Jurnal TICOM: Technology of Information and Communication
Volume 12, Nomor 2, Januari 2024 Hal: 50-55 E-ISSN: 2962-7982

Implementasi Kriptografi Berbasis Web dengan Algoritma Advanced Encryption Standard (AES) 256 dan Kompresi Huffman untuk Pengamanan File di SMK Satria

Nizamuddin Aulia Kafa[1], Dolly Virgian Shaka Yudha Sakti[2]

[1,2] Teknik Informatika, Program Studi Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail: [1]151150579@student.budiluhur.ac.id, [2]dolly.virgianshaka@budiluhur.ac.id
(*: corresponding author)

Abstrak—**Keamanan dan kerahasiaan data di zaman yang sudah maju ini terutama bagi suatu lembaga atau organisasi tertentu merupakan hal yang sangat dibutuhkan. Di suatu lembaga atau organisasi pastinya memiliki data yang sangat dirahasiakan dan pihak yang tidak berwenang pastinya tidak memiliki akses dalam membuka atau merubah data. Untuk menjaga keamanan data yang bersifat rahasia diperlukan suatu sistem yang digunakan untuk menyandikan data yang berupa file dan membukanya diperlukan kunci rahasia yang sifatnya sulit untuk dideteksi orang yang tidak berhak membukanya. Keamanan data pada Tata Usaha di SMK SATRIA yang bergerak dalam bidang pengolahaan data yang berfungsi untuk mengatur pelaksanaan administrasi program dan anggaran sekolah, terutama pada data guru, siswa keuangan dan nilai yang penting untuk mendukung pembelajaran sekolah. Agar tidak terjadi manipulasi data maka diperlukan sebuah aplikasi kriptografi yang dapat mengamankan data dari terjadinya manipulasi data. Dalam penelitian ini membahas konsep yang dapat digunakan untuk menjaga keamanan data yaitu ilmu kriptografi. Metode kriptografi yang akan digunakan dalam penelitian penelitian ini adalah AES 256 sebagai algoritma kriptografinya dan metode kompresi Huffman sebagai mekanisme kompresi. Sehingga keamanan data akan lebih terhindar dari manipulasi data oleh pihak yang tidak berwenang dan data file tidak terlalu besar setelah dijalankan proses. Dokumen yang diuji adalah dokumen bertipe .doc, .docx, .xls, .xlsx, dan .pdf. Dari hasil implementasi dan pengujian didapat kesimpulan bahwa algoritma kriptografi AES 256 dan kompresi Huffman dapat diimplementasikan dalam menjaga kerahasiaan dan keamanan data.**

Kata Kunci—**Kriptografi, Kompresi, Algoritma, AES 256, Huffman**

Abstract—***Data security and confidentiality in this advanced era, especially for certain institutions or organizations, is something that is really needed. An institution or organization must have data that is strictly confidential and unauthorized parties certainly do not have access to open or change the data . To maintain the security of confidential data, a system is needed that is used to encode data in the form of files and to open them, a secret key is needed which is difficult to detect for people who are not authorized to open it. Data security in Administration at SATRIA Vocational School which operates in the field of data processing which functions to regulate the administration of school programs and budgets, especially teacher, student financial and value data which is important to support school learning. To prevent data manipulation, a cryptographic application is needed that can secure data from data manipulation. This research discusses concepts that can be used to maintain data security, namely cryptography. The cryptographic method that will be used in this research is AES 256 as the cryptographic algorithm and the Huffman compression method as the compression mechanism. So data security will be more protected from data manipulation by unauthorized parties and data files will not be too large after the process is carried out. The documents tested were documents of type .doc, .docx, .xls, .xlsx, and .pdf. From the implementation and testing results, it was concluded that the AES 256 cryptographic algorithm and Huffman compression could be implemented to maintain data confidentiality and security.***

