Terbit online pada laman web jurnal: **https://journal.budiluhur.ac.id/index.php/bit**

JURNAL BIT
(Budi Luhur Information Technology)

Vol. 16 No. 2 (2019) 47 - 53 ISSN Media Elektronik: **2685-127X**

Aplikasi Pengamanan File Dengan Metode Kriptografi AES 192, RC4 Dan Metode Kompresi Huffman

Wahyu Pramusinto [1)], Nugroho Wizaksono[2)], Ari Saputro[3)]
Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi luhur [1)2)3)]
Jl. Ciledug Raya, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : wahyu.pramusinto@budiluhur.ac.id[1)], Nugrohowizaksono@gmail.com[2)], ari.saputro@budiluhur.ac.id[3)]

Abstract

Generally in a company, digital documents or files are stored on a computer without special security. This documents can be taken by other people who have access to a computer. For security, documents in plain text must be changed to cipher text so they cannot be read. However, before converted the file to cipher text, the document is compressed so that the size becomes smaller. After that, the encryption process will be carried out. In this research we made an application to create documents that cannot be read by others. The application uses the Huffman compression algorithm, then encrypted using AES 192 and RC4 cryptography. This application is based on the web with the PHP programming language dan MySQL database server. From the experimental of this application, the file managed to become unreadable dan successfully secured. This application can also update files that have been encrypted to normal.

Keywords: *cryptography, AES, RC4, Compression file*

Abstrak

Pada umumnya di sebuah perusahaan, dokumen digital atau *file* disimpan di komputer tanpa adanya pengamanan secara khusus. Hal ini memungkinkan terjadinya pencurian dokumen karena dokumen tersebut bisa diambil oleh orang lain yang memiliki akses ke komputer. Untuk keamanan, dokumen dalam bentuk plain teks harus diubah menjadi *cipher text* agar tidak bisa dibaca. Akan tetapi sebelum diubah menjadi *cipher text*, dokumen dikompresi terlebih dahulu agar ukurannya menjadi lebih kecil. Setelah itu baru dilakukan proses enkripsi. Pada penelitian ini dibuat sebuah aplikasi untuk mengamankan dokumen sehingga tidak bisa dibaca oleh orang lain. Aplikasi menggunakan algoritma kompresi Huffman, kemudian dienkripsi menggunakan kriptografi AES 192 dan RC4. Aplikasi ini dibuat berbasis web dengan bahasa pemrograman PHP dan database server MySQL. Dari hasil percobaan didapat kesimpulan bahwa aplikasi ini berhasil mengamankan *file* sehingga tidak bisa dibaca. Aplikasi ini juga dapat mengembalikan *file* yang sudah dienkripsi menjadi seperti semula.

Kata kunci: kriptografi, AES, RC4, kompresi huffman

1. Pendahuluan

Selain memberikan dampak positif, teknologi informasi juga memberikan dampak negatif. Salah satu dampak negatif dalam perkembangan teknologi adalah adanya penyadapan data dan ukuran file yang menjadi makin besar. Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu data, pesan dan informasi. Data menjadi hal vital di masa ini, terkait betapa pentingnya pihak atau orang berkepentingan yang dapat mengakses data tersebut. Data bisa berbentuk dokumen digital seperti word, pdf, excel, dan lain-lain. Apabila ada pihak yang tak berkepentingan mengakses data tersebut, maka dikhawatirkan akan terjadi hal yang tidak diinginkan. Karena itulah dibutuhkan suatu aplikasi yang dapat menjaga keamanan dokumen dan mengurangi ukuran file agar lebih mudah disimpan di dalam komputer. Cara yang bisa dilakukan adalam dengan teknik kriptografi.

Secara umum kriptografi dapat diartikan sebagai ilmu dan seni penyandian yang bertujuan untuk menjaga keamanan dan kerahasiaan suatu data. Kriptografi juga tidak berarti hanya memberikan

Diterima Redaksi : 01-02-2020 | Selesai Revisi : 02-02-2020| Diterbitkan Online : 03-02-2020

keamanan informasi saja, namun kriptografi lebih ke arah teknik-tekniknya [1].
Berdasarkan jenis kuncinya dikelompokkan menjadi, algoritma simetris (konvensional) dan algoritma asimetris atau algoritma *public key*. AES dan RC4 termasuk ke dalam algoritma simetris.

