JURNAL KILAT
Vol. 7, No. 2, September 2018, P-ISSN 2089-1245, E-ISSN 2655-4925

ENKRIPSI DATA MENGGUNAKAN ADVANCED ENCRYPTION STANDARD 256

Yudi Wiharto[1], Ari Irawan[2]
[1] Program Studi Teknik Informatika, Universitas Budi Luhur, Jakarta, Indonesia
[2] Proram Studi Sistem Informasi, Universitas Tanri Abeng, Jakarta, Indonesia
[1]visited.mymail@gmail.com, [2]ari_irawan@tau.ac.id

ABSTRACT

Cryptography is important in securing data and information. Confidential, important information may not be publicly or otherwise protected. It is not impossible for anyone to see, damage, steal or misuse important data from an agency or company through a computer network. The solution is with cryptography or a method of data security that can maintain the confidentiality and authenticity of a data or information. This method is intended for confidential information when sent through network access, such as LAN or internet, cannot be utilized by unauthorized parties. Cryptography supports the aspect of information security, namely protection of confidentiality. Therefore the need to maintain the confidentiality of data and information is a cryptographic application. The process in the form of encryption and decryption used by the user to secure the data without changing the contents of the data. This application has a 32-character key but in its use is made into 2 keys, namely public and private key where the public key is the key filled by the user in accordance with the desire, while the private key is the default key entered by the application at random to meet the length of 32 characters. The AES algorithm used is the AES256 algorithm where this algorithm uses the principle with the number of rounds by key.

Keywords: *Cryptography, AES 256, Encryption, Decryption, Algorithm.*

ABSTRAK

Kriptografi penting dalam mengamankan data dan informasi. Informasi rahasia dan penting tidak boleh dilindungi secara publik atau sebaliknya. Bukan tidak mungkin bagi siapa pun untuk melihat, merusak, mencuri atau menyalahgunakan data penting dari agensi atau perusahaan melalui jaringan komputer. Solusinya adalah dengan kriptografi atau metode keamanan data yang dapat menjaga kerahasiaan dan keaslian data atau informasi. Metode ini ditujukan untuk informasi rahasia ketika dikirim melalui akses jaringan, seperti LAN atau internet, tidak dapat dimanfaatkan oleh pihak yang tidak berwenang. Kriptografi mendukung aspek keamanan informasi, yaitu perlindungan kerahasiaan. Oleh karena itu kebutuhan untuk menjaga kerahasiaan data dan informasi adalah aplikasi cryptographic. Prosesnya berupa enkripsi dan dekripsi yang digunakan oleh pengguna untuk mengamankan data tanpa mengubah isi data. Aplikasi ini memiliki kunci 32 karakter tetapi dalam penggunaannya dibuat menjadi 2 kunci, yaitu kunci publik dan privat di mana kunci publik adalah kunci yang diisi oleh pengguna sesuai dengan keinginan, sedangkan kunci privat adalah kunci default yang dimasukkan oleh aplikasi secara acak untuk memenuhi panjang 32 karakter. Algoritma AES yang digunakan adalah algoritma AES256 di mana algoritma ini menggunakan prinsip dengan jumlah putaran berdasarkan kunci.

Kata Kunci: *Kriptografi , AES 256, Enkripsi, Dekripsi, Algoritma.*

1. PENDAHULUAN

Teknologi komputer pada saat ini telah mengalami perkembangan yang sangat pesat, hingga merambah ke segala aspek menyangkut segala kebutuhan dalam pekerjaan manusia. Komputer membutuhkan suatu keamanan dalam penyimpanan data atau informasi adalah hal yang sangat penting dan tidak bisa diabaikan begitu saja. Semakin tinggi tingkatan teknologi komputer, maka akan semakin tinggi pula tingkat ancaman yang akan mengancam keamanan data didalam komputer. Kerahasiaan suatu file yang tersimpan pada komputer harus diberikan pengamanan dan sudah menjadi persyaratan mutlak yang sangat diperlukan untuk melindungi file tersebut terhadap berbagai ancaman seperti dapat dengan mudah seseorang melihat, merusak, mencuri ataupun menyalahgunakan data atau informasi penting dari suatu instansi atau perusahaan melalui jaringan komputer. Kriptografi bertujuan agar data atau informasi tidak dapat dibaca oleh orang yang tidak berhak. Algoritma AES adalah blok *chipertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi.