Keyword—***Cryptography, Compression, Algorithms, AES 256, Huffman***

I. PENDAHULUAN

Kerahasiaan dari data merupakan suatu hal yang dijaga kerahasiaannya dalam melakukan pertukaran data yang dirahasiakan agar tidak dapat dibaca atau dibuka oleh pihak yang tidak berhak[1]. Upaya dalam menjaga kerahasiaan dari data tersebut sudah tercetus sejak jaman dahulu tepatnya pada jaman romawi dengan metode pergeseran huruf atau karakter dengan dasar nilai tertentu. Pada jaman modern berbasis teknologi komputer, upaya tersebut telah berkembang dengan seiringnya zaman yaitu dengan menggunakan algoritma yang telah diciptakan oleh banyak ahli tetapi upaya tersebut masih saja dipecahkan oleh pihak yang tidak berwenang, maka dari itu dibutuhkannya suatu algoritma kriptografi yang dikembangkan untuk terjaganya suatu keamanan data oleh pihak yang tidak bertanggung jawab[2]. Untuk mengamankan data yang penting dapat dilakukan dengan cara enkripsi (*encryption*) atau sering disebut dengan Kriptografi [3]-[5]. Tujuan dari enkripsi kriptografi adalah membuat data penting sulit untuk dibaca sekalipun data tersebut berhasil dicuri pihak yang tidak bertanggung jawab. Dalam kriptografi terdiri dari dua hal, enkripsi (*encryption*) yaitu proses merubah data asli (*plaintext*) menjadi data samaran (*chippertext*) dan dekripsi

(*decryption*) yaitu proses pengembalian *ciphertext* menjadi *plain text* kembali [6].

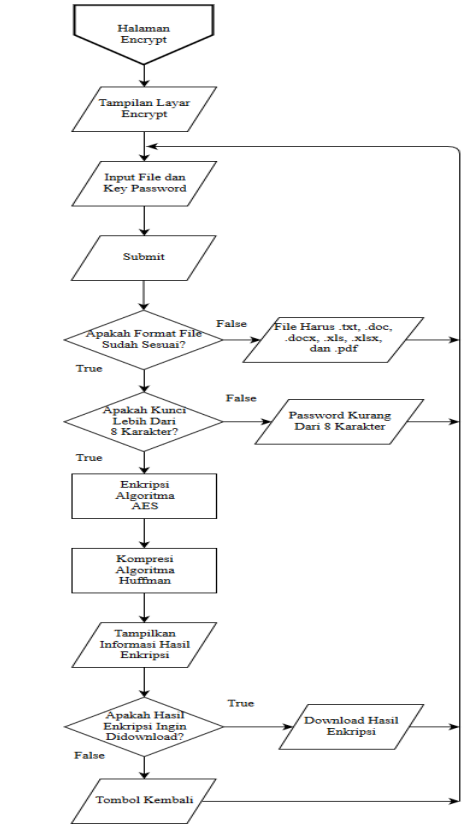
SMK Yayasan Tarbiyah Islamiyah Al-Alawiyah atau biasa disebut SATRIA adalah salah satu lembaga pendidikan formal yang memberikan pelayanan, dan bimbingan pendidikan sebagai lanjutan dari SMP/MTS. Tata Usaha (TU) pastinya menyimpan data guru, siswa, keuangan, dan nilai yang penting untuk mendukung pembelajaran sekolah. Agar tidak terjadi manipulasi data guru, siswa, keuangan dan nilai maka diperlukan sebuah aplikasi kriptografi yang dapat mengamankan data dari terjadinya pencurian data.Pada kedua proses enkripsi dan dekripsi dibutuhkan suatu pengaman yang menjamin bahwa data terlindungi pada prosesnya, pengaman tersebut dinamakan *key* [7]. Fungsi dari *key* bertujuan untuk mengawali dan membuka setiap proses enkripsi atau dekripsi, pada penelitian ini contoh *key* yang digunakan adalah algoritma simetris (*Simetric-key*) yaitu algoritma yang menggunakan satu kunci yang sama untuk melakukan proses penguncian yang disebut enkripsi dan pembuka kunci atau yang bisa disebut dekripsi, contoh algoritma yang digunakan adalah AES 256 dan Kompresi Huffman.

Aplikasi keamanan data yang akan dibuat dengan menggunakan kriptografi dengan algoritma AES 256 dan Kompresi Huffman merupakan metode untuk mengamankan suatu data [8]. Karena kebutuhan pengamanan informasi suatu data komputer melalui kriptografi sangat dibutuhkan di SMK Satria, kebutuhannya mencakup data-data soal, guru, siswa/siswi, nilai, dan keuangan agar tidak dapat dimanipulasi oleh orang yang tidak bertanggung jawab. Kriptografi AES 256 dan Kompresi Huffman cocok digunakan karena bentuk data yang akan dilakukan kriptografi ekstensi filenya berbentuk .doc, .docx, .xls, .xlsx, dan .pdf. AES256 merupakan satu dari banyak algoritma yang digunakan untuk melakukan kriptografi, *Advanced Encryption System* (AES) adalah algoritma kriptografi yang beroperasi dalam mode Block Cipher yang memproses blok data 128-bit dengan panjang kuncinya 128-bit untuk AES-128, 192-bit untuk AES-192 dan 256-bit untuk AES-256 [9]. Kompresi Huffman digunakan untuk merepresentasikan suatu data digital dengan sedikit bit, dan tetap mempertahankan kebutuhan dalam bentuk data aslinya walaupun data tersebut lebih kecil ukurannya [10].

