Jurnal Teknik Informatika (JUTIF)
Vol. 1, No. 2, Desember 2020, hlm. 69-77

DOI: https://doi.org/10.20884/1.jutif.2020.1.2.21
p-ISSN: 2723-3863
e-ISSN: 2723-3871

IMPLEMENTASI ADVANCED ENCRYPTION STANDARD PADA ENKRIPSI DAN DEKRIPSI DOKUMEN RAHASIA DITINTELKAM POLDA DIY

Berita Estu Widodo*[1], A. Sidiq Purnomo[2]

[1,2]Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta,
Jl. Wates Km. 10 Yogyakarta 55753, Indonesia
Email: [1]si.b.estu@gmail.com, [2]sidiq@mercubuana-yogya.ac.id

(Naskah masuk: 31 Agustus 2020, diterima untuk diterbitkan: 04 Oktober 2020)

Abstrak

Keamanan dan kerahasiaan dokumen merupakan aspek yang sangat penting dalam dunia informasi sekarang ini, terlebih bagi instansi pemerintah, 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. Algoritma kriptografi yang digunakan untuk enkripsi dan dekripsi dokumen adalah algoritma *Advanced Encryption Standard* (AES-256) dan menggunakan kunci simetris pada proses enkripsi dan dekripsi. Aplikasi Akrid (Aplikasi Kriptografi Dokumen) yang dirancang adalah sebuah aplikasi kriptografi menggunakan metode algoritma *Advanced Encryption Standard* (AES-256), berbasis *client server* yang dapat dijadikan sebagai alternatif untuk mengamankan dokumen rahasia. Aplikasi ini ditujukan untuk mengamankan dokumen dengan format: .doc, .docx, .xls, .xlsx, .pdf dan .text; dan dapat dilihat berapa lama waktu yang dibutuhkan untuk proses enkripsi dan dekripsi. Hasil dari penelitian dapat disimpulkan bahwa algoritma kriptografi AES dapat diimplementasikan pada proses enkripsi dan dekripsi dokumen dengan waktu enkripsi 0,212 *second* untuk dokumen ukuran 19,212 Kb dan 20,533 *second* untuk dokumen ukuran 1.966 Kb sedangkan pada proses dekripsi membutuhkan waktu 0,213 *second* untuk dokumen ukuran 19,212 Kb dan 20,882 *second* untuk dokumen dengan ukuran 1.966 Kb. Jadi proses enkripsi dan dekripsi dokumen dengan *file* berformat doc, docx, xls, xlsx, pdf atau text membutuhkan waktu rata-rata sekitar 0,01 *second*/1 Kb.

Kata kunci : AES-256, enkripsi, dekripsi, keamanan dokumen.

THE IMPLEMENTATION OF ADVANCED ENCRYPTION STANDARD ON THE ENCRYPTION AND DECRYPTION OF THE CONFIDENTIAL DOCUMENTS AT DITINTELKAM POLDA DIY

Abstract

Documents' security and confidentiality are the very important aspects in today's world of information, especially for government agencies, particularly when the information stored and sent is important and confidential. Cryptography is an appropriate solution or documents security method for maintaining the documents' confidentiality and authenticity, and can increase the security aspect of a document or information. The cryptographic algorithm that is used for the documents' encryption and decryption is the Advanced Encryption Standard (AES-256) algorithm and using a symmetric key in the encryption and decryption process. The AKRID (Document Cryptography Application) which be designed is a cryptography application using the Advanced Encryption Standard (AES-256) algorithm, based on client-server, which can be used as an alternative to secure confidential documents. This application is intended to secure the documents in the following formats: .doc, .docx, .xls, .xlsx, .pdf, and .txt; and the duration of the encryption and decryption process can be seen. The research results conclude that the AES cryptographic algorithm can be implemented in the documents' encryption and decryption process, with an encryption time of 0.212 seconds for 19,212-kb document size, and 20.533 seconds for a 1,966-Kb document size; while the decryption process takes 0.213 seconds for 19,212-Kb document size, and 20.882 seconds for 1,966-Kb document size. Hence, the process of encrypting and decrypting documents (in the following formats: .doc, .docx, .xls, .xlsx, .pdf, .txt) averagely takes about 0.01 second per 1 Kb.

Keywords: *AES-256, decryption, document's security, encryption.*

1. PENDAHULUAN

Perkembangan teknologi komunikasi membuat pertukaran informasi bisa dilakukan dengan cepat, namun hal tersebut juga memberikan dampak yang kurang menguntungkan bagi dunia komunikasi. Penyadapan dokumen atau data merupakan hal yang paling ditakuti oleh pengguna jaringan komunikasi pada saat ini. Berbagai usaha dilakukan untuk menjamin agar dokumen rahasia yang dikirimkan tersebut tidak bisa diakses oleh pihak lain[1]. Salah satu cara yang bisa dilakukan adalah dengan penyandian. Kriptografi merupakan teknik penyandian dalam menjaga kerahasiaan dan keaslian dokumen, serta dapat meningkatkan aspek keamanan dari akses pihak yang tidak mempunyai kepentingan[2].

Ada berbagai macam algoritma dalam kriptografi salah satunya adalah Algoritma *Advanced Encryption Standard*, algoritma ini digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST dan sampai saat ini belum ada yang bisa memecahkannya. Berdasarkan kondisi tersebut aplikasi kriptografi ini dirancang dengan mengimplementasikan algoritma *Advanced Encryption Standard* (AES-256) pada enkripsi dan dekripsi dokumen. Dengan harapan informasi-informasi penting yang terdapat dalam dokumen tersebut tidak jatuh kepihak yang tidak berkepentingan.