Dalam kriptografi ada istilah yang disebut dengan enkripsi (*encryption*) yaitu proses penyamaran data dari *plaintext* (data asli) menjadi *chiperterxt* (data tersandi) dan deskripsi (*decryption*) yaitu proses pemgembalian *chipertext* menjadi *plaintext* kembali.

2. METODE PENELITIAN

Untuk menyelesaikan penelitian ini diperlukan pengumpulan data yang berhubungan dengan masalah yang dibahas. Tujuannya sebagai sumber landasan pembahasan dan pembuatan rancangan sistem. Adapun metode penelitian yang digunakan dalam mengumpulkan data atau materi penulisan adalah dengan cara:

2.1 Pengamatan (observasi)

Pengamatan atau observasi merupakan salah satu teknik pengumpulan data/fakta yang cukup efektif untuk mempelajari suatu sistem. Pengamatan langsung ini dilakukan untuk mengetahui proses-proses yang sedang berjalan serta membuat keputusan yang menyangkut lingkungan fisiknya pada suatu kegiatan yang sedang berjalan.

2.2 Wawancara *(interview*)

Wawancara dilakukan untuk mendapatkan data dan informasi dalam bentuk tanya jawab kepada orang yang terlibat secara langsung yang merupakan obyek penelitian.

2.3 Studi Pustaka

Metode ini menggunakan dokumen sebagai sumber bacaan, baik buku-buku ilmiah maupun jurnal, terutama yang erat hubungannya dengan masalah yang di bahas dalam penelitian ini.

3. HASIL DAN PEMBAHASAN

Untuk mengimplementasikan enkripsi soal-soal tersebut diperlukan algoritma enkripsi agar soal-soal tersebut bisa di enkrip dan dikembalikan seperti semula atau dekrip tanpa mengalami perubahan. Dengan adanya aplikasi ini diharapkan isi dari soal-soal tersebut dapat disimpan dengan aman tanpa adanya permasalahan.

3.1 Kebutuhan Sistem

Pada dasarnya komputer merupakan sebuah sistem yang terdiri dari beberapa komponen yang saling berhubungan dan menghasilkan *input*, proses, *output* dan *storage*. Dengan kata lain komputer memerlukan beberapa komponen dan fungsi perangkat keras yang mendukung proses komputerisasi.

Spesifikasi sistem yang akan dibangun pada aplikasi ini adalah sebagai berikut :

a) Aplikasi harus dapat mendeteksi user yang akan *login*.
b) Aplikasi mampu mengacak informasi dengan cara mengenkripsi data dan mengembalikannya atau mendekripsikan data terebut seperti semula tanpa adanya perubahan dari data tersebut.
c) Proses enkripsi dan dekripsi harus dilakukan dengan cepat.

3.2 Skema Proses Sistem Aplikasi

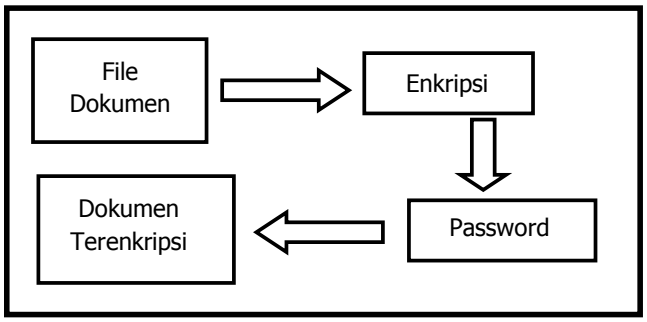
Tahapan-tahapan yang terjadi dalam skema proses aplikasi ini adalah sebagai berikut:

3.2.1 Skema Proses Enkripsi

Enkripsi merupakan proses pengacakan sebuah informasi atau data pada suatu aplikasi dengan menggunakan algoritma enkripsi, aplikasi yang kami buat menggunakan algoritma AES 256 yang menggunakan satu kunci.

