Perancangan Aplikasi Kriptografi Simetris Menggunakan Algoritma Hill Cipher dan Advanced Encryption Standard

Vijar Miansyah, Robertus laipaka
[1,2]STMIK Pontianak; Jl. Merdeka Barat No. 372, (0561) 735555
[3]Jurusan TeknikInformatika, STMIK Pontianak
e-mail: vijarmiansyah182.stmik@gmail.com, robertus.laipaka@stmikpontianak.ac.id

Abstrak

Proses pengiriman data yang dilakukan seperti media internet, maupun media lainnya. Pada dasarnya pengiriman data tersebut tanpa ada melakukan pengamanan terhadap konten dari data yang dikirim, sehingga ketika dilakukan penyadapan pada alur pengirimannya maka data yang disadap dapat langsung dibaca oleh penyadap. Untuk itu dibutuhkan perangkat lunak sebagai penunjang menggunakan metode penyandian (Kriptografi) tertentu sehingga pesan yang terkandung dalan data yang terkirim tersebut menjadi lebih aman. Penelitian ini menggunakan bentuk penelitian studi literatur dan eksperimen. Adapun teknik pengumpulan data yang digunakan dalam penelitian ini yaitu studi literatur untuk memperoleh teori kriptografi Hill Cipher dan Advanced Encryption Standard (AES). Sedangkan metode perancangan perangkat lunak menggunakan Waterfall karena proses perkembangan perngkat lunak ini berurutan, dimana kemajuan dipandang terus mengalir kebawah melewati fase-fase analisis, design, implementasi, pengujian dan pemeliharaan. Perancangan perangkat lunak menggunakan bahasa pemograman Java, hasil perancangan ini menghasilkan sebuah aplikasi kriptografi simetris menggunakan algoritma Hill Cipher dan Advanced Encryption Standard (AES). Dengan adanya perangkat ini, kerahasian dan keaslian informasi akan lebih terjaga.

Kata Kunci: *perangkat lunak kriptografi, kriptografi, Hill Cipher, Advanced Encryption Standard (AES), Java, Waterfall.*

Abstract

The process of sending data is done such as internet media, or other media. Basically sending the data without any security to the content of the data sent, so that when tapped on the delivery flow then the data tapped can be directly read by the tapper. For that required software as a supporter using an encryption method (Cryptography) so that the message contained in the data sent to be more secure. This research using a form of literature study and experiment. The data collection techniques used in this research is literature study to obtain Hill Cipher cryptography theory and Advanced Encryption Standard (AES). While the method of software design using Waterfall because the process of software development in sequence, Where progress is seen continually flowing down through the phases of analysis, design, implementation, testing and maintenance.This design result generates a symmetric cryptographic application using Hill Cipher algorithms and Advanced Encryption Standard (AES).With this device, confidentiality and authenticity of information will be more awake.

Keywords*: cryptography software, cryptography, Hill Cipher, Advanced Encryption Standard (AES), Java, Waterfall.*

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat memberikan kemudahan kepada penggunanya dalam melakukan pertukaran informasi. Contoh adalah jaringan internet yang memungkinkan pengguna untuk saling tukar pesan atau informasi. Penyadapan terhadap pesan atau informasi merupakan hal yang sangat merugikan bagi pengguna teknologi informasi saat ini. Dengan adanya kemungkinan penyadapan informasi tersebut, maka aspek keamanan dalam pertukaran informasi menjadi penting.

Kriptografi merupakan ilmu penyandian data yang mempunyai hubungan dengan aspek keamanan informasi seperti kerahasian, integritas data dan autentikasi[1]. Dengan menggunakan kriptografi, data sederhana yang dikirim (*plainteks*) diubah ke dalam bentuk data sandi (*cipherteks*), kemudian data sandi tersebut hanya dapat dikembalikan ke bentuk data sebenarnya hanya dengan menggunakan kunci (*key*) tertentu yang dimiliki oleh pihak yang sah saja[2]. Saat ini telah banyak berbagai macam algoritma kriptografi dimana setiap algoritma menawarkan kelebihan dan memiliki kekurangan masing-masing, misalnya kriptografi simetris *Hill Cipher* dan *Advanced Encription Standard* (AES).

Kedua algoritma simetris yang akan dipakai oleh penulis, memiliki kelebihan yaitu Algoritma ini dirancang sehingga proses enkripsi/dekripsi membutuhkan waktu yang singkat, ukuran kunci relatif lebih pendek dan autentikasi pengiriman pesan langsung diketahui dari ciphertext yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima pesan saja atau hanya satu kunci yang di pakai. Algoritma ini juga memiliki kelemahan yaitu Kunci harus dikirim melalui saluran yang aman. Pengirim dan penerima harus menjaga kerahasiaan kunci ini dan kunci harus sering diubah, mungkin pada setiap sesi komunikasi.