2. METODE PENELITIAN

Pada penelitian ini dibuat aplikasi kriptografi dokumen dengan menggunakan algoritma AES-256 berbasis *web*, secara garis besar tahapan penelitian dapat dijelaskan sebagai berikut:

2.1 Bahan Penelitian

Bahan penelitian diambil dari sumber buku, jurnal, artikel ilmiah dan sumber lain mengenai sistem kriptografi pendukung lain dalam pembangunan sistem.

2.1.1 Dokumen

Dokumen merupakan salah satu elemen terpenting dalam kegiatan administrasi perkantoran. Dokumen (berasal dari bahasa Latin: *documentum*) atau *sahifah* adalah sebuah tulisan penting yang memuat informasi, sedangkan menurut kamus besar bahasa Indonesia dokumen diartikan sebagai surat yang tertulis atau tercetak yang dapat dipergunakan sebagai bukti atau sebuah keterangan. Informasi yang berbentuk surat dan tertulis di instansi Polri disebut naskah dinas. Naskah Dinas digunakan sebagai alat komunikasi kedinasan yang dibuat dan atau dikeluarkan oleh pejabat yang berwenang di lingkungan Polri[3].

2.1.2 Kriptografi

Kriptografi di Kepolisian adalah kegiatan penyandian untuk pengamanan berita dan informasi rahasia mulai tingkat Markas Besar Kepolisian sampai dengan kewilayahan yang dilaksanakan dengan menerapkan konsep, teori dan seni dari ilmu kripto beserta ilmu pendukung lainnya secara sistematis, metodologis dan konsisten serta terikat pada etika profesi sandi. Direktorat Intelejen Keamanan yang selanjutnya disebut "Ditintelkam" adalah unsur pelaksana tugas pokok fungsi Intelijen tingkat Polda yang berada di bawah Kapolda dengan tugas pokok fungsi diantaranya sebagai penyelenggara intelejen bidang keamanan termasuk persandian[4].

Kata kriptografi berasal dari bahasa Yunani yang terdiri dari dua kata yaitu *crypto* yang berarti *secret* (rahasia) dan *graphia* yang berarti *writing* (tulisan). Kriptografi adalah ilmu yang mempelajari bagaimana cara menyembunyikan pesan. Dalam penerapannya, kriptografi merupakan suatu metode enkripsi atau penyandian data yang hanya diketahui atau mempunyai arti bagi suatu kelompok pengguna tertentu. Beberapa istilah yang sering digunakan dalam kriptografi adalah sebagai berikut[5],[6]:

1. *Plaintext* adalah teks terang atau informasi asli sebelum dienkripsi.
2. *Enkripsi* adalah proses kriptografi dari *plaintext* menjadi *ciphertext*.
3. *Ciphertext* merupakan hasil dari proses enkripsi atau informasi acak yang berasal dari *plaintext* yang telah dilakukan kriptografi.
4. *Dekripsi* adalah kebalikan dari enkripsi yaitu proses mengembalikan *ciphertext* menjadi *plaintext*.
5. *Kriptoanalisis* adalah ilmu yang mempelajari teknik matematika untuk memecahkan kriptografi.
6. *Kriptoanalis* adalah orang yang melakukan kriptografi.
7. *Kriptologi* adalah ilmu tentang kriptografi dan kriptonalisis.

2.1.3 Advanced Encryption Standard

Algoritma *Advanced Encryption Standard (*AES) adalah suatu algoritma *block chiper* dan mempunyai sifat simetri yang menggunakan kunci simetri pada waktu proses enkripsi dan dekripsi. Pada tahun 2001, AES digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST (National Institute of Standard and Technology) sebagai pengganti algoritma DES (Data Encryption Standard) yang sudah berakhir masa penggunaannya. Penyandian AES menggunakan proses yang berulang yang disebut dengan *ronde*. Jumlah *ronde* yang digunakan oleh AES tergantung dengan panjang kunci yang digunakan. Setiap *ronde*

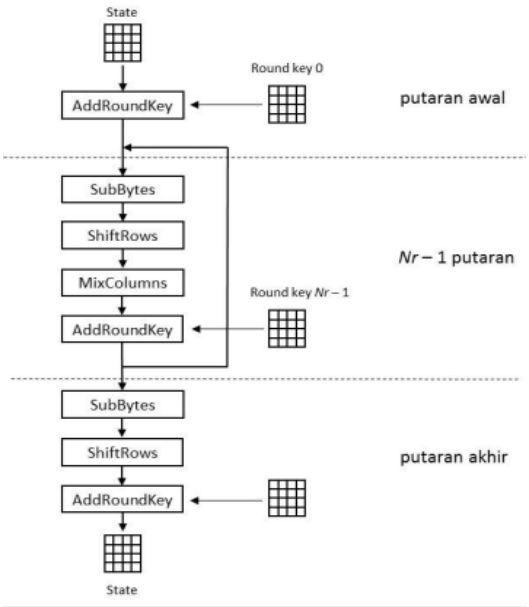
membutuhkan kunci *ronde* dan masukan dari *ronde* berikutnya. Kunci *ronde* dibangkitkan berdasarkan kunci yang diberikan. Algoritma AES dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit. Panjang kunci mempengaruhi jumlah *round* (perputaran), perbedaan dari ketiga kunci tersebut dapat digambarkan dalam Tabel 1 Perbandingan Jumlah Kunci AES [5],[6].