Permasalahan yang terjadi pada penelitian ini adalah bagaimana cara mengamankan dokumen digital yang tersimpan di komputer. Permasalahan lain adalah bagaimana cara mengimplementasikan algoritma kompresi Huffman, kriptografi AES, RC4 untuk mengamankan dokumen?

Tujuan dari penelitian ini sebagai berikut : dapat mengimplementasikan metode algoritma kriptografi *Advanced Encryption Standard* (AES), RC4, dan kompresi Huffman dalam bentuk aplikasi nyata, mengamankan data dan mengatur *file* yang bersifat privasi agar tidak dapat disalah gunakan oleh orang yang tidak bersangkutan, dan menghasilkan aplikasi penyimpanan dan pengamanan *file* berbasis web yang mudah digunakan dan dimengerti oleh *user*.

Agar tidak menyimpang dari topik penelitian, maka batasan masalahnya sebagai berikut : algoritma kriptografi yang digunakan adalah Advanced Encryption Standard (AES) dan RC4, menggunakan metode kompresi Huffman, ukuran file maksimal yang dapat diterima oleh aplikasi ini hanya sebesar 10MB, aplikasi yang dibuat berbasis web dengan menggunakan bahasa pemrograman PHP dan database server MySQL, tipe file yang digunakan dalam aplikasi ini adalah file dengan ekstensi : doc, docx, xls, xlsx, dan pdf, dan panjang kunci yang digunakan dalam algoritma kriptografi mengunakan metode AES sebesar 192 bit.

Penelitian berjudul aplikasi pengamanan email ldengan algoritma advanced encryption standard (AES), Rivest Cipher 4 (RC4) dan Caesar Cipher berhasil mengubah isi email menjadi menjadi suatu informasi yang tidak dapat dimengerti oleh siapapun dan diharapkan keamanan dalam pengiriman informasi melalui email dapat terjamin kerahasiaan dari sebuah informasi tersebut. Selain isi email, aplikasi ini juga dapat mengenkripsi lampiran email. Email diterima oleh penerima dalam bentuk cipher text dan dapat didekripsi kembali [2].

Penelitian dengan judul Aplikasi Keamanan Data Dengan Kriptografi RC4 dan Steganografi EOF Pada Media Video di MAN 1 Jakarta membuat sebuah aplikasi untuk mengamankan data. Metode yang digunakan adalah dengan cara melakukan enkripsi pada file dengan metode RC4, kemudian hasil enkripsi ini disembunyikan di dalam video dengan steganografi EOF. File yang bisa diproses pada penelitian ini adalah file doc, docx, pdf, xls dan xlsx. File video yang bisa digunakan untuk menampung adalah file mp4 dan avi. File *video* yang sudah disisipi *file* masih bisa dijalankan dan file yang disisipkan bisa dikembalikan lagi seperti semula tanpa perubahan apapun [3]

. Penelitian berjudul Kombinasi Algoritma AES, RC4 Dan Elgamal Dalam Skema Hybrid Untuk Keamanan Data mengkombinasikan beberapa algoritma kriptografi untuk menjaga kerahasiaan data. Kesimpulan dari penelitian ini adalah Model kombinasi dalam skema hybrid AES dan RC4 dapat digunakan untuk mengenkripsi pesan dengan aman karena dilakukan pengamanan lapis 2 dan juga pengamanan dilakukan pada kunci AES dan RC4 menggunakan Elgamal. [4].

Penelitian berkaitan kompresi file dengan judul Kompresi File Menggunakan Konversi Biner Hexadecimal dan Algoritma Huffman Encoding menyebutkan bahwa Data umumnya dikompresi terlebih dahulu agar proses pertukaran data tidak memakan waktu yang terlalu lama. Hasil dari penelitian ini adalah algoritma Huffman Encoding dan konversi Biner Hexadecimal dapat diimplementasikan untuk kompresi file. Kesimpulan dari aplikasi ini yaitu metode Huffman Encoding dan konversi Biner Hexadecimal dapat melakukan kompresi dengan baik pada 26 macam ekstensi *file* [5].

Penelitian dengan judul Algoritma Enkripsi RC4 Sebagai Metode Obfuscation Source Code PHP membuat sebuah aplikasi untuk mengamankan source code PHP. Hasil penelitian ini merupakan sebuah metode *obfuscation* untuk source code PHP dengan memanfaatkan algoritma enkrispi RC4 yang diterapkan dalam aplikasi [6].

