PENINGKATAN PENGAMANAN DATA FILE MENGGUNAKAN ALGORITMA KRIPTOGRAFI AES DARI SERANGAN BRUTE FORCE

Indra Gunawan
STIKOM Tunas Bangsa Pematangsiantar
indra@amiktunasbangsa.ac.id

Abstrak

Abstrak— Data merupakan suatu komponen yang dapat dikumpulkan menjadi satu kesatuan dan dikelola, sehingga dapat dijadikan sebuah informasi yang bermanfaat. Setiap informasi yang sudah dikumpulkan nantinya akan disimpan kedalam bentuk digital berupa file-file yang kemudian dapat dikelola sedemikian rupa sebelum file tersebut dipublikasikan. Dengan penggunaan informasi kedalam bentuk file, dapat menghemat dari pada penggunaan jenis kertas. Akan tetapi seiring dengan perkembangan zaman, penggunaan dari pada file-file dalam bentuk digital memiliki tingkat keamanan yang mulai riskan karena sering terjadi pembobolan data dari pihak-pihak yang tidak memiliki hak untuk penggunaan data. Beberapa metode yang digunakan untuk pembobolan data adalah dengan menggunakan metode brute force. Maka untuk dapat mengantisipasi dari pembobolan data tersebut, diperlukan beberapa metode yang dapat digunakan untuk mengamankan data yang diantaranya menggunakan ilmu kriptografi dengan metode AES (*Advanced Encryption Standard*). Dengan penggunaan metode ini diharapkan sedikitnya dapat meningkatkan keamanan data file yang digunakan dan disimpan didalam arsip digital.

Kata Kunci : Kriptografi, Pembobolan, Data, Brute Force

1. Pendahuluan

Sebuah teknologi didalam dunia ilmu komputer selalu berkembang disetiap waktu, sehingga dapat memberi kemudahan kepada pengguna untuk menggunakan fasilitas tersebut kapan saja

DOI: https://doi.org/10.29103/techsi.v13i1.2395

dan dimana saja. Didalam penggunaan data, bisa digunakan dengan cara yang sangat sederhana karena data yang digunakan sudah bertipe digital dan tidak harus menggunakan data yang bertipe arsip. Data digital paling sering digunakan oleh pengguna pada saat ini dikarenakan sangat efisiensi dan dapat dibawa serta digunakan dimana saja. Hal ini dikarenakan kapasitas dari pada data digital yang sangat ramping dan dapat disimpan di sebuah media penyimpanan seperti HDD, *Flashdisk*, *Memory Card* hingga ke *Cloud Drive*.

Karena terlalu sederhananya penggunaan data dengan memanfaatkan media penyimpanan seperti *Cloud Drive*, membuat pengguna tidak begitu khawatir dalam melakukan penyimpanan data digital mereka, karen data tersebut dapat diakses kapan saja dan dimana saja melalui konektivitas internet [1].

Dengan penggunaan data dalam bentuk file, sering kali terjadi penyerangan dan pembobolan data. Apabila data-data yang tersimpan itu terasa penting, pasti ada saja pihak-pihak yang tidak memiliki hak untuk melakukan pencurian data. Dalam melakukan pencurian data biasanya banyak cara yang bisa dilakukan, diantaranya dengan menggunakan algoritma *Brute Force*.

Algoritma *Brute Force* dapat melakukan serangkaian serangan dengan menggunakan penerkaan kombinasi kunci yang sangat sederhana serta melakukan pembajakan dan pencarian kode secara acak dengan cara yang jelas dan lempang [2]. Hal ini dapat dilakukan terhadap pencarian data yang apabila data tersebut memiliki isi yang sangat penting, bisa saja ada pihak-pihak tertentu untuk melakukan pembobolan data dan mengambil isi data tersebut. Dengan kata lain, setiap data yang memiliki isi yang sangat penting bila data tersebut tidak diproteksi, maka kemungkinan yang sangat besar akan dibobol atau deretas oleh pihak-pihak tertentu.