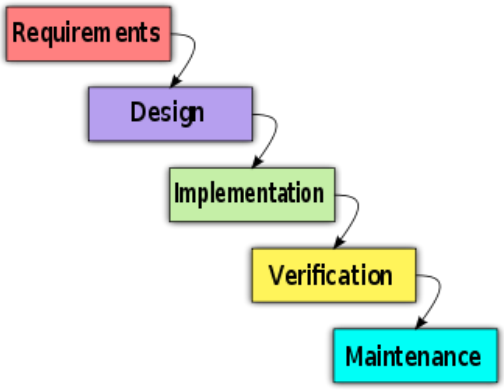
Pada penelitian ini, enkripsi dan dekripsi di lakukan untuk mengamankan suatu file teks yaitu (**.txt*) dan file dokumen (**.docx*). Digunakannya algoritma AES dalam mengenkripsi file dikarenakan AES ini merupakan algoritma block cipher dengan menggunakan sistem permutasi dan subsitusi (P-Box dan S-Box) yang dilakukan pada pesan yang akan dienkripsi untuk menghasilkan ciphertext. AES ini mempunyai kelebihan yaitu setiap putaran akan menghasilkan kunci (subKey) yang berbeda sehingga memiliki tingkat keamanan yang baik dan juga memiliki kecepatan dalam proses enkripsi dan dekripsi tetapi algoritma AES ini memiliki kelemahan juga yaitu kelemahan terletak pada kunci yang digunakan untuk enkripsi. Kunci yang digunakan harus berbeda-beda agar kriptanalist kesulitan untuk mendapatkan kunci dari setiap *chipertext*.

Algoritma *Hill Cipher* dalam mengenkripsi file dikarenakan algoritma ini sangat sulit dipecahkan oleh *kriptanalis* apabila dilakukan hanya dengan mengetahui berkas ciphertext saja, karena *Hill Cipher* tidak mengganti setiap abjad yang sama pada *plaintext* dengan abjad lainnya yang sama pada *ciphertext* karena menggunakan matriks sebagai kunci dan perkalian matriks pada dasar *enkripsi* dan *dekripsi*. Akan tetapi bukan berarti teknik ini tidak ada kelemahannya, *Hill Cipher* dapat dipecahkan dengan cukup mudah apabila *kriptanalis* memiliki berkas *cipherteks* dan potongan berkas *plainteks*.

2. METODE PENELITIAN

Penelitian ini berbentuk studi literature dan perancangan eksperimen, sebagai bahan untuk mengumpulkan dan melakukan perancangan, implementasi system untuk membuat gambaran yang jelas dimasalah yang dihadapi. Teknik pengumpulan data dilakukan dengan

melakukan studi literatur dan observasi. Hasil dari observasi dikumpulkan menjadi data latih yang akan digunakan untuk membangun algoritma Hill Cipher dan AES untuk melakukan penyandian. Metode perancangan perangkat lunak menggunakan *Waterfall* karena proses perkembangan perangkat lunak ini berjalan satu arah dari awal sampai proyek selesai [3].



Gambar 1. Fase – Fase Waterfall

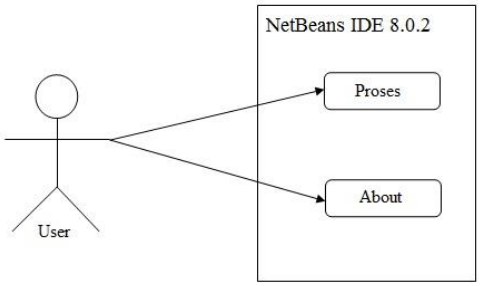
Metode perancangan perangkat lunak yang digunakan dalam penelitian ini yaitu menggunakan Waterfall.
Ada pun *Fase-fase* waterfall model sebagai berikut :

1. *Requeirment Analusis and Definition*
   Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh *software* yang akan dibangun.
2. *Software and Software Desain*
   Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software.* Untuk mengetahui sifat dari program yang akan dibuat, maka para *software enginer* harus mengerti tentang domain informasi dari *software*.
3. *Implementation and Unit Testing*
   Desain program diterjemahkaan ke dalam kode-kode dengan menggunakan bahas pemrograman yang ditentukan. Program yang dibangun langsung diuji baik secara unit.
4. *Integration and System Testing*
   Tahap ini merupakan implementasi dari tahap desain yang secara teknis nantinya dikerjakan oleh *programmer*. Penyatuan unit-unit program kemudian diuji keseluruhan (*system testing*).
5. *Operation and Maintenance*
   Semua fungsi-fungsi *software* harus diuji cobakan, agar *software* bebas dari *eror* dan hasilnya harus benar-benar sesuai dengan kebutuhan yang didefinisikan sebelumnya. Pemeliharaan suatu *software* diperlukan, termasuk didalamnya adalah pengembangan.