II. METODE PENELITIAN

A. *Flowchart* Proses Enkripsi Aplikasi

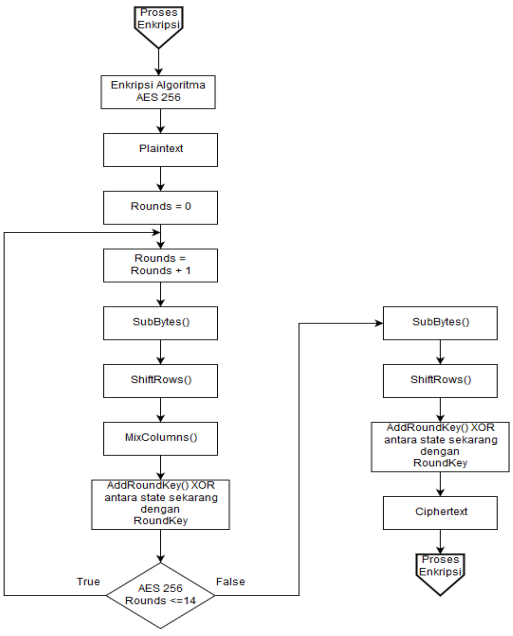
Alur utama pada *flowchart* proses enkripsi digambarkan pada Gambar 1, menjelaskan jalur proses melakukan enkripsi dimana admin melakukan enkripsi file dan memasukan *password* agar terjaga kerahasiaannya.



Gambar 1. *Flowchart* Proses Enkripsi Aplikasi

B. *Flowchart Proses Enkripsi AES 256*

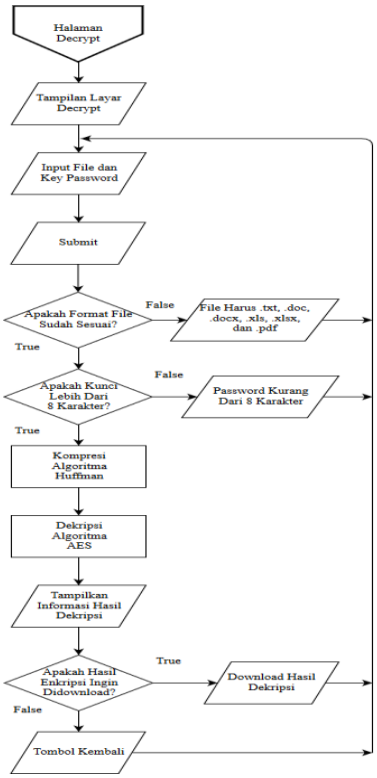
Alur utama pada *flowchart* proses enkripsi Algoritma AES digambarkan pada *flowchart* Gambar 2 menjelaskan algoritma AES 256 melakukan enkripsi file memasukan *password* yang dari *plaintext* diubah menjadi *ciphertext*.



Gambar 2. *Flowchart* Proses Enkripsi AES 256

C. *Flowchart* Proses Dekripsi Aplikasi

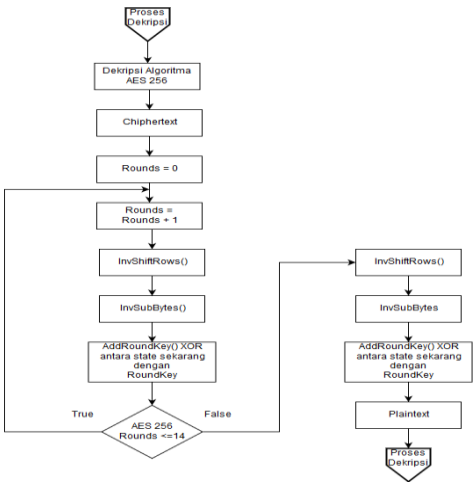
Alur utama pada *flowchart* proses dekripsi digambarkan pada Gambar 3 menjelaskan jalur proses melakukan dekripsi dimana admin dapat mengembalikan file yang sudah dienkripsi kembali keadaan semula tanpa mengubah isi file.