Adapun langkah-langkah pada proses enkripsi simetris ini dapat diuraikan sebagai berikut:

(1) Menginputkan data atau informasi yang ingin dikunci.
(2) Menginputkan bit dengan bilangan angka yang digunakan sebagai kunci rahasia algoritma simetris.
(3) Melakukan proses enkripsi dengan algoritma AES 256 dan melakukan *upload* ke dalan server lokal.
(4) Menghasilkan *output* berupa *file chiper*. Dapat dilihat pada gambar 1.

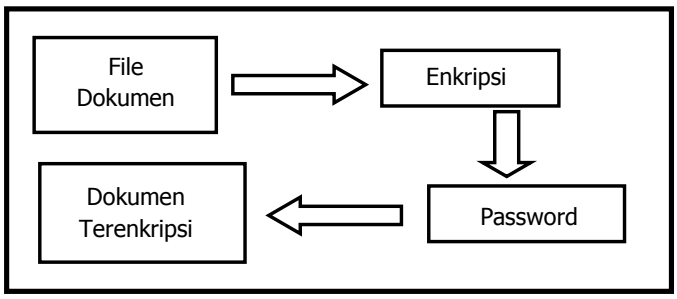


Gambar 1. Proses Enkripsi

3.2.2 Skema Proses Dekripsi

Dekripsi merupakan proses pengembalian atau pemulihan sebuah informasi atau data yang telah dienkripsi pada aplikasi ini, dengan menggunakan algoritma AES 256. Proses ini adalah dimana informasi atau data acak (*chiper*) yang terenkripsi menjadi data asli (*plain*), dengan algoritma simetris. Adapun langkah-langkah pada proses dekripsi ini dapat diuraikan sebagai berikut:

1. *File chiper* di *download* dari server lokal.
2. *File chiper* di*dekripsi*.
3. Mendekrip file tersebut dengan kunci
4. Menghasilkan *output* berupa file informasi atau data yang utuh dan dapat dibaca seperti awal. Dapat dilihat pada gambar 2.



Gambar 2. Proses Dekripsi

3.2.3 Algoritma Enkripsi

```
1.Input File dan Password
2.AddRoundKey() (Initial Round) XOR antara File dengan Password
3.Rounds = 0
4.Rounds = Rounds +1
5.SubBytes()
6.ShiftRows()
7.MixColumns()
8.AddRoundKey() XOR antara state sekarang dengan RoundKey
9.If Rounds <= 14
10.SubBytes()
12.ShiftRows()
13.AddRoundKey() XOR antara state sekarang dengan RoundKey
14.        else
15.         Kembali Ke baris 4
16.File terenkripsi
17.END
```

Gambar 3. Algoritma Enkripsi

3.2.4 Algoritma Dekripsi

```
1.Input File terenkripsi dan Password
2.AddRoundKey() (Initial Round) XOR antara Chipertext dan Password
3.Rounds = 0
4.Rounds = Rounds +1
5.InvShiftRows()
6.InvSubBytes()
7.AddRoundKey() XOR antara state sekarang dengan RoundKey
8.InvMixColumns()
9.If Rounds <= 14
10. InvShiftRows()
12. InvSubBytes()
13.AddRoundKey() XOR antara state sekarang dengan RoundKey
14.        else
15.         Kembali Ke baris 4
16.FileTerdekripsi
17.END
```