Tabel 1. Perbandingan Jumlah Kunci AES

	Kunci (*NK Words*)	**Block (*Nb Words*)**	**Putaran (*Nr*)**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Proses enkripsi terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, input yang telah dicopikan ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak *Nr*. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns.* Sedangkan pada proses dekripsi algoritma yang digunakan merupakan kebalikan dari algoritma enkripsi AES. Algoritma dekripsi AES menggunakan transformasi *invers* pada semua transformasi dasar yang digunakan pada algoritma enkripsi AES.

Proses enkripsi dapat dirujuk pada Gambar 1 Diagram Proses Enkripsi[5],[6].



Gambar 1. Diagram Proses Enkripsi

Algoritma AES mengambil kunci *cipher* dan melakukan ekspansi kunci (*key expansion*) untuk membentuk *key schedule*. Ekspansi kunci menghasilkan total Nb(Nr+1) *word*. Algoritma ini membutuhkan set awal *key* yang terdiri dari Nb *word*, dan setiap *round* Nr membutuhkan data kunci sebanyak Nb *word*. Hasil *key schedule* terdiri dari *array* 4 *byte word linear* yang dinotasikan dengan $[w_i]$. *SubWord* adalah fungsi yang mengambil 4 *byte word* input dan mengaplikasikan S-Box ke tiap-tiap data 4 *byte* untuk menghasilkan *word* output. Fungsi *RotWord* mengambil *word* $[a_0, a_1, a_2, a_3]$ sebagai input, melakukan permutasi siklik, dan mengembalikan *word* $[a_0, a_1, a_2, a_3]$. *Rcon*[i] terdiri dari nilai-nilai yang diberikan oleh $[x^{i-1}, \{00\}, \{00\}, \{00\}]$, dengan x^{i-1} sebagai pangkat dari x (x dinotasikan sebagai $\{02\}$) [5],[6].

2.2 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data-data yang dibutuhkan dalam perancangan sistem, adapun beberapa pendekatan yang dilakukan antara lain:

2.1.1 Observasi

Observasi dilaksanakan di Ditintelkam (Direktorat Intelejen dan Keamanan) Polda DIY untuk mengetahui keadaan objek penelitian yang sebenarnya. Hal ini bertujuan untuk memperoleh penjelasan mengenai data-data dan informasi yang dibutuhkan dalam penelitian.

2.1.2 Wawancara

Wawancara dilakukan terhadap narasumber dengan cara tanya jawab secara langsung dengan petugas sandi untuk mengetahui bagaimana proses kriptografi yang sudah dilakukan.

2.1.3 Studi Literatur

Mengkaji beberapa penelitian sejenis yang membahas mengenai perancangan aplikasi kriptografi sejenis yang telah dibuat sebelumnya sebagai acuan dalam pengembangan aplikasi. Berikut merupakan penelitian sejenis yang menjadi acuan penelitian:

1. Penelitian yang berjudul "Implementasi Algoritma *Advanced Encryption Standard* (AES) 128 Untuk Enkripsi dan Dekripsi *file* Dokumen" menyatakan bahwa keamanan data atau informasi adalah hal yang sangat penting bagi pengguna jaringan *internet*, penyadapan pesan atau informasi merupakan salah satu hal yang sangat merugikan maka perlu adanya peningkatan keamanan dalam pertukaran informasi. Pada penelitian ini akan mengimplementasikan algoritma AES-128 untuk enkripsi dan dekripsi data yang berupa *file* dokumen (PDF, DOC, TXT). Hasil uji coba enkripsi dan dekripsi *file* relatif lebih cepat dibanding dengan penelitian sebelumnya (Yulius

Rio, 2016). Proses enkripsi untuk 7,1 MB dibutuhkan waktu 3,3 detik, dibandingkan dengan 1,8 MB dibutuhkan waktu 1 detik. Proses dekripsi untuk 7,2 MB dibutuhkan waktu 2.5 detik dibandingkan dengan 1,8 MB dibutuhkan waktu 0.4 detik[7].