Gambar 3. *Flowchart* Proses Dekripsi Aplikasi

D. *Flowchart* Proses Dekripsi *AES 256*

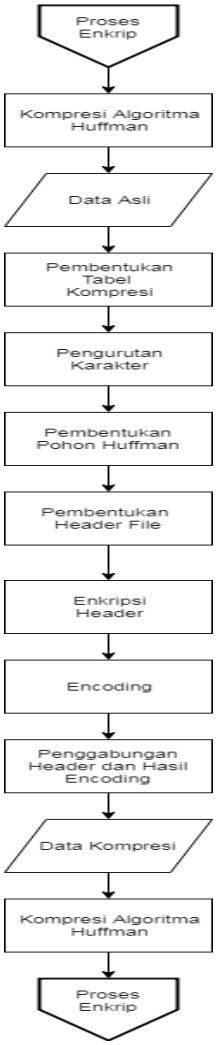
Alur utama pada *flowchart* proses dekripsi Algoritma AES digambarkan pada Gambar 4, menjelaskan algoritma AES melakukan enkripsi file memasukan *password* yang dari ciphertext diubah menjadi *plaintext*. Jika Hasil putaran *Rounds* berjumlah 14 = AES 256.



Gambar 4. *Flowchart* Proses Dekripsi AES 256

E. *Flowchart* Proses Kompresi *Huffman*

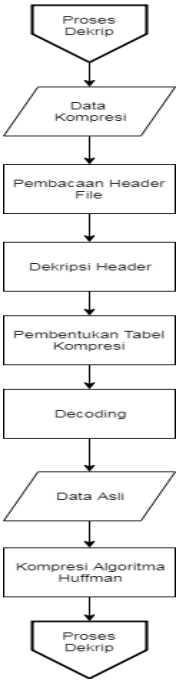
Alur utama pada *flowchart* enkripsi kompresi Huffman digambarkan pada Gambar 5, menjelaskan algoritma kompresi Huffman melakukan proses *encoding* dan kompresi file, sehingga ukuran file menjadi lebih kecil. Contoh proses kompresi memiliki String "ABACCDA" melakukan proses menjadi binner, membuat pohon Huffman atau tabel Huffman, ulangi langkah-langkah tersebut sampai seluruh karakter di-*encoding* dan binner tersebut menjadi berikut ini: 'FGGGFFHHIF': 0100101101110 rangkaian ini diubah menjadi 13 bit.



Gambar 5. *Flowchart* Proses Kompresi Huffman

F. *Flowchart Proses Dekompresi Huffman*

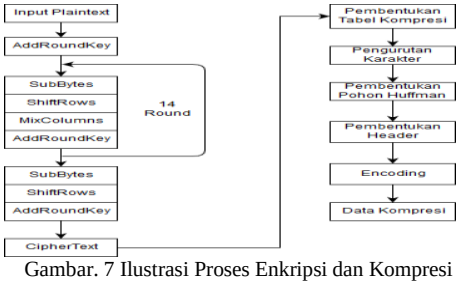
Alur utama pada *flowchart* dekompresi Huffman digambarkan pada Gambar 6, menjelaskan algoritma kompresi Huffman melakukan proses *decoding* dan mengembalikan ukuran file seperti semula. Contoh proses Dekompresi dari hasil kompresi: "0100101101110" melakukan proses membaca semua string binner, dekripsi *header*, tabel Huffman, pohon Huffman dan ulangi langkah langkah tersebut sampai membaca karakter daun menjadi *decoding* dan menghasilkan string "FGGGFFHHIF".



Gambar 6. *Flowchart* Proses Dekompresi Huffman

G. Ilustrasi Proses Enkripsi dan Kompresi

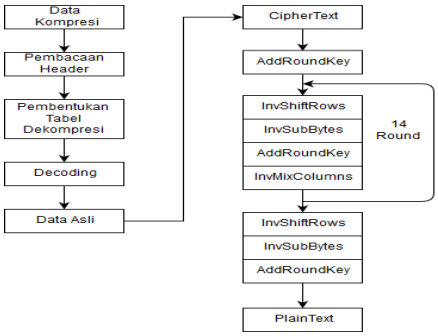
Pada Gambar 7, dijelaskan bahwa terjadinya proses Enkripsi yang menggunakan algoritma AES 256 dan dilanjutkan dengan proses kompresi Huffman yang berguna untuk memperkecil suatu ukuran file agar tidak terlalu besar.