Gambar 4. Algoritma Dekripsi

3.2.5 Diagram Alir (*FlowChart*)

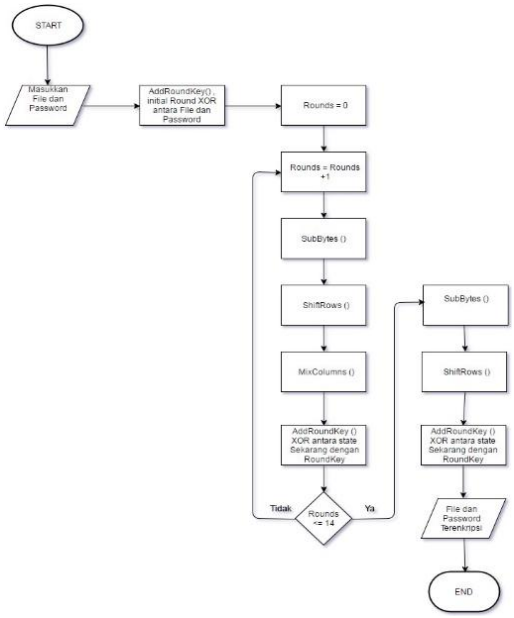
Flowchart merupakan diagram yang menunjukkan bagaimana cara kerja dari sebuah aplikasi untuk masuk pada program pada saat pertama kali dijalankan. Sesuai dengan rancangan layar, maka *flowchart* yang akan dibuat terdiri dari proses - proses sebagai berikut :

1) Enkripsi
   ini menjelaskan bagaimana proses *enkripsi*, mulai dari proses memilih data yang akan *enkripsi*, memberikan *password*. Flowchart dapat dilihat pada gambar 5.

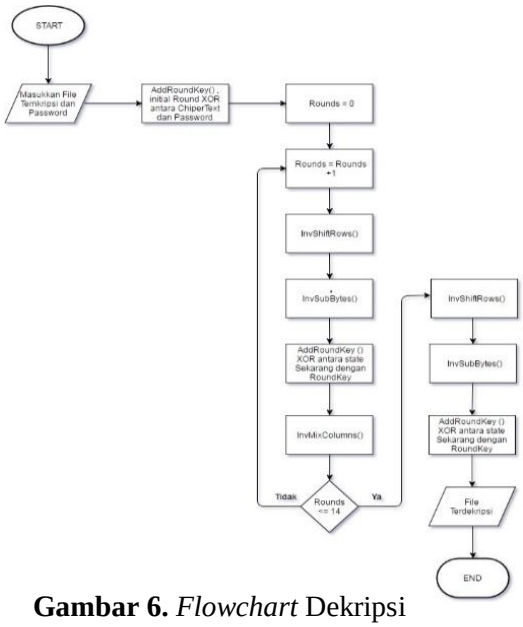
2) Dekripsi
   ini menjelaskan bagaimana proses *dekripsi*, mulai dari proses memilih data yang akan *dekripsi*, memberlikan *password*. Flowchart dapat dilihat pada gambar 6.

3.3 Data Masukan

Data masukan yang dimaksud tersebut ialah file atau dokumen yang dapat dienkripsi dengan aplikasi ini. Beberapa dokumen yang dapat dienkripsi adalah dokumen-dokumen yang berekstensi .doc, .docx, .xls, .xlsx, .pdf, .ppt, .pptx. Untuk ukuran data diatas 15 Mb proses pengenkripsian berjalan dengan lambat karena aplikasi ini tidak menggunakan algoritma tambahan seperti algoritma kompresi.



Gamber 5. *Flowchart* Enkripsi



Gambar 6. *Flowchart* Dekripsi

3.4. Langkah Pengujian

Pada form enkripsi ini terdapat 2 (dua) field yang dapat diisi yaitu field untuk memilih file dan untuk memasukkan *password*. Jika pengguna ingin melakukan *enkripsi* maka pengguna harus menekan tombol *choose file* yang ada pada sisi kanan form. Setelah itu pengguna harus memilih *file* apa yang akan di*enkripsi*. Setelah *file* dipilih, pengguna harus memasukkan *password* untuk menjaga keamanan dari *file* yang dienkripsi. *Password* ini selain digunakan untuk menjaga keamanan *file*, juga digunakan untuk melakukan *dekripsi* pada *file* yang sama.

Setelah memasukkan *password*, pengguna harus menekan tombol enkripsi. Jika enkripsi berhasil maka akan muncul dialog "proses berhasil" setelah itu muncul *popup browse* untuk memilih direktori penyimpanan *password*. *Password* tersebut disimpan dalam bentuk *text*, jadi *password* tidak sembarangan disimpan didalam suatu *folder*. Jika *password* yang dimasukkan hilang atau lupa, maka pengguna dapat melihat *password* yang telah disimpan dan *password* tersebut dapat disebarkan kepada siswa pada saat ujian dimulai untuk melakukan proses *dekripsi* terhadap file tersebut. Jika file gagal dienkripsi maka akan muncul dialog "proses gagal". Tampilan form menu Enkripsi dapat dilihat pada gambar 7.