2. Penelitian yang berjudul "Pengamanan *Source Code* Program Menggunakan Algoritma *Advanced Encryption Standard* Dan Algoritma *Base* 64" menyatakan bahwa sisi negatif dari perkembangan teknologi adalah terdapat berbagai ancaman kejahatan yang dapat menyerang ketika proses pengembangan perangkat lunak, diantaranya adalah pencurian data dan manipulasi data perangkat lunak yang sedang dikembangkan. Upaya untuk mengurangi resiko ancaman yaitu dengan menerapkan kriptografi. Penelitian ini tentang pengamanan data *source code* menggunakan kriptografi simetris algoritma *Advanced Encryption Standard* (AES) yang umum digunakan untuk mengenkripsi sebuah teks dan *Base* 64 untuk pembentukan masukan kunci publik. Penelitian ini mengindikasikan peningkatan sebesar 54,03% pada 373 *sample* data proyek yang diuji dikarenakan *avalanche effect*[8].
3. Penelitian yang berjudul "Teknik Pengamanan *File* Dokumen Berbasis *Text* Menggunakan Metode *Advanced Encryption Standard* (AES)" menyatakan bahwa hal terpenting dalam komunikasi menggunakan komputer dan jaringan komputer adalah keamanan pesan, data, ataupun informasi dalam proses pertukaran data, sehingga menjadi salah satu pendorong munculnya teknologi Kriptografi. Penelitian bertujuan untuk merancang Aplikasi Kriptografi AES untuk Keamanan *File* Dokumen baik itu secara *online* maupun *offline*. Dari hasil uji coba dinyatakan bahwa perancangan aplikasi kriptografi berhasil dilakukan dan diimplementasikan untuk mengamankan *file* dokumen[9].
4. Penelitian yang berjudul "Rekayasa Perangkat Lunak Kriptografi Menggunakan Algoritma *Advanced Encryption Standard* (AES) 128 *Bit* Pada Sistem Keamanan *Short Message Service* (SMS) Berbasis *Android"* menyatakan bahwa *sms* rentan terhadap risiko pencurian informasi, karena sering pesan yang dikirimkan adalah pesan yang bersifat pribadi dan rahasia. Maka, dapat disimpulkan bahwa diperlukan *system* kriptografi yang diterapkan di *android*, guna mengamankan *sms* yang dikirim kepada penerima, tanpa rasa takut informasi akan bocor. Algoritma *Edvanced Encryption Standard* (AES) berjalan selama pengiriman pesan di *android* dapat menjadi satu jalan keluar untuk menyelesaikan masalah di atas, aplikasi ini memiliki tampilan yang sederhana dan menarik, karena itu aplikasi yang mudah digunakan dan tempat yang kokoh dalam solusi pertukaran informasi melalui *sms*[10].
5. Penelitian yang berjudul "Analisis Algoritma Pada Proses Enkripsi dan Dekripsi File Menggunakan Advances Encryption Standart (AES)" menyatakan bahwa dalam penggunaan teknologi, manusia tak pernah lepas dari kebutuhan akan sebuah informasi. Beberapa informasi dapat berupa *file* gambar, dokumen, dan video. Beberapa informasi memiliki privasi yang tidak boleh tersebar oleh *public*, oleh karena itu diperlukan cara dalam mengamankan informasi, salah satunya adalah menggunakan metode kriptografi. Dari hasil uji coba yang dilakukan pada proses enkripsi dan dekripsi disimpulkan bahwa *file* yang melalui uji coba dekripsi akan berubah bentuk menjadi *file* yang tak bias dibaca, *file* dapat kembali kebentuk asli jika melalui proses dekripsi dengan menggunakan kunci yang sama saat enkripsi. Dan waktu proses hasil enkripsi-dekripsi data dapat dipengaruhi oleh besar ukuran data yang akan di uji[11].
6. Penelitian yang berjudul "Implementasi Algoritma Advanced Encryption Standart (AES) Pada Enkripsi Dan Dekripsi QR-Code" menyatakan bahwa Keamanan data merupakan masalah yang sangat penting dalam perkembangan teknologi saat ini. Oleh sebab itu dibutuhkan sebuah cara yang dapat menjaga keamanan merujuk pada perlindungan informasi dari penyingkapan pihak yang tidak sah. Salah satu mekanisme untuk meningkatkan keamanan data adalah dengan menggunakan teknik kriptografi. AES diimplementasikan dalam bahasa pemrograman PHP dan diterapkan pada QR *Code* karena merupakan sebuah teknologi *labelling* yang dapat menyimpan data dalam bentuk pola yang dapat diisi dengan informasi. Dari hasil implementasi algoritme AES dapat disimpulkan bahwa aplikasi ini dapat mengekripsi semua jenis karakter berupa *string*, huruf, angka, dan simbol. Pada saat mendekripsi QR *Code* aplikasi akan mengaktifkan fungsi kamera dan melakukan *scanning QR Code* yang akan menjadi *plaintext* kembali. Waktu eksekusi enkripsi dan dekripsi AES adalah 0.0034 detik untuk proses enkripsi dan untuk proses dekripsi membutuhkan waktu 0.0029 detik[12].

2.3 Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah model pendekatan RAD (*Rapid Application Development*), berikut ini adalah tahap-tahap pengembangan aplikasi[13] .

2.3.1 Fase Perencanaan

Dari hasil observasi dan wawancara, analisis kebutuhan masalah yang ditemukan adalah

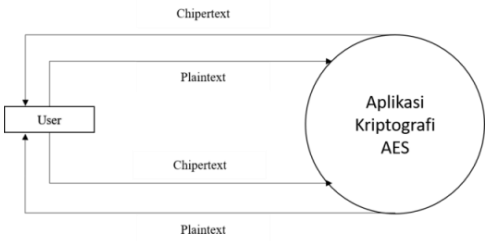
keamanan dokumen internal satuan kerja belum terkelola keamanannya, dari analisis tersebut maka tujuan dari penelitian ini adalah untuk membuat suatu aplikasi enkripsi yang mengimplementasikan metode *Advance Encryption Standard* berbasis *client sever* dan *user friendly* sehingga bisa menjadi alternatif dalam menjaga kerahasiaan dokumen rahasia internal di Ditintelkam Polda DIY.

2.3.2 Fase Perancangan

Dalam fase ini dilakukan perancangan proses, perancangan database dan perancangan interface.