Penelitian dengan judul Peningkatan Keamanan Pesan Dengan Kriptografi RC4 dan Steganografi LSB Pada File JPEG dihasilkan sebuah aplikasi dengan Visual Studio 2008 yang mengkombinasikan antara algoritma RC4 dan metode LSB untuk pengamanan data. Menurut penelitian ini lamanya waktu proses enkripsi dan dekripsi sangat bergantung pada banyaknya jumlah karakter, lamanya waktu proses enkripsi dan dekripsi berbanding lurus dengan banyaknya jumlah karakter yang digunakan [7].

Pada penelitian dengan judul Implementasi Algoritma RC4 Untuk Proteksi File MP3 membuat sebuah aplikasi untuk mengamankan file mp3. Aplikasi dibuat menggunakan system operasi Windows 8 yang merubah sistem Windows 8 yaitu bagian shell32. Dari hasil pengujian enkripsi dan deskripsi berhasil dilakukan dengan memberikan informasi nama file,ukuran bytes, dan estimasi waktu [8].

Penelitian dengan judul Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4 dibuat untuk mengamankan file gambar. Pada penelitian ini dilakukan modifikasi algoritma RC4 dengan menambahkan sebuah blok *initial vector* pada proses enkripsi maupun dekripsi serta melakukan pemindahan sejumlah bit pada posisi tertentu [9].

Penelitian berjudul Aplikasi Penyembunyian Multimedia Menggunakan Metode End Of File (EOF) Dan Huffman Coding membuat sebuah aplikasi dengan bahasa pemrograman Java untuk mengamankan file gambar, video dan audio. Hasil dari penelitian ini menunjukkan bahwa metode *End of File* efektif dalam menyisipkan data dengan ukuran besar, serta dengan diterapkannya metode *Huffman Coding*, maka ukuran data yang disisipkan berkurang [10].

Pada penelitian dengan judul Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen membuat suatu implementasi kriptografi AES-128 untuk melakukan enkripsi dan enkripsi data yang berupa file dokumen pdf, doc dan txt. Algoritma Advanced Encryption Standard (AES) dipilih karena memiliki suatu tingkat keamanan pertukaran informasi yang cukup bagus, dan pada penelitian ini diuji coba *file* dokumen untuk melihat kecepatan waktu yang dibutuhkan selama proses enkripsi dan dekripsi [11].

2. Metode Penelitian

Berikut ini adalah metode penelitian yang digunakan

Gambar 1 : metode penelitian

a. Studi Pustaka

Pada tahap ini akan dilakukan studi pustaka dari berbagai literatur. Studi pustaka dilakukan dengan cara mengumpulkan, membaca dan juga memahami junal, makalah serta referensi lain guna mendapatkan informasi yang dibutuhkan dalam menunjang penelitian.

b. Analisa Kebutuhan

Pada tahap ini dilakukan analisa apa saja yang diperlukan dalam membuat aplikasi.

c. Perancangan Aplikasi

Pada tahap ini akan dilakukan perancangan aplikasi, membuat gambaran aplikasi yang akan dibuat, pemilihan bahasa pemrograman yang digunakan.

d. Implementasi Aplikasi

Pada tahap ini akan dibuat aplikasi penyimpanan *file* dengan algoritma AES, RC4 dan Huffman pada bagian enkripsi-dekripsi serta kompresi dengan bahasa pemrograman PHP.

e. Pengujian Aplikasi

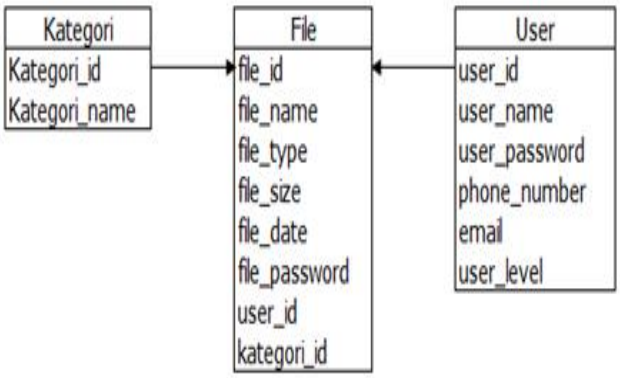
Pada tahap ini akan dilakukan pengujian dari aplikasi yang telah dibuat, serta mengevaluasi apabila masih terdapat kesalahan dan kekurangan.

f. Analisa hasil

Tahap ini adalah untuk melakukan analisa hasil dari aplikasi yang dibuat berdasarkan beberapa variabel inputan.