Gambar 7: Tampilan Form *Enkripsi*

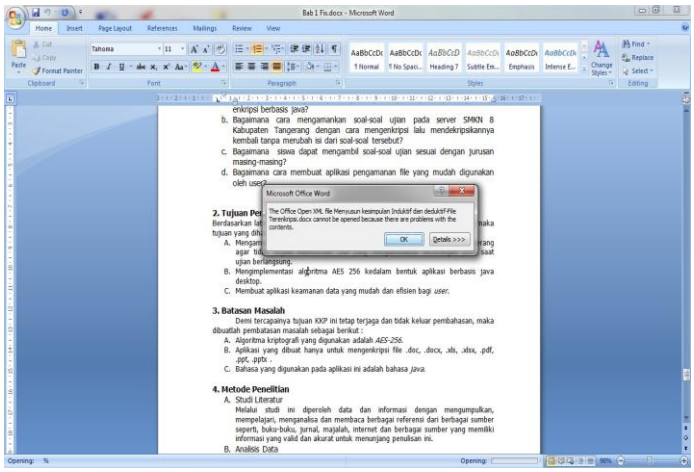
Pada form dekripsi berikut ini terdapat 2 (dua) *field* yang dapat diisi yaitu *field* untuk memilih *file* dan untuk memasukkan *password*. Jika pengguna ingin melakukan proses dekripsi, maka pengguna harus menekan tombol *choose file* yang ada pada sisi kanan *form*. Setelah itu pengguna harus memilih *file* apa yang akan didekkripsi. Setelah file dipilih pengguna harus menginputkan *password* yang sama pada saat file tersebut dienkripsi. Jika dekripsi berhasil maka akan muncul dialog "proses berhasil". Jika file gagal dienkripsi maka akan muncul dialog "proses gagal". Tampilan form dekripsi dapat dilihat pada gambar 8.

Gambar 8: Tampilan Form *Dekripsi*

Perbandingan dokumen ini adalah perbandingan dari beberapa dokumen atau file yang sudah terenkripsi dan yang belum terenkripsi. Pertama, file asli (contoh menggunakan file .doc, atau .docx) yang akan dienkripsi terlihat pada gambar 9.

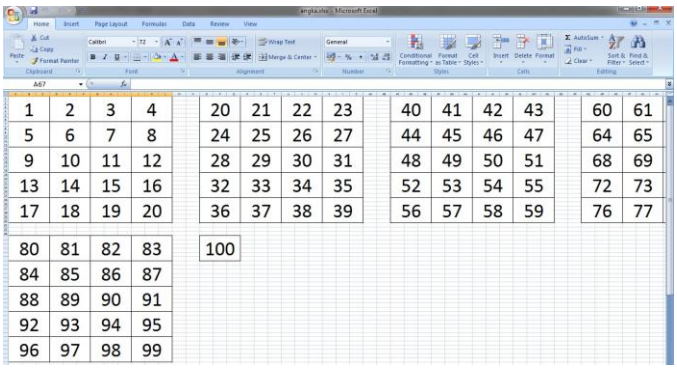
Gambar 9: *File* berekstensi .doc atau .docx Asli

File tersebut dapat dilihat karena belum terenkripsi, apabila file tersebut sudah terenkripsi maka tidak dapat dibuka lagi. File yang sudah terenkripsi dapat dilihat pada gambar 10.



Gambar 10: File berekstensi .doc atau .docx hasil Ter*enkripsi*.