Gambar. 7 Ilustrasi Proses Enkripsi dan Kompresi

H. Ilustrasi Proses Dekripsi dan Dekompresi

Pada Gambar 8 dijelaskan bahwa terjadinya proses dekompresi yang berguna untuk proses pengembalian filesebelum terjadinya kompresi dan proses dekripsi yang berguna untuk mengembalikan sebelum proses enkripsi.



Gambar 8. Proses Dekripsi dan Dekompresi

III. Hasil dan Pembahasan

A. Tampilan Layar Halaman *Login*

Tampilan dari halaman *Login* pada Gambar 9 menjelaskan login yang akan dilakukan oleh admin dengan cara memasukan nip dan *password* admin lalu klik *Login* yang akan mengarahkan ke halaman Beranda, *button* "Daftar Admin Disini" juga berfungsi untuk pendaftaran Id admin yang baru.

Gambar 9. Halaman Login

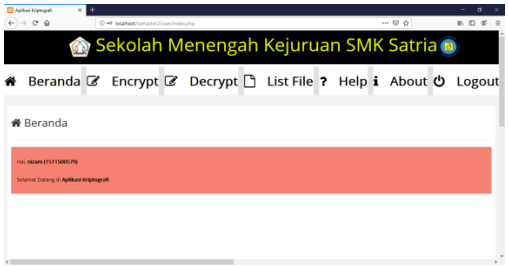
B. Tampilan Layar Halaman *Register*

Tampilan dari halaman *Register* pada Gambar 10 menjelaskan *register* yang akan dilakukan dengan cara mendaftarkan nip, nama, *password* dan *confirm password* yang ingin dimasukan oleh admin baru dan klik Daftar pada *button* yang tersedia.

Gambar. 10 Tampilan Layar Halaman Registrasi

C. Tampilan Layar Halaman Beranda

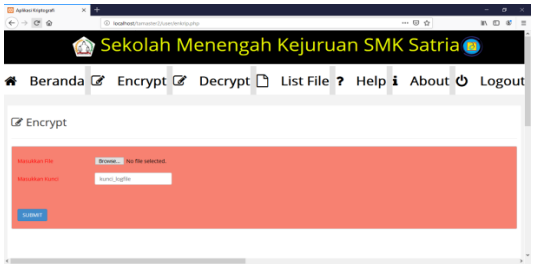
Tampilan dari Halaman Beranda pada Gambar 11 menjelaskan beberapa *button* halaman yang tersedia seperti Beranda, *Encrypt*, *Decrypt*, *List File*, *Help*, *About* dan *Logout* serta menampilkan nip dan nama admin yang telah *login*.



Gambar 11. Tampilan Layar Halaman Beranda

D. Tampilan Layar Halaman *Encrypt*

Tampilan dari Halaman *Encrypt* pada Gambar 12 menjelaskan tentang proses enkripsi dengan cara memilih file yang akan dienkripsi dan memasukan *key* yang ingin dimasukan dengan minimum *key* 8 digit.



Gambar 12. Tampilan Layar Halaman *Encrypt*

E. Tampilan Layar Hasil *Encrypt*

Tampilan dari halaman hasil *Encrypt* pada Gambar 13 menjelaskan hasil dari enkripsi yang akan menampilkan nama file, tipe file, ukuran file, file hasil dan waktu proses melakukan enkripsi, hasil enkripsi dapat dilihat di drive D:/hasil atau juga bisa dilakukan dengan menekan tombol *download*.

Gambar 13. Tampilan Layar Hasil Encrypt

F. Tampilan Layar Halaman *Decrypt*

Tampilan dari halaman *Decrypt* pada Gambar 14 menjelaskan tentang proses dekripsi yang dilakukan dengan cara mencari file yang sudah di enkripsi dan memasukan *key* yang sebelumnya pernah dimasukan.