3. Hasil Dan Pembahasan

Aplikasi ini memerlukan sebuah basis data untuk menyimpan informasi yang ada di didalamnya. Ada 3 tabel yang dibutuhkan oleh aplikasi ini. Berikut adalah *class diagram* aplikasi

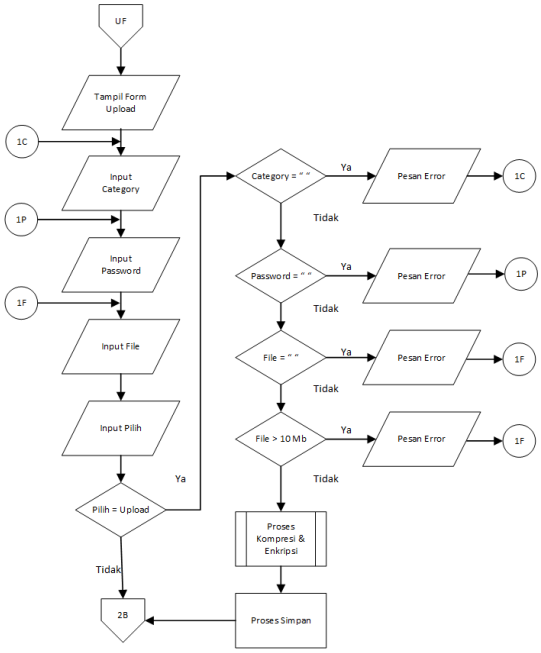


Gambar 2 : Class diagram aplikasi

Cara Kerja Aplikasi

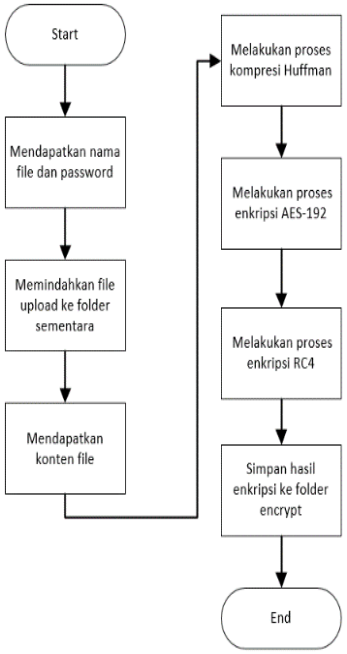
Untuk melakukan enkripsi *file*, user/admin dapat memilih *upload file* pada menu *upload*, kemudian memilih kategori dan memilih *file* doc, docx, xsl, xlsx, ppt, pptx, txt dan pdf, baru kemudian mengisi password untuk melakukan proses enkripsi. Namun ukuran file doc, docx, xsl, xlsx, ppt, pptx, txt dan pdf tidak boleh lebih besar dari ukuran file yang telah ditentukan. Untuk mengembalikan file yang sudah dienkripsi menjadi file asli, user/admin dapat memilih menu *download* dan memasukkan password yang digunakan saat enkripsi

Pada saat upload file ada beberapa kondisi yang harus dipenuhi. Di bawah ini adalah *flowchart* yang menjelaskan proses saat mengupload *file*.



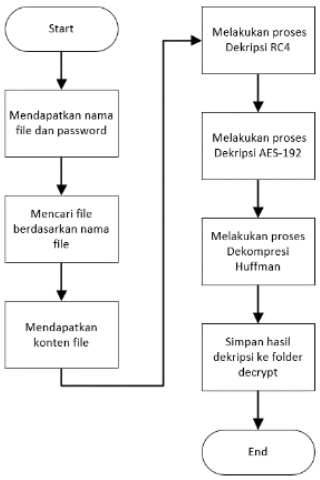
Gambar 3 : Flowchart Proses Upload File

Gambar 4 adalah f*lowchart* yang menjelaskan alur proses kompresi dan enkripsi pada aplikasi. Dari flowchart dapat dilihat bahwa kompressi Huffman dilakukan terlebih dahulu, kemudian file dienkripsi menggunakan AES-192 dan RC4.