Maka dari pada itu dibutuhkan beberapa proteksi yang dapat memberikan keamanan data terhadap file data. Algoritma dari beberapa ilmu kriptografi dapat digunakan untuk melakukan sebuah proteksi terhadap file data. Beberapa algoritma kriptografi itu diantaranya algoritma kriptografi AES (*Advanced Encryption Standart*). Algoritma kriptografi AES dapat digunakan untuk

DOI: https://doi.org/10.29103/techsi.v13i1.2395

melakukan penyisipan keamanan data di berbagai jenis pesan seperti *image cover*, *Text* dan Multimedia[3].

Algoritma kriptografi AES dapat memberikan keamanan terhadap banyak jenis data tertutama data yang bersifat dokumen seperti *word, pdf, excel, power point* dan *text*. Dengan menggunakan algoritma kriptografi AES, keamanan data bisa ditingkatkan kepada jumlah bit yang tinggi seperti 64, 128, 256 hingga 512 bit. Hal ini bertujuan untuk menjaga originalitas dari pada isi data yang akan diamankan agar tidak gampang untuk dibajak dan dimodifikasi oleh pihak-pihak yang tidak memiliki wewenang terhadap data tersebut[4].

Selain melakukan pengamana file, kriptografi AES juga dapat melakukan pengamanan data terhadap isi data seperti pengamanan databse atau basis data. Seperti beberapa penelitian sudah meniliti untuk mengamankan database pada suatu aplikasi test masuk karyawan secara online dengan menggunakan algoritma AES-128. Algoritma ini diimplementasikan pada sistem ujian online untuk meminimalisir kemungkinan bocornya soal test seleksi pegawai oleh pihak yang tidak bertanggung jawab[5].

Data dan informasi merupakan suatu aset yang sangat penting bagi perusahaan ataupun individu. Setiap pengguna komputer selalu melakukan penyimpan data yang sudah digunakan untuk ". . meningkatkan keamanan data dari yang disimpan agar setiap informasi yang sudah diolah dapat terjaga kerahasiaannya . . [6]". Pada era teknologi sekarang ini, pengamanan dan keamanan data sangatlah penting dan harus selalu diperhatikan, hal ini bertujuan untuk menjaga keaslian dari pada data yang disimpan dan dikirim. Dengan kata lain mengamankan data dari serangan brute force itu sangatlah penting, karena algoritma brute force dapat melakukan pembobolan dan melakukan kombinasi untuk mencari dan menghasilkan data dari sebuah perangkat elektronik seperti komputer[7].

2. Metode Penelitian

Dalam pengimplementasian sistem ini dibutuhkan algoritma untuk menyelesaikan langkah-langkah instruksi yang digunakan dalam memproses enskripsi dan dekripsi data-data yang bersifat rahasia. Contoh perhitungan yang digunakan dalam proses

DOI: https://doi.org/10.29103/techsi.v13i1.2395

enkripsi algoritma AES dengan melakukan kombinasi plainteks dengan kunci dengan contoh sebagai berikut :

Plaintext : Toko R41 Siantar
Kunci : KriptoAES128-bit

Langkah yang harus diselesaikan dengan merubah bentuk plaintext dan kunci kedalam bentuk hexadesimal. Maka hasilnya adalah

Plaintext : 546f 6b6f 3a52 3431 5f53 4941 4e52 4152
Kunci : 4b72 6970 746f 4145 5331 3238 2d62 6974

Selanjutnya merubah kembali posisi plaintext dan kenci kedalam bentuk matriks dengan ordo 4x4