Metode pengujian yang digunakan adalah metode pengujian *blackbox* dan *whitebox*. Pengujian blackbox berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan engineers untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program[4].Untuk pengujian whitebox yaitu menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi, masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian White-Box dilakukan dengan memeriksa logika dari kode program dan pembuatan kasus uji bisa mengikuti standar pengujian dari standar pemrograman yang seharusnya[5].

3. HASIL DAN PEMBAHASAN

Sistem yang dikembangkan secara garis besar adalah perangkat lunak yang memiliki fungsi untuk melakukan enkripsi dan dekripsi tipe file Txt dan Docx dengan menggunakan algoritma Hill Cipher dan AES, Berikut ini arsitektur dari aplikasi kriptografi simetris menggunakan algoritma Hill Cipher dan AES.

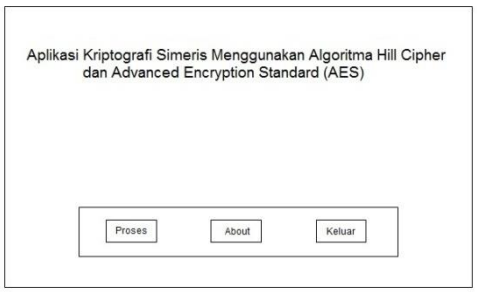


Gambar 2. Arsitektur Aplikasi Kriptografi Simetris Menggunakan Algoritma Hill Cipher Dan Advanced Encryption Standard (AES)

Perangkat lunak kriptografi ini dirancang dengan mneggunakan bahasa pemograman java dengan *software development* yaitu *Netbeans IDE*. Hak akses pada perangkat lunak ini berupa *user*. User dapat mengakses *form* Proses dan About.

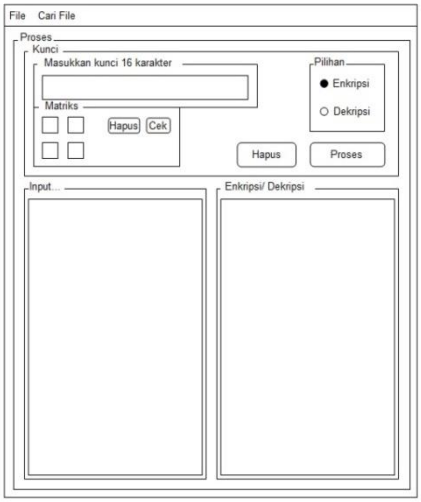
3.1 Tampilan antarmuka

Menu utama dari perangkat lunak kriptografi ini dirancang menggunakan form standard dari Netbeans IDE 8.0.2. pada tampilan menu utama ini terdapat beberapa button yaitu button proses, button about dan button keluar. Tampilan perancangan dapat dilihat pada gambar dibawah ini.



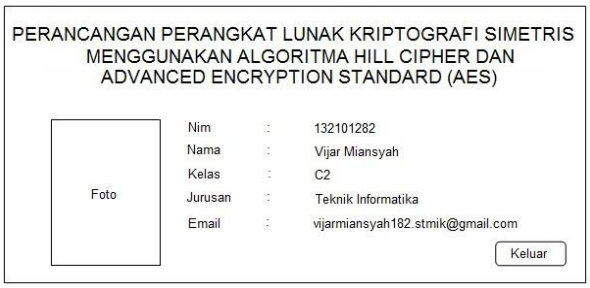
Gambar 3. Tampilan MenuUtama

Berikut merupakan perancangan interface dari form proses terdiri dari beberapa lebel yaitu : masukkan kunci, matriks, input, enkripsi/dekripsi dan pilihan. Terdapat juga radioButton yaitu : enkripsi dan dekripsi. Terdapat Menu item yaitu menu file dan cari file dan beberapa Button yaitu : cari, hapus, proses, simpan dan keluar dan ada beberapa textBox dan textArea.



Gambar 4. Tampilan Form Proses

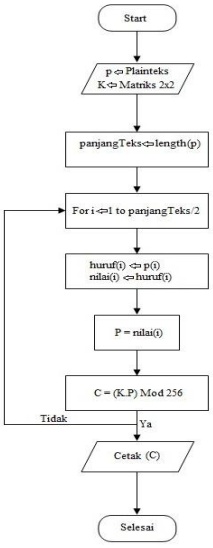
Pada gambar dibawah merupakan perancangan interface dari form About terdiri dari lebel : judul, nim, nama, kelas, jurusan dan email yang ada disebelah kanan. Terdapat juga pictureBox sebelah kiri dan button keluar dibawah.