2.3.2.1 Perancangan Proses

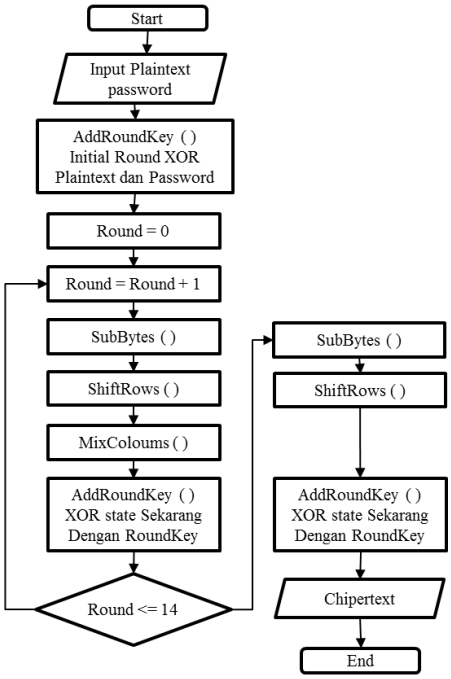
Pada proses enkripsi, *file* asli atau *plaintext* dienkripsi menggunakan metode enkripsi *Advance Encryption Standard* (AES) untuk menghasilkan *file* yang tidak bisa dibaca atau *chipertext* sedangkan pada proses dekripsi, *chipertext* yang dihasilkan dari proses enkripsi dikembalikan menjadi *file* semula dengan cara membalik proses enkripsi. Alur proses tersebut dapat dilihat pada Gambar 2 Diagram konteks sistem.



Gambar 2. Diagram Konteks Sistem

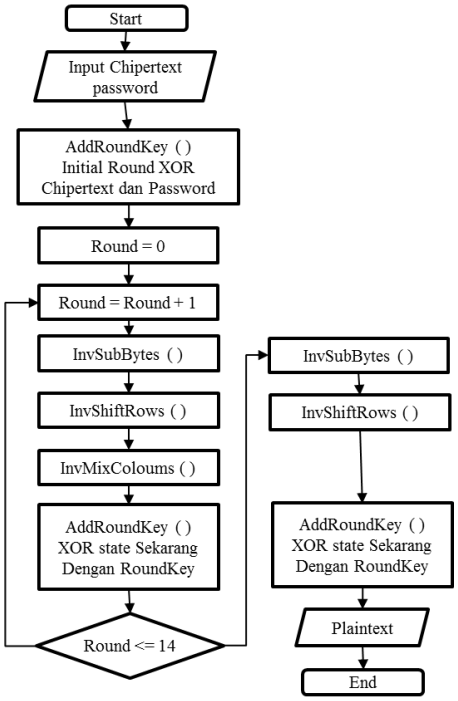
Dalam diagram konteks pada gambar 2, terdapat dua *entity* yang menunjang proses aplikasi yaitu *user* dan *file*. *User* terdiri dari admin dan petugas yang ditunjuk sebagai operator sistem pada bagian atau unit yang menggunakan aplikasi sedangkan *file* merupakan dokumen dengan ekstensi .doc, .docx, .xls, .xlsx, .pdf dan .txt.

Flowchart proses enkripsi yang dirancang dapat dilihat pada Gambar 3. Berdasarkan gambar 3, proses enkripsi dimulai dengan memasukkan *file* yang akan dienkripsi. *File* akan ditampilkan sementara, apabila *file* tidak sesuai maka *user* harus memasukkan kembali *file* yang benar. Selanjutnya *user* harus memasukkan *string* yang akan digunakan sebagai kunci untuk enkripsi *file*. *String* dapat berupa karakter apapun dengan panjang minimal 8 karakter, jika *password* kurang dari 8 karakter maka *user* harus memasukkan kembali *password* dengan panjang minimal 8 karakter. Jika inputan *file* dan *password* benar, proses enkripsi dimulai dengan memanggil fungsi enkripsi *Advance Encryption Standard*. Pada proses awal atau *initial round* dilakukan transformasi *AddRoundKey*() terhadap *plaintext* dengan inputan *password*. Selanjutnya dilakukan proses transformasi *SubBytes*(), *ShiftRows*(), *MixColumns*(), dan *AddRoundKey*() antara *state* sekarang dengan R*oundKey* hasil *KeyExpansion*() secara berulang dari R*ound-1* sampai *Round*-13, pada putaran terakhir atau *round*-14 dilakukan proses yang sama seperti *Round* sebelumnya, hanya pada putaran ini tidak dilakukan transformasi *MixColumns*(). Setelah selesai, proses selanjutnya adalah menyimpan *file* ke dalam *database*. Proses enkripsi selesai dan data *file* terenkripsi ditampilkan.



Gambar 3. *Flowchart* Proses Enkripsi

Flowchart proses enkripsi yang dirancang dapat dilihat pada Gambar 4.

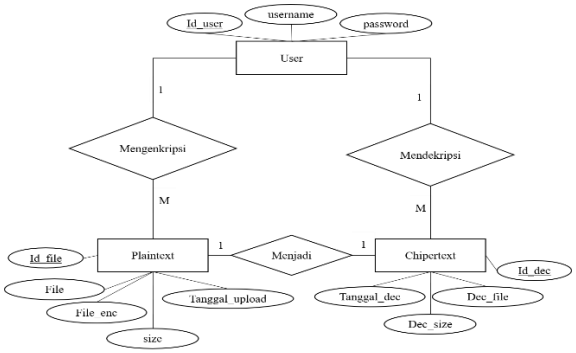


Gambar 4. *Flowchart* Proses Dekripsi

Berdasarkan Gambar 4, proses dekripsi dimulai dengan membaca *file* dan *password* dan proses selanjutnya sama seperti pada proses enkripsi, tetapi pada proses dekripsi yang dilakukan adalah transformasi *invers*.