54	3A	5F	4E
6F	52	53	52
6B	34	49	41
6F	31	41	52

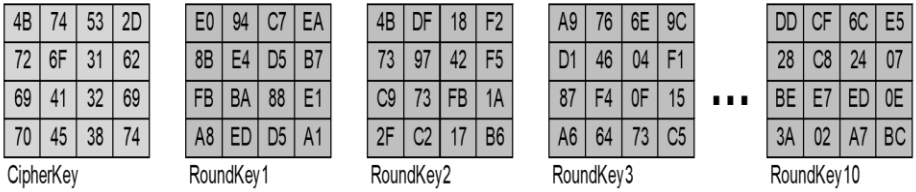
4B	74	53	2D
72	6F	31	62
69	41	32	69
70	45	38	74

Proses enkripsi algoritma AES akan melakukan 4 proses , yaitu *SubBytes, ShiftRows, MixColumn* dan *addArroundKey*. Tahapan-tahapan yang dilakukan seperti :

A. Melakukan Ekspansi Kunci

Algoritma AES menciptakann suatu ekspansi kunci untuk menghasilkan *key schedule*. Proses ekspansi dari 128 bit menjadi 1408 bit disebut dengan *key schedule*.

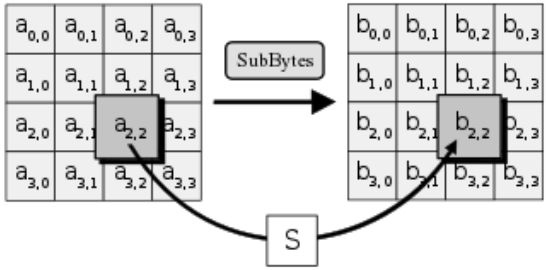
DOI: https://doi.org/10.29103/techsi.v13i1.2395



Gambar 1. Proses Ekspansi Kunci AES

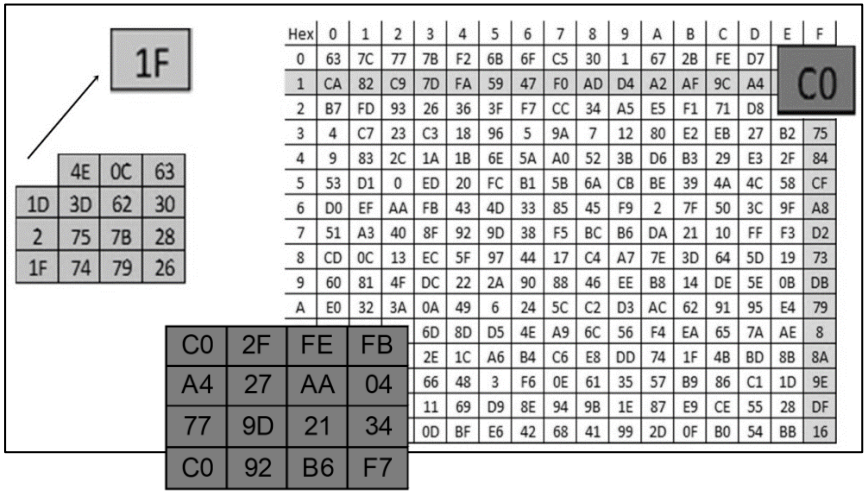
B. Melakukan Transformasi SubBytes

Pada proses ini, algoritma AES akan melakukan pemetaan setiap *byte* dari array yang digunakan dengan menggunakan bantuan tabel substitusi



Gambar 2. Proses Substitusi

Hasil keseluruhan dari transformasi *SubByte* seperti gambar berikut :