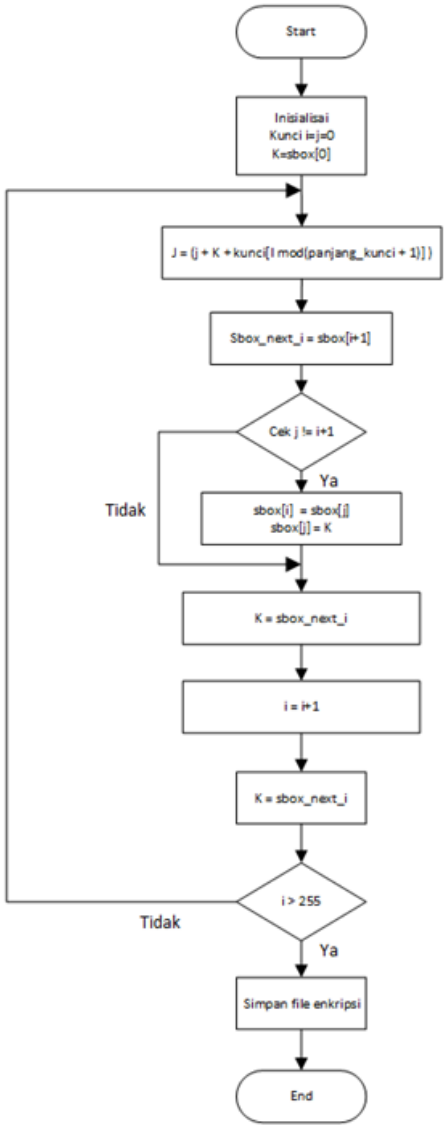
Gambar 4 : Flowchart Proses Kompresi dan Enkripsi

Gambar 5 adalah *flowchart* yang menjelaskan alur proses dekripsi dan dekompresi yang ada pada aplikasi. Pada flowchart dapat dilihat untuk mendapatkan file asli berturut-turut dilakukan proses dekripsi RC4, dekripsi AES-192 dan dekompresi Huffman.



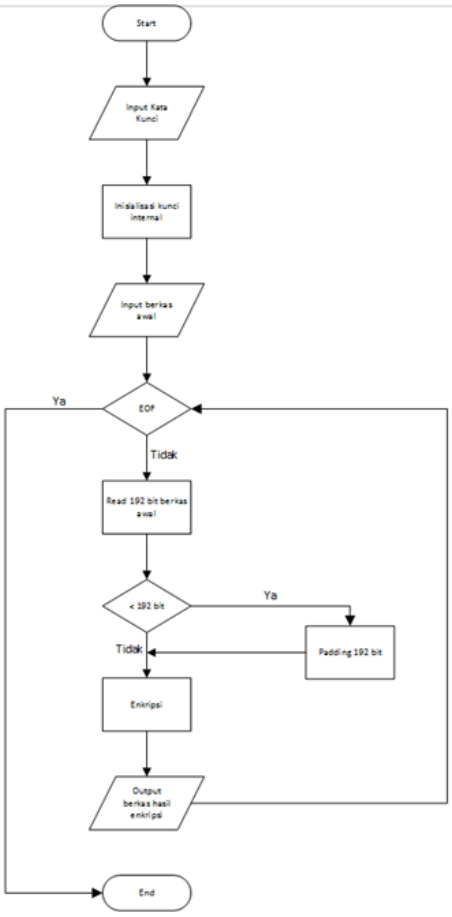
Gambar 5: Flowchart Proses Dekripsi dan Dekompresi