2.3.2.2 Perancangan Database

Aplikasi ini dirancang menggunakan *database* MySQL[15]. Struktur *Entity Relationship Diagram* (ERD) dapat dilihat pada Gambar 5.



Gambar 5. Struktur *Entity Relationship Diagram*

2.3.2.3 Perancangan Interface

Interface atau antarmuka dirancang untuk memberikan fasilitas komunikasi antara pemakai dengan sistem yang bertujuan membantu mengarahkan alur penelusuran masalah sampai ditemukan solusi.

2.3.3 Fase Konstruksi

Pada tahap ini dilakukan penyusunan program. Rancangan-rancangan program yang telah didefinisikan disusun menggunakan bahasa pemrograman PHP[16], *Source code* ditulis menggunakan *text editor* Notepad ++ dan memanfaatkan *library* algoritma enkripsi *Advance Encryption Standard* (Rijndael) *mode counter* (CTR)[16].

2.3.4 Fase Pelaksanaan (Implementasi)

Pengujian aplikasi Akrid dilakukan menggunakan metode *blackbox* dengan menjalankan semua fungsi yang ada, kemudian dilihat apakah fungsi-fungsi tersebut berjalan sesuai dengan yang diharapkan. Aplikasi dijalankan melalui *web browser* untuk melakukan proses enkripsi dan dekripsi.

3. HASIL DAN PEMBAHASAN

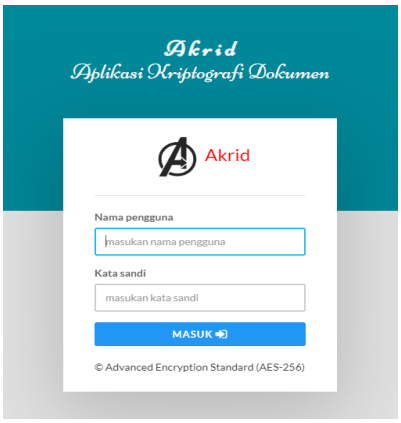
Aplikasi yang dirancang adalah sebuah aplikasi kriptografi menggunakan metode *Advanced Encryption Standard* (AES-256) berbasis *client server*.

Pada tahap implementasi, penulis melakukan simulasi konfigurasi *client server* sehingga *user* dapat langsung menjalankan aplikasi di komputer masing-masing. Dalam aplikasi ini pengguna bisa berbagi/mengirimkan *file* dokumen yang telah terenkripsi dengan *user client* lain maupun dengan *server client* melalui *tools share* yang terdapat pada menu dekripsi berkas tanpa harus *download file* dokumen hasil enkripsi terlebih dahulu.

Pengujian proses enkripsi dan dekripsi dilakukan menggunakan dokumen dengan ekstensi docx, doc, xls, xlsx, pdf dan txt dengan ukuran *file* maksimal 2 MB.

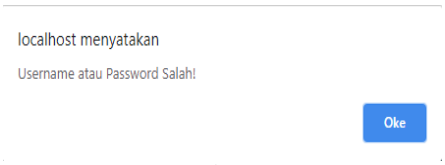
3.1 Tampilan Utama Aplikasi

Tampilan halaman pertama yang muncul dapat dilihat pada gambar 6. Pada halaman *login*, *user* diminta untuk menginputkan *username* dan *password* supaya bisa menggunakan aplikasi.



Gambar 6. Halaman *Login* Aplikasi

Jika user salah dalam menginputkan *username* dan *password login* maka akan ditampilkan notifikasi seperti pada Gambar 7.

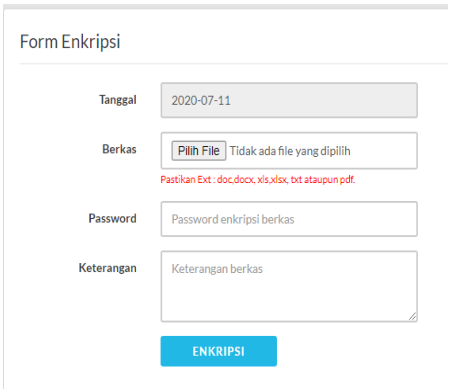


Gambar 7. Notifikasi *Login* Gagal

3.2 Proses Enkripsi

Tampilan *form* enkripsi seperti pada Gambar 8, *user* diminta untuk *upload file* dokumen dengan format doc, docx, xls, xlsx, pdf atau txt dan *password* untuk enkripsi.

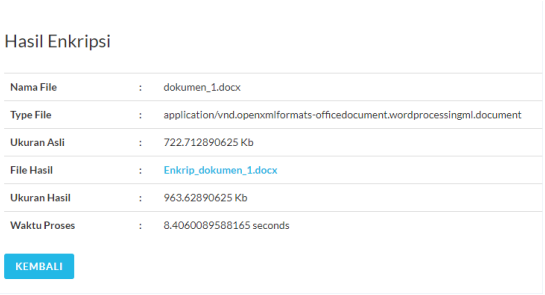
Pada Gambar 8, jika inputan *file* maupun *password* kosong maka akan muncul notifikasi untuk *upload file* dan *password*. Jika *file* yang diupload tidak sesuai dengan format yang disyaratkan atau *password* kurang dari 8 karakter maka akan muncul notifikasi seperti pada Gambar 9.