Gambar 5. Tampilan Form About

3.2 Proses Enkripsi

Pada proses ini merupakan gambaran pengamanan data yang dikirim agar terjaga kerahasiaanya. Berikut ini contoh proses enkripsi menggunakan algoritma Hill Cipher dan AES.



Gambar 6. Flowchart Proses Enkripsi Hill Cipher

Pertama plaintext dienkripsi dengan menggunakan algoritma Hill Cipher :

Plaintext : stmik pontianak1

Kunci matriks Hill Cipher : $\begin{bmatrix}\mathbf{4} & \mathbf{1}\\ \mathbf{7} & \mathbf{2}\end{bmatrix}$

Bagi plaintext menjadi matrik 2x1 dan konversi plaintext ke dalam bentuk nilai bilangan karakter.

$\begin{bmatrix}s\\ t\end{bmatrix} = \begin{bmatrix}115\\ 116\end{bmatrix} \begin{bmatrix}m\\ i\end{bmatrix} = \begin{bmatrix}109\\ 105\end{bmatrix} \begin{bmatrix}k\\ \end{bmatrix} = \begin{bmatrix}107\\ 20\end{bmatrix} \begin{bmatrix}p\\ o\end{bmatrix} = \begin{bmatrix}112\\ 111\end{bmatrix} \begin{bmatrix}n\\ t\end{bmatrix} = \begin{bmatrix}110\\ 116\end{bmatrix} \begin{bmatrix}i\\ a\end{bmatrix} = \begin{bmatrix}105\\ 97\end{bmatrix} \begin{bmatrix}n\\ a\end{bmatrix} = \begin{bmatrix}110\\ 97\end{bmatrix}$
$\begin{bmatrix}k\\ 1\end{bmatrix} = \begin{bmatrix}107\\ 49\end{bmatrix}$

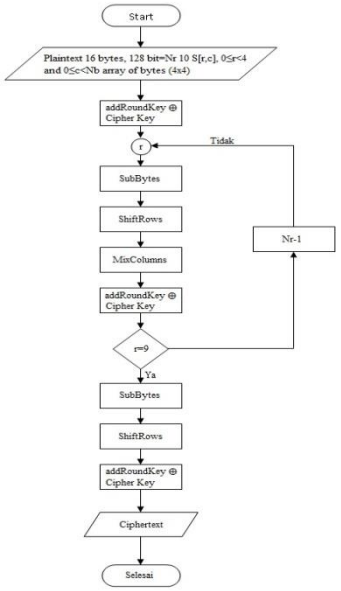
Kemudian seluruh nilai bilangan karakter di kalikan dengan matriks kunci hill cipher dan lakukan operasi mod 256 setiap angka matriks.

$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}115\\ 116\end{bmatrix} = \begin{bmatrix}576\\ 1037\end{bmatrix}$ mod 256 = $\begin{bmatrix}64\\ 13\end{bmatrix}$
$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}109\\ 105\end{bmatrix} = \begin{bmatrix}541\\ 973\end{bmatrix}$ mod 256 = $\begin{bmatrix}29\\ 205\end{bmatrix}$
$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}107\\ 20\end{bmatrix} = \begin{bmatrix}448\\ 789\end{bmatrix}$ mod 256 = $\begin{bmatrix}192\\ 21\end{bmatrix}$
$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}112\\ 111\end{bmatrix} = \begin{bmatrix}559\\ 1006\end{bmatrix}$ mod 256 = $\begin{bmatrix}47\\ 238\end{bmatrix}$
$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}110\\ 116\end{bmatrix} = \begin{bmatrix}556\\ 1002\end{bmatrix}$ mod 256 = $\begin{bmatrix}44\\ 234\end{bmatrix}$
$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}105\\ 97\end{bmatrix} = \begin{bmatrix}517\\ 929\end{bmatrix}$ mod 256 = $\begin{bmatrix}5\\ 161\end{bmatrix}$
$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}110\\ 97\end{bmatrix} = \begin{bmatrix}537\\ 964\end{bmatrix}$ mod 256 = $\begin{bmatrix}61\\ 196\end{bmatrix}$
$\begin{bmatrix}4 & 1\\ 7 & 2\end{bmatrix} \begin{bmatrix}107\\ 49\end{bmatrix} = \begin{bmatrix}447\\ 847\end{bmatrix}$ mod 256 = $\begin{bmatrix}191\\ 79\end{bmatrix}$

Pada perhitungan diatas didapat ciphertext sementara yaitu

Ciphertext sementara : 64 13 29 205 192 21 47 238 44 234 5 161 61 196 191 79

Selanjutnya ciphertext sementara akan menjadi input untuk proses enkripsi menggunakan AES dan akan mengahasilkan output ciphertext akhir.