Gambar 6 adalah flowchart yang menjelaskan proses enkripsi RC4



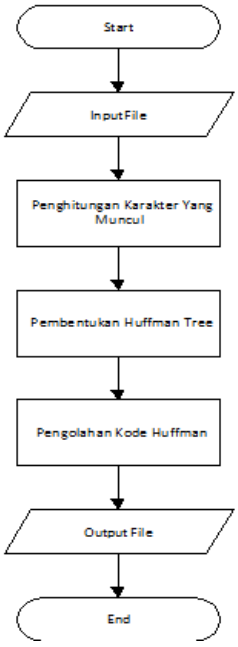
Gambar 6 : Flowchart Enkripsi RC4

Gambar 7 adalah *flowchart* proses enkripsi AES



Gambar 7 : Flowchart Enkripsi AES

Gambar 8 adalah *flowchart* proses kompresi Huffman



Gambar 8 : Flowchart Kompresi Huffman

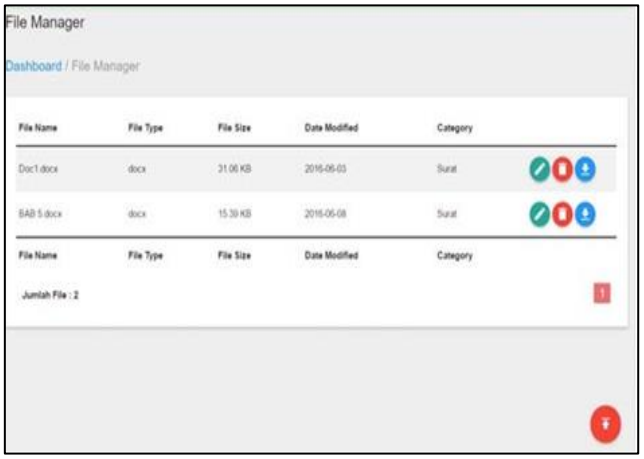
Spesifikasi *Hardware* dan *Software*

Agar aplikasi bisa berjalan dengan baik, spesifikasi perangkat yang dipakai untuk implementasi juga harus mendukung. H*ardware* yang dipakai dalam pengujian aplikasi ini adalah sebuah laptop dengan spesifikasi Processor Intel i5-3570, RAM 8 GB DDR3, VGA HD7850 2GB DDR5, Monitor 22" LCD 1920x1080, Keyboard, Mouse dan Harddisk 1 TB

Perangkat lunak atau *software* yang dipakai dalam pengujian aplikasi ini adalah Sistem Operasi Windows 10 64 bit, XAMPP Version 5.6.23, Browser Google Chrome Microsoft Word, Excel, Powerpoint , Foxit PDF Reader, dan Notepad ++.

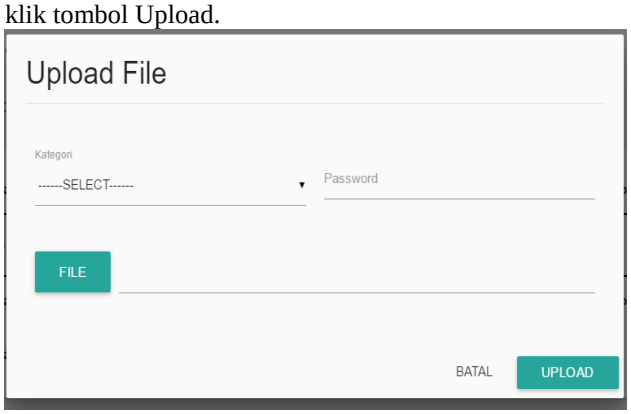
Tampilan Layar

Tampilan layar pada gambar 9 berisi data file yang sudah pernah dienkripsi. *File* bisa di-download ke komputer.



Gambar 9 : Tampilan Layar File Manager

Gambar 10 adalah tampilan halaman untuk upload file yang akan dikompresi dan dienkripsi. Pilih kategori, masukkan password dan file, kemudian klik tombol Upload.



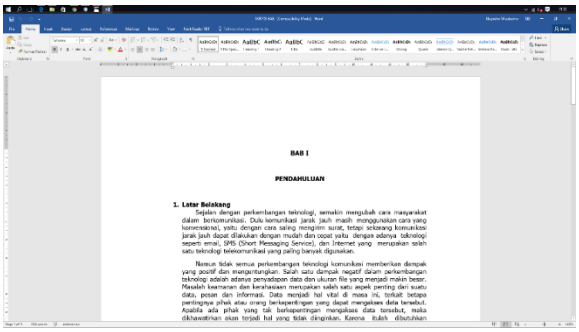
Gambar 10 : Tampilan Layar Upload File

Gambar 11 adalah tampilan layar halaman dekripsi. Upload file yang sebelumnya sudah dienkripsi, kemudian masukkan password yang digunakan saat enkripsi.