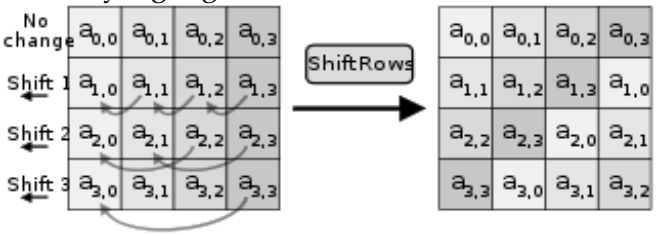


Gambar 3. Proses Transformasi *SubByte*

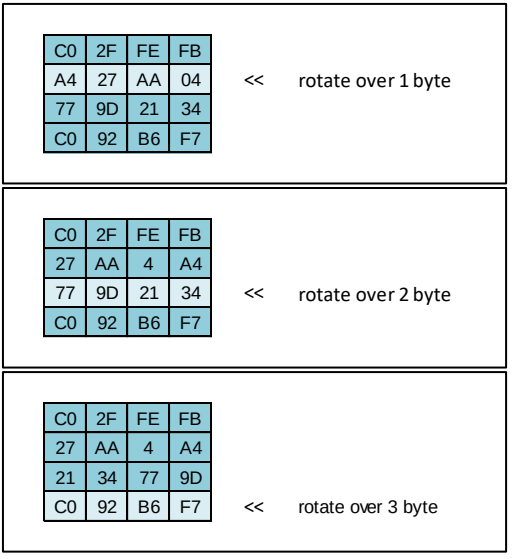
DOI: https://doi.org/10.29103/techsi.v13i1.2395

C. Melakukan Transformasi ShiftRow

Melakukan pergeseran secara *wrapping* dari masing-masing 3 baris terakhir jumlah *array*. Jumlah pergeseran tergantung dari pada nilai baris yang digunakan.



Gambar 4. Proses Pergeseran secara *wrapping*

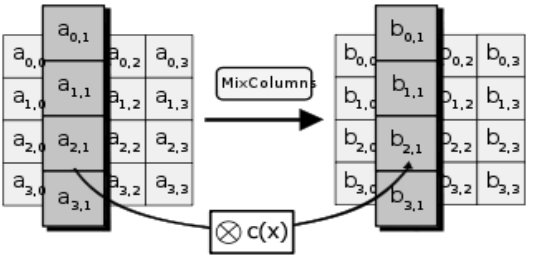


Gambar 5. Hasil Pergeseran secara *wrapping*

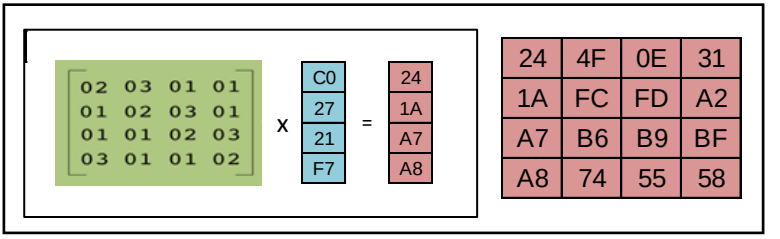
D. Melakukan Transformasi MixColumn

Untuk proses ini, melakukan perkalian di setiap kolom dari pada array dengan menggunakan perkalian matriks dan dapat dinyatakan seperti berikut

DOI: https://doi.org/10.29103/techsi.v13i1.2395



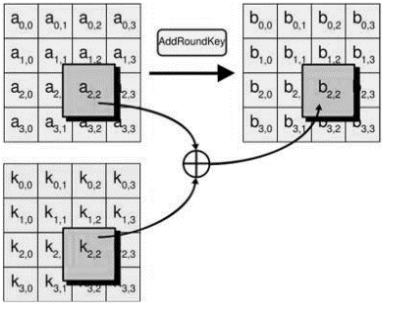
Gambar 6. Proses Perkalian Matriks



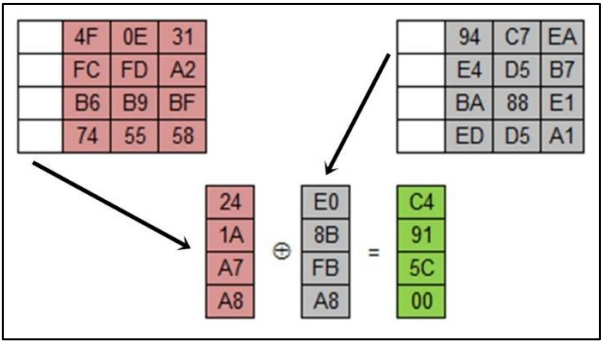
Gambar 7. Hasil Perkalian Matriks

E. Melakukan Transformasi AddRoundKey

Untuk proses ini, melakukan proses operasi XOR terhadap *arround key* dengan *array*, lalu hasilnya akan disimpan di *array state*.