Gambar 7. Flowchart proses Enkripsi AES

Ciphertext sementara dikonversi dari bilangan desimal ke bilangan hexadesimal.
Ciphertext sementara : 64 13 29 205 192 21 47 238 44 234 5 161 61 196 191 79
Dalam hexadesimal : 40 0d 1d cd c0 15 2f ee 2c ea 05 a1 3d c4 bf 4f
Kunci AES : ankatantahun2013
Kunci dalam hexadesimal : 61 6e 6b 61 74 61 6e 74 61 68 75 6e 32 30 31 33

Tabel 1. Input State dan Key

State

40	C0	2c	3d
0d	15	Ea	C4
1d	2f	05	Bf
cd	Ee	A1	4f

proses enkripsi

⊕

Key

61	74	61	32
6e	61	68	30
6b	6e	75	31
61	74	6e	33

key Schedule

Hexadecimal
Contoh : 40 4 hex (0100) | 0 hex (0000) → 1 byte

Tabel 2. hasil *AddRoundKey*

21	b4	4d	0f
63	74	82	f4
46	41	70	8e
ac	9a	c3	7c

Tabel state di atas merupakan hasil dari XOR plaintext dengan key diatas tadi atau disebut *addRoundKey*. Kemudian saat putaran awal lakukan *subByte*, sebagai contoh elemen 21 dipetakan dengan tabel *S-Box*, untuk x=2 dan y=1, sehingga didapatkan titik temu pada kotak fd, dapat dilihat pada tabel dibawah ini :

Tabel 3. hasil proses *SubBytes*

21	b4	4d	0f
63	74	82	f4
46	41	70	8e
ac	9a	c3	7c

⇨

fd	8d	e3	76
fb	92	13	bf
5a	83	51	19
91	b8	8b	10

Tabel 4. S-Box

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Setelah setiap elemen dipetakan, sehingga didapatkan elemen baru kemudian dilanjutkan dengan proses shiftRows yaitu proses pergeseran bit dimana bit paling kiri akan di pindahkan menjadi bit paling kanan. Jumlah pergeseran bergantungpada nilai baris(*r*). Baris *r* = 1 digeser sejauh *1 byte*, baris *r = 2* digeser sejauh 2 *byte*dan baris *r = 3* digeser sejauh 3 *byte*, sedangkan baris *r = 0*,tidak digeser.

Tabel 5. Sebelum Pergeseran

Fd	8d	e3	76
Fb	92	13	bf
5a	83	51	19

91	b8	8b	10

Setelah dilakukan pergeseran atau *shiftRows* didapat elemen baru, sebagai berikut :

Tabel 6. Sesudah Pergeseran

Fd	8d	e3	76
92	13	Bf	fb
51	19	54	83
10	91	b8	8b

Kemudian lakukan proses *MixColums* yaitu mengoperasikan setiap elemen yang berada dalam satu kolom pada state, *MixColumns* mengalikan setiap kolom dari *array state.*

Gambar 8. proses MixColums

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Fd	8d	e3	76
92	13	Bf	fb
51	19	54	83
10	91	b8	8b

Tabel 7. Proses MixColumns

fd	=	0d
92		01
51		Ce
10		dd

Tabel 8. Hasil MixColumn

0d	bc	Eb	e9
01	11	c2	0d
ce	04	27	16
dd	17	be	ef

Selanjutnya proses *addRoundKey* yaitu melakukan operasi XOR terhadap sebuah *Round Key* dengan *array State*, hasilnya disimpan di *array state*. Untuk proses enkripsi dan dekripsi proses *addRoundKey sama*. Setiap *Round Key* terdiri dari *Nb word* tiap *word* akan dijumlahkan dengan word atau kolom yang bersesuai dengan state

Tabel 9. state *addRoundKey* dan *Round Key*

0d	bc	eb	e9
01	11	c2	0d
ce	04	27	16
dd	17	be	Ef

$\oplus$

64	10	71	42
a9	28	40	70
a8	26	53	62
42	36	59	6a

Tabel 10. Proses XOR Kolom

0d	$\oplus$	64		69
01		a9		a8
ce		a8		66
dd		42		9f

Tabel 11. hasil addRoundKey

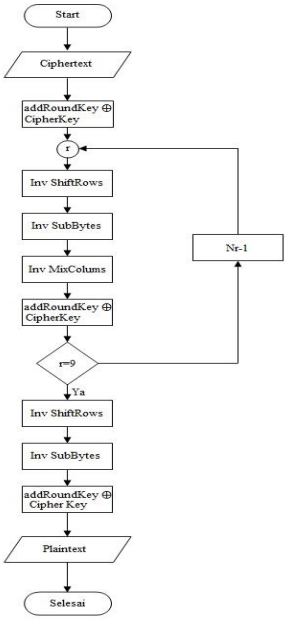
69	ac	94	Ab
a8	39	82	7d
66	22	74	74
9f	21	e7	c5