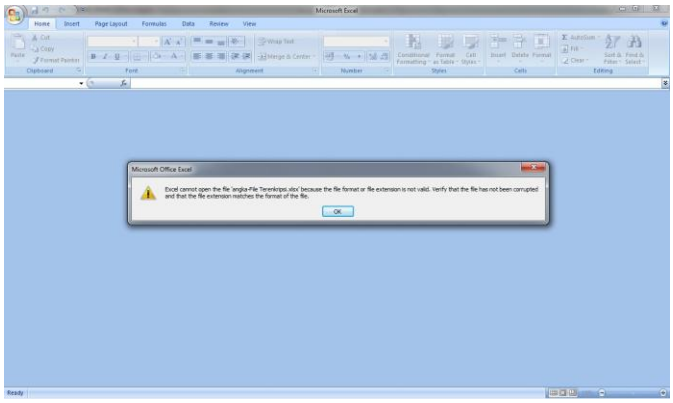
Kedua, *file* asli (contoh menggunakan *file* dengan format .xls atau .xlsx) yang akan dienkripsi. File tersebut dapat dilihat pada gambar 11.



1	2	3	4	20	21	22	23	40	41	42	43	60	61
5	6	7	8	24	25	26	27	44	45	46	47	64	65
9	10	11	12	28	29	30	31	48	49	50	51	68	69
13	14	15	16	32	33	34	35	52	53	54	55	72	73
17	18	19	20	36	37	38	39	56	57	58	59	76	77
80	81	82	83	100									
84	85	86	87										
88	89	90	91										
92	93	94	95										
96	97	98	99										

Gambar 11: File berekstensi .xls atau .xlsx Asli

Apabila *file* tersebut sudah terenkripsi akan tetap bisa dibuka namun isinya sudah teracak sehingga tidak bisa dibaca. Hasil enkripsi file tersebut dapat dilihat pada gambar 12.

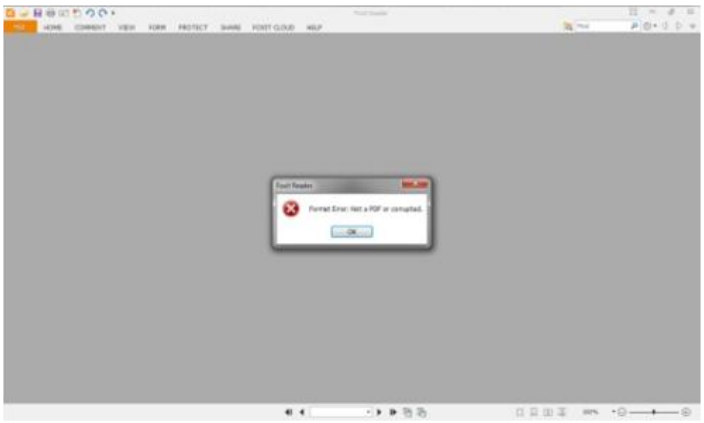


Gambar 12: File berekstensi .xls atau .xlsx hasil Terenkripsi

Ketiga,*file* asli (contoh menggunakan file dengan format .pdf) yang akan *dienkripsi*. File tersebut dapat dilihat pada gambar 13.

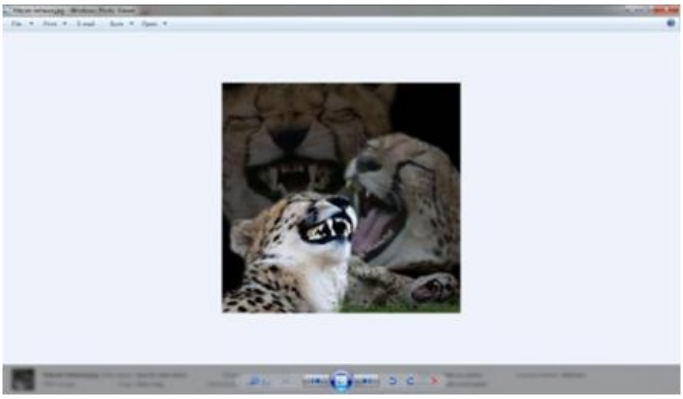
Gambar 13: File berekstensi .pdf Asli

Apabila file tersebut sudah terenkripsi akan tetap bisa dibuka namun isinya sudah teracak sehingga tidak bisa dibaca. Hasil enkripsi file tersebut dapat dilihat pada gambar 14.