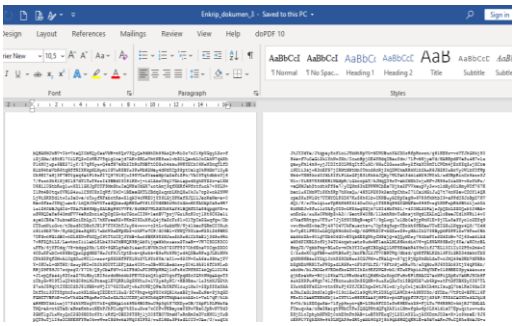
Gambar 8. *Form* enkripsi

Gambar 9. Notifikasi Inputan Tidak Sesuai

Jika inputan dari *user* benar, maka dapat dilihat informasi hasil proses enkripsi dokumen seperti pada Gambar 10 dan pada Gambar 11 bisa dilihat *file* hasil enkripsi.



Gambar 10. Hasil Enkripsi

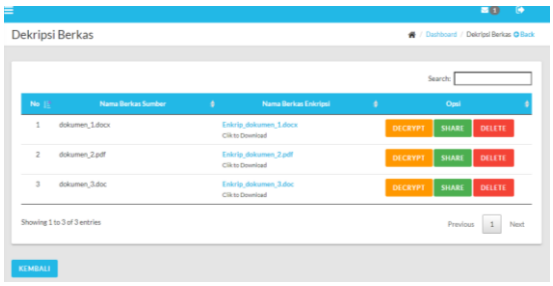


Gambar 11. File Hasil Enkripsi

3.3 Proses Dekripsi

Menu dekripsi menampilkan *file* yang telah dienkripsi, dalam menu ini *user* bisa mendekripsi *file* maupun *share file* kepada *user* lainnya, tampilan menu dekripsi dapat dilihat pada Gambar 12. Jika *user* memilih opsi "***Decrypt***" maka akan ditampilkan *form* dekripsi yang dapat dilihat seperti Gambar 13. Pada Gambar 13 user diminta inputan *password* dekripsi berkas, Jika inputan *password* tidak sesuai maka sistem akan menampilkan notifikasi seperti Gambar 14.

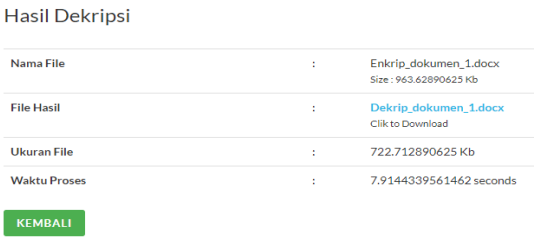
Hasil proses dekripsi dapat dilihat pada Gambar 15. Gambar 15 merupakan tampilan informasi hasil dekripsi *file*, *user* bisa langsung *download file* hasil dekripsi dengan klik nama *file* hasil dekripsi. *File* hasil proses dekripsi dapat dilihat pada Gambar 16.



Gambar 12. Menu Dekripsi

Gambar 13. *Form* Dekripsi

Gambar 14. Notifikasi *Password* Tidak Sesuai



Gambar 15. *Form* Hasil Dekripsi

Gambar 16. *File* Hasil Dekripsi

Pada gambar 16, dapat dijelaskan bahwa proses dekripsi berhasil dilakukan, format *file* kembali seperti semula dan dapat dibaca.

Jika pada Gambar 13 Menu Dekripsi *user* memilih opsi "***Share***" maka akan ditampilkan *form share file* seperti Gambar 17.

Dalam Gambar 18 dinyatakan bahwa sistem berhasil mengirimkan *file* kepada penerima yang dipilih pada *form share file*.

Gambar 17. *Form Share File*

Gambar 18. Notifikasi *Share File*

Perbandingan ukuran hasil dan waktu proses dapat dilihat pada Tabel 2. Pada Tabel 2, dokumen yang dilakukan ujicoba berhasil dienkripsi dan didekripsi. Ukuran dokumen hasil enkripsi bertambah sebesar 25 % dari ukuran semula, hal ini disebabkan karena adanya penambahan *byte* hasil pembentukan *key schedule* pada pembangkitan kunci (*key expansion*) yang nantinya akan digunakan sebagai pembangkitan kunci (*key expansion*) pada proses dekripsi dari dokumen itu sendiri, sedangkan pada proses dekripsi ukuran dokumen dapat kembali seperti ukuran dokumen semula. Waktu yang dibutuhkan enkripsi maupun dekripsi tidak ditentukan dari tipe dokumen tetapi dipengaruhi dari ukuran dokumen, dokumen dengan ukuran 19,212 Kb membutuhkan waktu enkripsi 0,212 *second* dan dekripsi 0,213 *second* atau 0,01 *second* per 1 Kb sedangkan dokumen dengan ukuran 1.966,9 Kb membutuhkan waktu enkripsi 20,553 *second* dan 20,882 *second* atau 0,01 *second* per 1 Kb.