Dapat diihat hasil dari MixColumns pada state diatas. Pada *array state* diatas baru putaran pertama, lakukan lagi proses yang sama secara berulang hinggan 9 putaran dan 1 untuk finalRound. Jadi ciphertext akhir dari proses enkripsi yaitu :

Ciphertext : 59 01 ab f4 c9 f53d f2 cd e8 cc da 67 10 81 fa

3.4 Proses Dekripsi

Pada proses ini merupakan proses kebalikan dari enkripsi. Pesan yang telah di enkripsi dikembalikan kebentuk asalnya (teks asli) disebut dengan dekripsi pesan. Berikut ini contoh proses dekripsi menggunakan algoritma Hill Cipher dan AES.



Gambar 9. Flowcart Proses Dekripsi Menggunakan Algoritma AES

Ciphertext : 59 01 ab f4 c9 f53d f2 cd e8 cc da 67 10 81 fa

Kunci AES : 61 6e 6b 61 74 61 6e 74 61 68 75 6e 32 30 31 33

Tabel 12. Input State dan Key

State

59	c9	cd	67
01	f5	e8	10
ab	3d	cc	81
f4	f2	da	fa

proses enkripsi

⊕

Key

61	74	61	32
6e	61	68	30
6b	6e	75	31
61	74	6e	33

key Schedule

Tabel 13. hasil *AddRoundKey*

38	bb	ac	55
6f	74	80	20
c0	53	b9	b0

95	86	94	c9

Tabel state di atas merupakan hasil dari XOR plaintext dengan key diatas tadi atau disebut *addRoundKey*. kemudian dilanjutkan dengan proses invshiftRows yaitu proses kebalikan dari proses shiftRows dimana bit paling kanan akan di pindahkan menjadi bit paling kiri. Jumlah pergeseran bergantungpada nilai baris(*r*). Baris *r* = 1 digeser sejauh *1 byte*, baris *r = 2* digeser sejauh 2 *byte*dan baris *r = 3* digeser sejauh 3 *byte*, sedangkan baris *r = 0*,tidak digeser.

Tabel 14. Sebelum Pergeseran

38	bb	ac	55
6f	74	80	20
c0	53	b9	b0
95	86	94	c9

Setelah dilakukan pergeseran atau *shiftRows* didapat elemen baru, sebagai berikut :

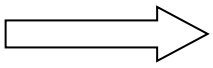
Tabel 15. Sesudah Pergeseran

38	Bb	ac	55
20	6f	74	80
b9	b0	c0	53
86	94	c9	95

Kemudian lakukan proses *InvSubByte* sama seperti proses subByte, sebagai contoh elemen 21 dipetakan dengan tabel *invS-Box*, untuk x=3 dan y=8, sehingga didapatkan titik temu pada kotak 76, dapat dilihat pada tabel dibawah ini :

Tabel 16. hasil proses *InvSubBytes*

38	bb	Ac	55
20	6f	74	80
b9	b0	c0	53
86	94	c9	95



76	fe	Aa	ed
54	06	Ca	3a
db	fc	1f	50
dc	E7	12	ad

Tabel 17. *inv S-Box*

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Kemudian lakukan proses *InvMixColums* yaitu mengoperasikan setiap elemen yang berada dalam satu kolom pada state, *InvMixColumns* mengalikan setiap kolom dari *array state.*

Tabel 18. proses *InvMixColums*

76	fe	aa	ed
54	06	ca	3a
db	fc	1f	50

dc	e7	12	ad

Gambar 10. Proses *InvMixColumns*

$$\begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{matrix} 76 \\ 54 \\ db \\ dc \end{matrix} = \begin{matrix} ef \\ ce \\ 4d \\ 19 \end{matrix}$$

Tabel 19. Hasil *InvMixColumn*

ef	4d	c3	23
ce	d6	22	89
4d	b6	44	02
19	62	86	6a

Selanjutnya proses *addRoundKey* yaitu melakukan operasi XOR terhadap sebuah *Round Key* dengan *array State*, hasilnya disimpan di *array state*. Untuk proses enkripsi dan dekripsi proses *addRoundKey* sama. Setiap *Round Key* terdiri dari *Nb word* tiap *word* akan dijumlahkan dengan word atau kolom yang bersesuai dengan state