Gambar 8. Proses Operasi XOR



C4	DB	C9	DB
91	18	28	15
5C	0C	31	5E
00	99	80	F9

Gambar 9. Hasil operasi XOR

3. Hasil dan Pembahasan

Hasil dan pembahasan ini dituangkan kedalam bentuk aplikasi agar dapat melakukan percepatan proses, baik itu proses perhitungan, proses enkripsi dan dekripsi.

Gambar 10. Tampilan Awal Aplikasi

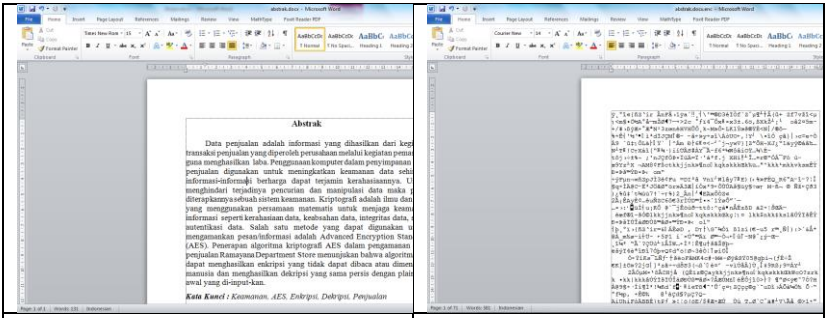
Pada gambar 10 merupakan tampilan awal dari pada aplikasi yang akan digunakan untuk melakukan pengamanan file. Terdapat tombol untuk melakukan proses enkripsi dan deskripsi. Tahapan yang pertama yaitu dengan menekan tombol Encryption Files, lalu

DOI: https://doi.org/10.29103/techsi.v13i1.2395

selanjutnya akan muncul form untuk melakukan pencarian file yang akan di enkripsi.

Gambar 11. Proses Enkripsi

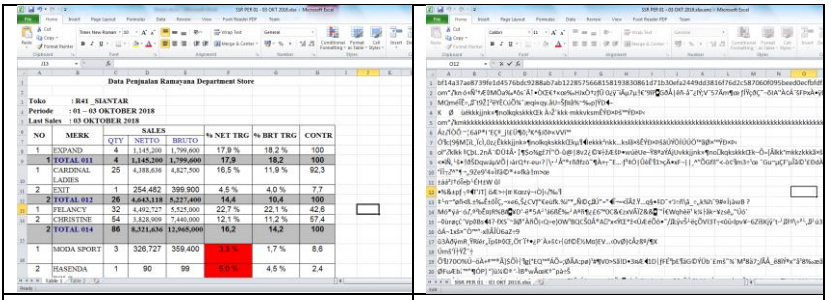
Dari gambar 10 melakukan prose enkripsi terhadap file yang akan diamankan dengan tahapan-tahapan yang dimulai dari 1. Tombol enkripsi file yang berfungsi untuk menampilkan form pencarian file. 2. Form pencarian file yang berfungsi untuk menelusuri file yang akan dienkripsi. 3. Form validasi untuk proses enkripsi terhadap file yang sudah dipilih.



