakan Algoritma Vigenere Cipher', *JISKA (Jurnal Informatika Sunan Kalijaga) 7.1*, 1, pp. 33–45.

[3] Sari, M., Dwi Purnomo, H. and Sembiring, I. (2022) 'Algoritma Kriptografi Sistem Keamanan SMS di Android', *Journal of Information Technology*, 2(1), pp. 11–15.

[4] Nasution, A.B. (2019) 'Implementasi Pengamanan Data Dengan Menggunakan Algoritma Caesar Cipher Dan Transposisi Cipher', *(JurTI) Jurnal Teknologi Informasi 3.1*, 1, pp. 1–6.

[5] Saputra, I. and Darma Nasution, S. (2019) 'Analisa Algoritma SHA-256 Untuk Mendeteksi Orisinalitas Citra Digital', *Prosiding Seminar Nasional Riset Information Science (SENARIS)*, 1.

[6] Fawzan, I. (2021) 'INTEGRASI OPENSSL DAN AES-256 UNTUK MENINGKATKAN KEAMANAN TRANSMISI DATA', *Diss. Universitas Siliwangi*, 1.

[7] Dakhi, O. *et al.* (2020) 'Analisis Sistem Kriptografi dalam Mengamankan Data Pesan Dengan Metode One Time Pad Cipher', *INVOTEK: Jurnal Inovasi Vokasional Dan Teknologi 20.1*, 1, pp. 27–36.

[8] Sembiring, H., Manik, F.Y. and Zaidah, T. (2019) 'Penerapan Algoritma Secure Hash Algorithm (SHA) Keamanan Pada Citra', *Media Informasi Analisa dan Sistem 1*, 1, pp. 33–36.

[9] Ramdani, F.C., Rahmatulloh, A. and Shofa, R.N. (2023) 'Implementation of JSON Web Token on Authentication with HMAC SHA-256 Algorithm', *Sistemasi: Jurnal Sistem Informasi* , 1, pp. 194–205.

[10] Ignasius, A. and Sakti, D.V.S.Y. (2022) 'Penerapan Algoritma AES (Advance Encryption Standart) 128 untuk Enkripsi Dokumen di PT. Gunung Geulis Elok Abadi', *SKANIKA 5.1*, 1, pp. 1–10.