Gambar 11 : Tampilan Layar Dekripsi File

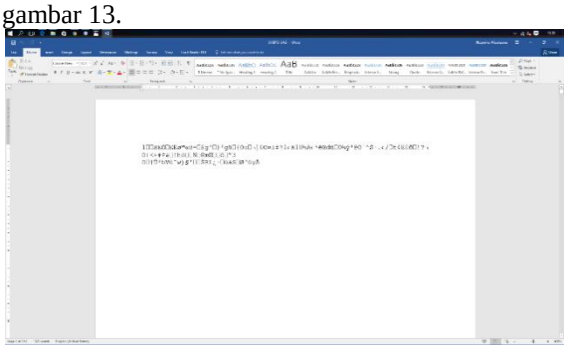
Pengujian Aplikasi

Berikut adalah hasil uji coba untuk *file* yang berekstensi doc yang ditunjukkan pada gambar 12.



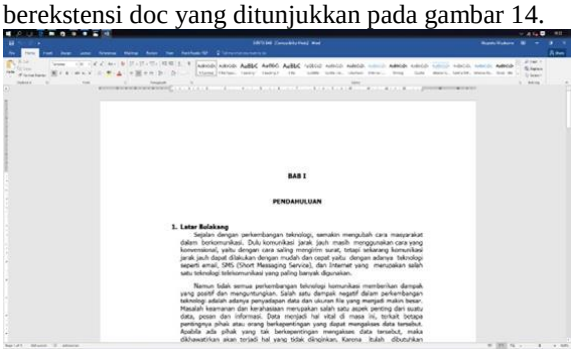
Gambar 12: File doc (Asli)

Setelah *file* berhasil dienkripsi, maka file DOC yang sudah dienkripsi masih bisa dibuka namun konten *file* sudah terenkripsi. Seperti terlihat pada gambar 13.



Gambar 13 : File doc Hasil Kompresi dan Enkripsi

Berikut adalah hasil uji coba dekripsi untuk file yang berekstensi doc yang ditunjukkan pada gambar 14.



Gambar 14 : File hasil dekripsi

Dalam pengujian ini, akan dibahas perbandingan antara proses enkripsi dan dekripsi *file*. Pengujiannya yaitu nama *file*, kata kunci, ukuran waktu proses enkripsi, dan waktu proses dekripsi seperti tabel 1.

Tabel 1 : Tabel Pengujian

Nama File	Kata Kunci	Ukuran asli	Ukuran File Kompresi	Ukuran File Enkrip	Waktu Enkrip	Waktu Dekrip
Pdf_1.pdf	25f9e794323b453885f5181f1b624d0b	7.266 KB	7.084 KB	9.445 KB	14 3.87 s	40 0 s
Pdf_2.pdf	25f9e794323b453885f5181f1b624d0b	1.528 KB	1.528 KB	2.037 KB	30. 38 s	52. 06 s
Xls_1.xlsx	25f9e794323b453885f5181f1b624d0b	3.355 KB	3.354 KB	4.472 KB	66. 81 s	22 6.44 s
Xls_2.xlsx	25f9e794323b453885f5181f1b624d0b	302 KB	301 KB	401 KB	5.9 7 s	30. 83 s
Xlsx_1	25f9e794323b453885f5181f1b624d0b	365 KB	351 KB	468 KB	7 s	66. 85
Xlsx_2	25f9e794323b453885f5181f1b624d0b	2.296 KB	2.199 KB	2.932 KB	44. 06 s	23 0.88 s
Doc_1	25f9e794323b453885f5181f1b624d0b	6.835 KB	4.576 KB	6.101 KB	93. 17 s	35 0.22 s
Doc_2	25f9e794323b453885f5181f1b624d0b	22 KB	9 KB	12 KB	0.1 9 s	3.0 4 s
Docx_1	25f9e794323b453885f5181f1b624d0b	3.303 KB	3.302 KB	4.403 KB	65. 81 s	21 8.69 s
Docx_2	25f9e794323b453885f5181f1b624d0b	66 KB	65 KB	87 KB	1.2 9 s	12. 39 s

Kelebihan aplikasi ini antara lain file yang disimpan dalam server sudah dienkripsi sehingga file aman dan isi file tidak mengalami perubahan. Aplikasi ini dapat dijalankan secara offline ataupun online. Terdapat user akses sehingga hanya dapat digunakan oleh orang yang berhak.