Tabel 20. state *addRoundKey* dan *Round Key*

Ef	4d	c3	23
Ce	d6	22	89
4d	b6	44	02
19	62	86	6a

$\oplus$

64	10	71	42
a9	28	40	70
a8	26	53	62
42	36	59	6a

Gambar 21. Proses XOR Kolom

ef		64		8b
ce	$\oplus$	A9	=	67
4d		A8		e5
19		42		5b

Tabel 22. hasil addRoundKey

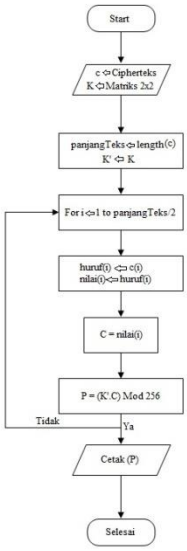
8b	5d	b2	61
67	ae	62	f9
e5	90	17	60
5b	54	df	d2

Dapat diihat hasil dari InvMixColumns pada state diatas. Pada *array state* diatas baru putaran pertama, lakukan lagi proses yang sama secara berulang hinggan 9 putaran dan 1 untuk finalRound.

Jadi plaintext sementara dari proses dekripsi AES yaitu :

Ciphertext : 40 0d 1d cd c0 15 2f ee 2c ea 05 a1 3d c4 bf 4f

Setelah dapatkan plaintext sementara dari proses AES selanjutnya yaitu proses dekripsi menggunakan Hill Cipher.



Gambar 11. Flowchart Proses Dekripsi Menggunakan Algoritma Hill Cipher

Proses dekripsi pada Hill Cipher pada dasarnya sama dengan proses enkripsinya. Namun matriks kunci harus dibalik (invers) terlebih dahulu.

$$C = K \,.\, P$$
$$K{-}1.\, C = K{-}1.\, K \,.\, P$$
$$K{-}1.\, C = I \,.\, P$$
$$P = K{-}1.\, C$$

Sehingga proses dekripsi dapat ditulis dengan persamaan:
$P = K{-}1.\, C$
P = *plaintext.*
K-1 = invers matriks kunci.
C = *ciphertext*.

Dengan menggunakan kunci $K = \begin{bmatrix} 4 & 1 \\ 7 & 2 \end{bmatrix}$ maka proses dekripsi diawali dengan mencari invers matriks K.

$$K = \begin{bmatrix} 4 & 1 \\ 7 & 2 \end{bmatrix} = \frac{1}{2.4-1.7}\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}$$
$$= \frac{1}{8-7}\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}$$
$$= \frac{1}{1}\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}$$
$$= \begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}$$

Kemudian seluruh nilai bilangan karakter di kalikan dengan invers matriks kunci hill cipher dan lakukan operasi mod 256 setiap angka matriks.

$$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}\begin{bmatrix} 64 \\ 13 \end{bmatrix} = \begin{bmatrix} 115 \\ 116 \end{bmatrix} \text{ mod } 256 = \begin{bmatrix} 115 \\ 116 \end{bmatrix}$$
$$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}\begin{bmatrix} 29 \\ 205 \end{bmatrix} = \begin{bmatrix} 109 \\ 105 \end{bmatrix} \text{ mod } 256 = \begin{bmatrix} 109 \\ 105 \end{bmatrix}$$
$$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}\begin{bmatrix} 192 \\ 21 \end{bmatrix} = \begin{bmatrix} 107 \\ 20 \end{bmatrix} \text{ mod } 256 = \begin{bmatrix} 107 \\ 20 \end{bmatrix}$$
$$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix}\begin{bmatrix} 47 \\ 238 \end{bmatrix} = \begin{bmatrix} 112 \\ 111 \end{bmatrix} \text{ mod } 256 = \begin{bmatrix} 112 \\ 111 \end{bmatrix}$$

$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix} \begin{bmatrix} 44 \\ 234 \end{bmatrix} = \begin{bmatrix} 110 \\ 116 \end{bmatrix} \bmod 256 = \begin{bmatrix} 110 \\ 116 \end{bmatrix}$
$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 161 \end{bmatrix} = \begin{bmatrix} 105 \\ 97 \end{bmatrix} \bmod 256 = \begin{bmatrix} 105 \\ 97 \end{bmatrix}$
$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix} \begin{bmatrix} 61 \\ 196 \end{bmatrix} = \begin{bmatrix} 110 \\ 97 \end{bmatrix} \bmod 256 = \begin{bmatrix} 110 \\ 97 \end{bmatrix}$
$\begin{bmatrix} 2 & -1 \\ -7 & 4 \end{bmatrix} \begin{bmatrix} 191 \\ 79 \end{bmatrix} = \begin{bmatrix} 107 \\ 49 \end{bmatrix} \bmod 256 = \begin{bmatrix} 107 \\ 49 \end{bmatrix}$