Gambar 14. Tampilan Layar Halaman *Decrypt*

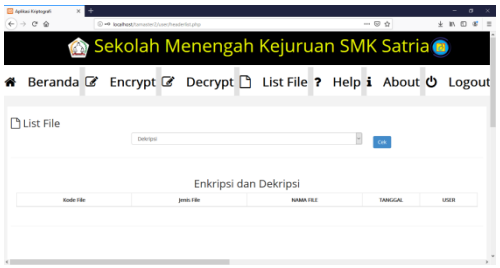
G. Tampilan Layar Hasil *Decrypt*

Tampilan dari halaman *Decrypt* pada Gambar 15 menjelaskan tentang proses dekripsi yang dilakukan dengan cara mencari file yang sudah di enkripsi dan memasukan *key* yang sebelumnya pernah dimasukan.

Gambar 15. Tampilan Layar Hasil *Decrypt*

H. Tampilan Layar *List File*

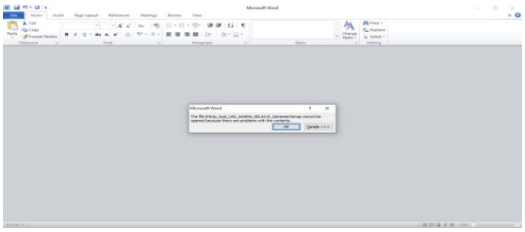
Tampilan Layar List File pada gambar 16 menjelaskan tentang proses menampilkan hasil enkripsi dan dekripsi, ketika *user* menekan tombol enkripsi maka akan menampikan hasil dari enkripsi dan ketika *user* menekan tombol dekripsi maka akan menampilkan hasil dari dekripsi.



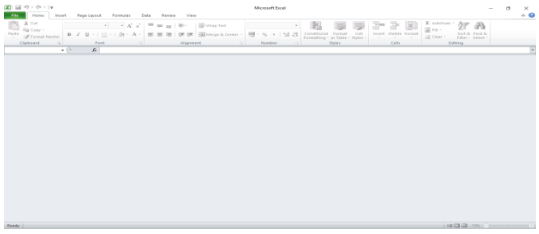
Gambar 16. Tampilan Layar *List File*

I. Tampilan File Setelah *Enkripsi*

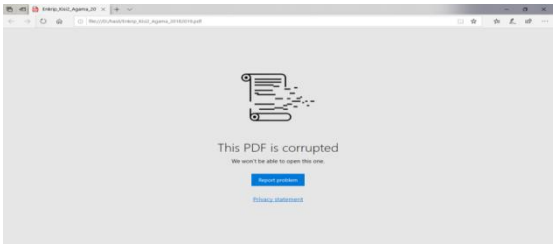
Gambar 17, 18, dan 19 menampilkan hasil dari enkripsi yang dilakukan, *file* .doc, xlsx dan pdf tidak dapat dibuka atau dinyatakan *corrupted*.



Gambar 17. Tampilan File setelah Enkripsi (1)



Gambar 18. Tampilan File Setelah Enkripsi (2)



Gambar 19. Tampilan File Setelah Enkripsi (3)

J. Tabel Hasil Pengujian

Pada Hasil Pengujian ini akan membahas perbandingan file yang sudah dienkripsi dan dekripsi *file* yang diuji berupa .doc, .xlsx dan .pdf pengujian *file* mencakup ukuran *file*, waktu proses enkripsi dekripsi, dan tipe *file*. Hasil pengujian

Enkripsi ditampilkan oleh Tabel 1 dan pengujian Dekripsi ditampilkan Tabel 2.

TABEL I
HASIL PENGUJIAN ENKRIPSI

No.	Nama *File* Awal	Ukuran Awal (bytes)	Tipe *File*	Waktu Enkripsi (Seconds)	Ukuran Hasil Enkripsi (bytes)	Nama *File* Hasil Enkripsi
1	Soal UTS AGAMA KELAS 10 Semester Genap.docx	57456	.doc	7	7802	Enkrip_ Soal UTS AGAMA KELAS 10 Semester Genap.docx
2	Laporan Keuangan	276948	.xlsx	31	77853	Enkrip_ Laporan Keuangan .xlsx
3	Kisi Agama 2018/2019 .pdf	1272212	.pdf	142	277952	Enkrip_ Kisi Agama 2018/2019 .pdf