Kekurangan dari aplikasi ini yaitu aplikasi ini hanya dibatasi untuk mengenkripsi file dokumen pdf, doc, docx, txt, xls, dan xlsx saja dan ukuran satu file nya dibatasi maksimal 10 MB. Dari hasil uji coba, semakin besar ukuran file maka proses enkripsi maupun dekripsi semakin lama. Spesifikasi *hardware* yang rendah mempengaruhi lamanya proses enkripsi dan dekripsi.

4. Kesimpulan

Kesimpulan dari penelitian ini adalah dengan adanya aplikasi pengamanan *file* dokumen ini, *file* penting dapat lebih terjaga kerahasiaannya karena tersimpan di server dan hanya bisa diakses melalui aplikasi ini. Metode Algoritma AES dan RC4 menghasilkan *file ciphertext* memiliki ukuran lebih

besar dibandingkan *plaintext* nya. Metode Algoritma Kompresi Huffman menjadikan *file* menjadi lebih kecil. Kesimpulan lain yang bisa diambil yaitu waktu melakukan proses enkripsi dan dekripsi berbanding lurus dengan ukuran *file* yang diproses. Selain itu waktu juga bergantung pada spesifikasi *hardware* perangkat yang digunakan

Beberapa saran yang dapat diberikan untuk pengembangan aplikasi antara lain semua tipe file sebaiknya bisa dienkripsi di dalam aplikasi ini. komputer dengan spesifikasi yang lebih baik digunakan untuk performa yang lebih baik.

5. Daftar Pustaka

[1] Bhaudhayana, G. W. and Widiartha, I. M. .2015. *Implementasi algoritma kriptografi aes 256 dan metode steganografi lsb pada gambar bitmap*. Jurnal ilmu komputer Universitas Udayana, 8(2), pp. 15–25.

[2] Pramusinto, W., Putra, R.A. 2018. *Pengamanan Email Dengan Algoritma Advanced Encryption Standard (AES), Rivest Cipher 4 (RC4) dan Caesar Cipher*. *Prosiding SNST 2018.*

[3] Hanapi, A, Pramusinto, P. 2017. *Aplikasi Keamanan Data Dengan Kriptografi RC4 dan Steganografi EOF Pada Media Video di MAN 1 Jakarta*. Prosiding Seminar Nasional Multi Disiplin Ilmu 2017. Jakarta, 22 April 2017. ISSN 2087-0930.

[4] Widarma, A. 2016. *Kombinasi Algoritma AES, RC4 Dan Elgamal Dalam Skema Hybrid Untuk Keamanan Data* . Journal Of Computer Engineering System And Science), Vol 1 No 1, Januari 2016.

[5] Geofandy, K., Aubrey, E., Agung, H. 2019. *Kompresi File Menggunakan Konversi Biner Hexadecimal dan Algoritma Huffman Encoding.* Jurnal Ilmiah Teknologi Informasi Terapan, Vol 5 No 3 2019.

[6] Setiawan, O. Fiati, R., Listyorini, T. *Algoritma Enkripsi RC4 Sebagai Metode Obfuscation Source Code PHP*. 2014. Prosiding SNATIF Ke -1 Tahun 2014

[7] Sulaiman, R., Isnanto, B. 2018. *Peningkatan Keamanan Pesan Dengan Kriptografi RC4 dan Steganografi LSB Pada File JPEG*. Konferensi Nasional Sistem Informasi 2018

[8] Kirman. *Implementasi Algoritma RC4 Untuk Proteksi File MP3*. 2018. Jurnal Pseudocode, Volume V Nomor 1, Februari 2018, ISSN 2355-5920

[9] Zebua, T., Ndruru, E. 2017. *Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4.* Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) Vol. 4, No. 4, Desember 2017.

[10] Nasution, Y.R, Johar, A., Coastera, F.F. 2017. *Aplikasi Penyembunyian Multimedia Menggunakan Metode End Of File (EOF) Dan Huffman Coding.* Jurnal Rekursif, Vol. 5 No. 1 Maret 2017, ISSN 2303-0755.

[11] Prameshwari, A., & Sastra, N. 2018. *Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen*. Jurnal Eksplora Informatika, 8(1), 52-58.