Tabel 1. Perbandingan Ukuran dan Waktu Proses Enkripsi dan Dekripsi

No	Dokumen	Ukuran (Kb)			Selisih	%	Waktu (Second)			
		Asli	Enkripsi	Dekripsi			Enkripsi	Per 1 Kb	Dekripsi	Per 1 Kb
1	Dokumen_1.docx	722,722	963,629	722,722	240,907	25%	8,406	0,012	7,914	0,011
2	Dokumen_2.pdf	871,375	1.161,844	871,375	290,469	25%	12,445	0,014	10,855	0,012
3	Dokumen_3.doc	926,500	1.235,344	926,500	308,844	25%	11,342	0,012	10,522	0,011
4	Dokumen_4.txt	144,604	192,816	144,604	48,212	25%	1,814	0,013	1,790	0,012
5	Dokumen_5.xlsx	397,596	530,141	397,596	132,545	25%	4,506	0,011	4,511	0,011
6	Dokumen_6.xls	1.448	1.930,68	1.448	482,680	25%	17,617	0,012	15,361	0,011
7	Dokumen_7.rtf	107,755	143,684	107,755	35,929	25%	1,191	0,011	1,136	0,011
8	Dokumen_8.docx	19,212	25,629	19,212	6,417	25%	0,212	0,011	0,213	0,011
9	Dokumen_11.docx	1.966,892	2.622,535	1.966,892	655,643	25%	20,553	0,010	20,882	0,011
10	Dokumen_12.pdf	1.450,431	1.933,918	1.450,431	483,487	25%	15,155	0,010	15,413	0,011
11	Dokumen_13.xls	83	110,68	83	27,68	25%	0,866	0,010	0,892	0,011
12	Dokumen_14.pdf	33,053	44,082	33,053	11,029	25%	0,351	0,011	0,357	0,011
13	Dokumen_15.txt	8,935	11,926	8,935	2,991	25%	0,097	0,011	0,099	0,011

4. KESIMPULAN

Aplikasi Kriptografi Dokumen "**Akrid"** berhasil mengimplementasikan metode kriptografi *Advance Encryption Standard* (AES) dalam proses enkripsi dan dekripsi dokumen dengan format doc, docx, xls, xlsx, pdf dan txt. Berdasarkan hasil pengujian, ukuran dokumen berpengaruh dalam proses enkripsi dan dekripsi semakin besar ukuran dokumen, waktu yang dibutuhkan akan semakin lama. Pada proses enkripsi waktu yang dibutuhkan antara 0,010 – 0,014 *second* untuk dokumen per 1 Kb dan pada proses dekripsi membutuhkan waktu antara 0,011 – 0,013 *second* per 1 Kb.

DAFTAR PUSTAKA

[1] S. A. Yulianto. E, Keamanan Dalam Media Digital, Bandung: Informatika, 2020.

[2] S. Simarmata. J, Kriptografi Teknik Keamanan Data dan Informasi, Andi Offset, 2020.

[3] Peraturan Kepala Kepolisian Negara Republik Indonesia No 7 Tahun 2017 Tentang Naskah Dinas, 2017.

[4] Peraturan Kepala Kepolisian Negara Republik Indonesia Nomor 19 Tahun 2014 Tentang Penyelenggaraan Persandian Di Lingkungan Kepolisian Negara Republik Indonesia, 2014.

[5] R. Munir, 2018. [Online]. Available: https://informatika.stei.itb.ac.id/Pengantar Kriptografi. [Diakses 28 Maret 2020].

[6] R. Munir, 2018. [Online]. Available: https://informatika.stei.itb.ac.id/Advanced Encryption Standart (AES). [Diakses 28 Maret 2020].

[7] A. Prameshwari dan N. Sastra, "Implementasi Algoritma Advanced Encryption Standart (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," Explora Informatika, 2018.

[8] C. Saefudin, G. Abdilah dan A. Maspupah, "Pengamanan Source Code Program

Menggunakan Algoritma Advanced Encryption Standart dan Algoritma Base 64," Makalah Dipresentasikan pada Seminar Nasional Teknologi Informasi, 2019.

[9] Munawir, Zulfan, Y. Yanti dan Mudianto, "Teknik Pengamanan File Dokumen Berbasis Text Menggunakan Metode Advanced Encryption Standart (AES)," Seminar Nasional II USM 2017, vol. Vol. 1, pp. 87-90, 2017.

[10] A. Arif dan P. Mandani, "Rekayasa Perangkat Lunak Kriptografi Menggunakan Algoritma Advanced Encryption Standart (AES) 128 Bit Pada Sistem Keamanan Short Message Service (SMS) Berbasis Android," Jurnal Teknoif, vol. 4, no. 1, 2016.

[11] F. Muharram, H. Aziz dan A. Manga, "Analisis Algoritma Pada Proses Enkripsi dan Dekripsi File Menggunakan Advances Encryption Standart (AES)," Prossiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi, vol. 3, no. 2, Desember 2018.

[12] D. Q. P. A. Paramarta, A. Kusyanti dan M. Data, "Implementasi Algoritma Advanced Encryption Standart (AES) Pada Enkripsi Dan Dekripsi QR-Code," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 2, no. 12, pp. 6728-6736, 2018.

[13] F. Sulianta, Strategi Merancang Arsitektur Sistem Informasi Masa Kini, Elex Media Komputindo, 2019.

[14] C. Vennes, AES In PHP, 2005-2014. [Online]. Available: https://www.movable-type.co.uk. [Diakses 23 Maret 2020].

[15] D. Setiyadi, Sistem Basis Data dan SQL, Mitra Wacana Media, 2020.

[16] J. Enterprice, PHP Untuk Programer Pemula, Elex Media Komputindo, 2019.