TABEL II.
HASIL PENGUJIAN DEKRIPSI

No.	Nama *File* Setelah Enkripsi	Ukuran Awal Enkripsi (*bytes*)	Tip *File*	Waktu Dekripsi (*Seconds*)	Ukuran Hasil Dekripsi (*bytes*)	Nama *File* Hasil Dekripsi
1	Enkrip_ Soal UTS AGAMA KELAS 10 SemesterGenap .docx	57802	.doc	6	57456	Dekrip_ Soal UTS AGAMA KELAS 10 Semester Genap .docx
2	Enkrip_ Laporan Keuangan .xlsx	277853	.xlsx	31	276948	Dekrip_ Laporan Keuangan .xlsx
3	Enkrip_ Kisi Agama 2018/2019 .pdf	1277952	.pdf	140	1272212	Dekrip_ Kisi Agama 2018/2019 .pdf

IV. KESIMPULAN

Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa :

a. File yang dihasilkan setelah melakukan enkripsi menjadi lebih aman dari adanya pencurian, jika file tersebut dapat dicuri sekalipun file tersebut tidak dapat dibuka atau dimanipulasi. Hasil file yang sudah di enkripsi ukurannya tidak terlalu besar karena menggunakan kompresi Huffman.
b. Telah dibuktikan bahwa waktu proses enkripsi dipengaruhi oleh ukuran dari file asli. Semakin besar ukuran file maka hasil dari file enkripsi tersebut menjadi lebih besar dan membutuhkan proses yang memakan cukup waktu.
c. Aplikasi enkripsi yang digunakan dapat menggunakan file berformat .doc, .xlsx, dan pdf sebagai file yang akan diamankan.

REFERENSI

[1] S. Rahmawati, I. Taufik, and G. Sandi, "Implementasi Algoritma AES (Advanced Encryption Standard) 256 Bit Dan Kompresi Menggunakan Algoritma Huffman Pada Aplikasi Voice Recorder," *SENTER* , pp. 91–99, 2017.

[2] A. S. Y. Irawan, et al., "Pengamanan File Video dengan Algoritma Advanced Encryption Standard (AES)," *SYSTEMATICS*, vol. 2, no. 1, pp. 28–32, 2020.

[3] A. E. Putri, A. Kartikadewi, and L. A. A. Rosyid, "Implementasi Kriptografi Dengan Algoritma Advanced Encryption Standard (AES) 128 Bit dan Steganografi Menggunakan Metode End of File (EOF) Berbasis Java Desktop pada Dinas Pendidikan Kabupaten Tangerang," *Applied Information Systems and Management (AISM)*, vol. 3, no. 2, pp. 69–78, 2020.

[4] R. N. Fitriana and D. Djuniadi, "Analisis Perbandingan Algoritma AES dan RC4 pada Enkripsi dan Dekripsi Data Teks Berbasis CrypTool 2," *SYSTEMIC: Information System and Informatics Journal*, vol. 7, no. 2, pp. 1–7, Dec. 2022.

[5] R. Satria Wibowo, C. Nugrahaeni, and K. Kunci, "Implementasi Keamanan File dengan Kompresi Huffman dan Kriptografi Advanced Encryption Standart (AES) pada Pengamanan File Data Antemortem," *Seminar Nasional Pengaplikasian Telematika (SINAPTIKA 2021)*, pp. 1–9, 2021.

[6] F. A. Sitorus, N. B. Nugroho, and U. F. S. S. Pane, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Bit untuk Keamanan Data Transaksi Penjualan pada PT. Mitsubishi Electric Indonesia," *Jurnal CyberTech*, vol. x, no. x, pp. 1–15, 2020.

[7] R. S. Y. Simbolon and P. Silitonga, "Pengamanan Data dengan Menggunakan Teknik Kriptografi Public RSA dan Knapsack," *KAKIFIKOM (Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer)*, vol. 03, no. 01, pp. 9–12, 2021.

[8] R. Febrianto and S. Waluyo, "Implementasi Algoritme Kriptografi Advanced Encryption Standard (AES-256) untuk Mengamankan Database Penilaian Karyawan pada KJPP NDR," *BIT (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 20, no. 1, pp. 44–49, 2023.

[9] L. Laurentinus, H. A. Pradana, D. Y. Sylfania, and F. P. Juniawan, "Perbandingan Kinerja RSA dan AES Terhadap Kompresi Pesan SMS Menggunakan Algoritme Huffman," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 171–177, Jul. 2020.

[10] F. D. Yonathan, H. Nasution, and H. Priyanto, "Aplikasi Pengaman Dokumen Digital Menggunakan Algoritma Kriptografi Hybrid dan Algoritma Kompresi Huffman," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 7, no. 2, pp. 181–195, 2021.