Gambar 14: File berekstensi .pdf hasil Terenkripsi

Keempat, file asli (contoh menggunakan file dengan format .jpg atau .jpeg) yang akan dienkripsi. File tersebut dapat dilihat pada gambar 15.



Gambar 15: *File* berekstensi .jpg atau .jpeg Asli

Apabila *file* tersebut sudah terenkripsi akan tetap bisa dibuka namun isinya sudah teracak sehingga tidak bisa dibaca. Hasil enkripsi file tersebut dapat dilihat pada gambar 16.

Gambar 16: File berekstensi .jpg atau .jpeg hasil Terenkripsi

Tabel 1: Tabel perbandingan file dokumen asli dan file dokumen hasil enkripsi

No	Nama File	Kunci	Nama File *Enkripsi*	Ukuran File Asli (kB)	Ukuran File Enkripsi (Kb)
1	Menyusun kesimpulan Induktif dan deduktif.docx	3456rewq0986-4567-1234lkjh	Menyusun kesimpulan Induktif dan deduktif-File Ter*enkripsi*.docx	14	14
2	Angka.xlsx	1234-qwer-5678-zxcv	Angka-File Terenkripsi.xlsx	26	26
3	cisco pert 1.pdf	mnbv-kjhg-iuyt-rewq-4321-zxcv	cisco pert 1-File Terenkripsi.pdf	4.005	4.005
4	Macan.jpg	plmn-rewq-7654-3456	Macan-File Terenkripsi.jpg	29	29

4. KESIMPULAN DAN SARAN

Adapun kesimpulan dan saran yang didapatkan dari hasil penelitian ini yaitu:

4.1 Kesimpulan

Berdasarkan permasalahan dan aplikasi yang telah dibuat, maka dapat ditarik kesimpulan antara lain adalah aplikasi ini sangat mudah untuk dimengerti dan digunakan oleh pengguna serta dapat membantu mengamankan data atau informasi yang sebelumnya dapat dilihat oleh umum atau orang lain. Aplikasi enkripsi ini memiliki tingkat kesulitan enkripsi yaitu 32bit.

4.2 Saran

Enkripsi dengan algoritma AES ini masih memiliki beberapa keterbatasan, untuk itu penulis menyarankan untuk mengembangkan aplikasi ini secara berkelanjutan diantaranya membuat aplikasi ini agar dapat mengenkripsi dan mendekripsi data dengan format .avi, .mkv,.mp3 dan lain sebagainya secara maksimal, serta dapat mengenkripsi dengan ukuran data yang lebih besar dengan kunci enkripsi yang lebih banyak demi keamanan data tersebut sehingga para pengguna dapat menggunakan aplikasi ini secara optimal.

Daftar Pustaka

Jurnal

[1] Yuniati, Voni, dkk. (2009). *Enkripsi* dan *dekripsi* dengan algoritma aes 256 untuk semua jenis file, Jurnal Informatika, Volume 5 Nomor 1.
[2] Mariana, dkk. (2010). Algoritma XTS-AES Untuk *Enkripsi* dan *Dekripsi* Text SMS Berbasis JAVA ME, Jurnal Teknik Informatika STMIK GI MDP.
[3] Angga, Christian. (2011). Perbandingan Super-Enkripsi Berulang vs Vigenere Chiper Kunci Berlapis Metode Triple DES. Sekolah Teknik Elektro Dan Informatika, Bandung
[4] Santoso, Imam Kartika & Robert Habibi. (2014). Kriptografi Pada Aplikasi Komunikasi Data dengan Algoritma AES 256, Jurnal Ilmu Komputer (SNIK 2014), Semarang.
[5] Noni Endriani, (2014). Implementasi Algoritma AES pada Aplikasi SMS Berbasis Android, Yogyakarta.
[6] Muhammad Humam, (2014). Peningkatan Keamanan Algoritma DES Pada Aplikasi Enkripsi SMS Android Menggunakan Algoritma AES 256 Bit, Universitas Dian Nuswantoro.
[7] Nagesh Kumar, Jawahar Thakur, Arvind Kalia, (2011). Performance Analysis of Symmetric Key Cryptography Algorithms: DES, AES and Blowfish, Journal Anu Books.