Sebelum di enkripsi	Sesudah di enkripsi

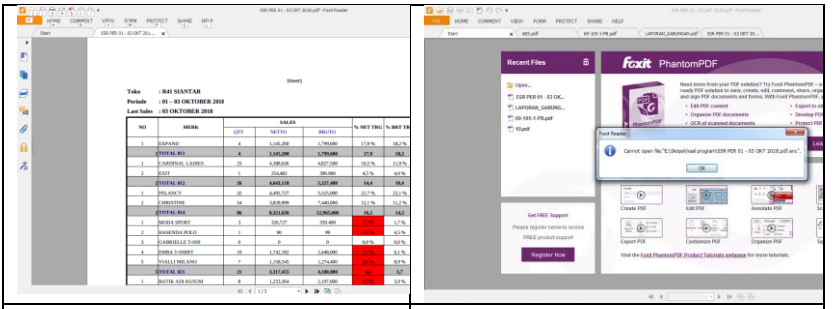
Gambar 11. Hasil proses enkripsi menggunakan file dokumen Word

DOI: https://doi.org/10.29103/techsi.v13i1.2395



Sebelum di enkripsi	Sesudah di enkripsi

Gambar 12. Hasil proses enkripsi menggunakan file dokumen Excel



Sebelum di enkripsi	Sesudah di enkripsi

Gambar 13. Hasil proses enkripsi menggunakan file dokumen Pdf

Sebelum di enkripsi	Sesudah di enkripsi

Gambar 14. Hasil proses enkripsi menggunakan file dokumen Jpg

Selanjutnya untuk proses dekripsi dilakukan dengan langkah yang hampir serupa dengan proses enkripsi, seperti gambar 15.

DOI: https://doi.org/10.29103/techsi.v13i1.2395

Gambar 15. Proses Dekripsi

Untuk langkah pertama dengan memilih tombol dekripsi file agar form pemilihan file tampil. Selanjutnya langkah ke 2 akan muncul form tampilan file yang berguna untuk menyeleksi/memilih file yang terenkripsi. Selanjutnya form yang ke-3 validasi proses dekripsi.

4. Kesimpulan

Kesimpulan yang dapat diambil dari pembahasan diatas adalah telah didapat suatu model baru yang berguna untuk melakukan pengamanan file data dan metode algoritma AES dapat bekerja dan berproses secara optimal dalam melakukan pengamanan file data.

Daftar Pustaka

Santiko, I., et al,. *Pemanfaatan Private Cloud Storage Sebagai Media Penyimpanan Data E-Learning Pada Lembaga Pendidikan*. Jurnal Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta, Vol. 10, No. 2, Tahun 2017, pp. 137-146.

Gunawan, I. *Penggunaan Brute Force Attack Dalam Penerapannya Pada Crypt-8 dan CSA-Rainbow Tool Untuk Mencari BISS*. Jurnal Nasional Informatika dan Teknologi Jaringan (InfoTekjar). Vol. 1, No. 1. September 2016, pp 52-56.

DOI: https://doi.org/10.29103/techsi.v13i1.2395

Yasfa, et al,. *Implementasi Proteksi Penyandian Pesan dengan Algoritma AES (Advanced Encryption Standard) untuk Penyisipan Pesan Berbasis Image Cover*. INSIGHT, Vol. 1, No. 3, Tahun 2018, pp. 240-245.

Setti, S. et al. *Implementasi Algoritma Advanced Encryption Standard dalam Pengamana Data Penjualan Ramayana Departement Store*. JURIKOM (Jurnal Riset Komputer), Vol. 7, No. 1. Februari 2020, pp 182-193.

Grehasen, G. dan Mulyati, S. *Pengamanan Database pada Aplikasi Test Masuk Karyawan Baru berbasis Web Menggunakan Algoritma Kriptografi AES-128 dan RC4*. BIT. Vol. 14, No. 1 Tahun 2017. pp. 52-60.

Saputra, R. *Desain Sistem Informasi Order Photo pada Creative Studio Foto dengan Menggunakan Bahasa Pemrograman Visual Basic .Net 2010*. Momentum, Vol. 17, No. 2, Tahun 2015. pp 86-93.

Gunawan, I. *Modifikasi Pengamanan File dengan Algoritma Hill Cipher Untuk Mengantisipasi dari Serangan Brute Force*. Jurnal TECHSI, Vo. 11, No. 1, April 2019. pp 237-246.

DOI: https://doi.org/10.29103/techsi.v13i1.2395