Pada perhitungan diatas didapat plaintext akhir yaitu
Plaintext Akhir : 115 116 109 105 107 20 112 111 110 116 105 97 110 107 49
Konversi plaintext menjadi karakter untuk mendapatkan pesan
Plaintext Akhir : 115 116 109 105 107 20 112 111 110 116 105 97 110 107 49
Plaintext Akhir : stmik pontianak1

3.4 pengujian Data

Tabel 23. Input Teks

Input Teks	Enkripsi	Dekripsi	Proses
A	Tidak	Tidak	Tidak dapat melakukan proses
Aa	Tidak	Tidak	Tidak dapat melakukan proses
Aaa	Tidak	Tidak	Tidak dapat melakukan proses
Aaaa	Tidak	Tidak	Tidak dapat melakukan proses
Aaaaa	Tidak	Tidak	Tidak dapat melakukan proses
Aaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaaaaaaaa	Tidak	Tidak	Tidak dapat melakukan proses
aaaaaaaaaaaaaaaa	Ya	Ya	Dapat melakukan proses

Tabel 24. Input File

Nama File	Tipe File	Ukuran File	Waktu Proses(milidetik)		File Hasil Dekripsi = File Asli
			Enkripsi	Dekripsi	
1	Txt	1 kb	0,14	0,14	Ya
1	Docx	9,85 kb	0,17	0,17	Ya
2	Txt	2 kb	0,40	0,40	Ya
2	Docx	10 kb	0,45	0,45	Ya
4	Txt	4 kb	0,55	0,55	Ya
4	Docx	10,1 kb	0,58	0,58	Ya
8	Txt	8 kb	0,70	0,70	Ya
8	Docx	10,2 kb	0,72	0,72	Ya
16	Txt	16 kb	0,90	0,90	Ya
16	Docx	10,4 kb	0,85	0,85	Ya
32	Txt	32 kb	01,50	01,50	Ya
32	Docx	10,9 kb	01,17	01,17	Ya
64	Txt	64 kb	04,74	04,74	Ya
64	Docx	11,9 kb	04,20	04.20	Ya
128	Txt	128 kb	14.25	14.25	Ya
128	Docx	13,5 kb	13.51	13.51	Ya

Dari hasil pengujian diatas dapat disimpulkan bahwa ukuran file sebelum dienkripsi sama dengan file yang sudah didekripsi, semakin besar ukuran file yang dienkripsi/dekripsi maka semakin besar pula waktu yang diperlukan untuk proses enkripsi/dekripsi dan ukuran file hasil dekripsi akan sama dengan file aslinya.

4. KESIMPULAN

Berdasarkan hasil implementasi perangkat lunak aplikasi kriptografi yang dibuat, maka penulis mengambil kesimpulan sebagai berikut :

1. Perangkat lunak kriptografi ini memiliki fungsi utama yaitu melakukan enkripsi pada file yang berformat .docx dan .txt dan melakukan dekripsi pada file hasil enkripsi.
2. Algoritma AES dan Hill Cipher merupakan dua lagoritma yang kuat. Tidak menutup kemungkinan kedua algoritma tersebut memiliki kelebihan dan kekurangannya masing-masing, yaitu :
   a. Algoritma AES ini mempunyai kelebihan yaitu setiap putaran akan menghasilkan kunci (subKey) yang berbeda sehingga memiliki tingkat keamanan yang baik dan juga memiliki kecepatan dalam proses enkripsi dan dekripsi tetapi algoritma AES ini memiliki kelemahan juga yaitu kelemahan terletak pada kunci yang digunakan untuk enkripsi. Kunci yang digunakan harus berbeda-beda agar kriptanalist kesulitan untuk mendapatkan kunci dari setiap *chipertext*.
   b. Algoritma *Hill Cipher* mempunyai kelebihan yaitu terletak pada kunci yang berbentuk matriks dan perkalian matriks pada dasar *enkripsi* dan *dekripsi*. Akan tetapi bukan berarti teknik ini tidak ada kelemahannya, *Hill Cipher* dapat dipecahkan dengan cukup mudah apabila *kriptanalis* memiliki berkas *cipherteks* dan potongan berkas *plainteks*.
3. Kekurangan dari perangkat lunak ini adalah kurangnya tidak bisa melakukan proses enkripsi/dekripsi teks kurang dari 16 karakter, kurangnya peringatan bila pengguna salah dalam pengoperasian program dan tidak adanya fitur pengingat password pada perangkat lunak ini.

5. SARAN

Penulis menyadari bahwa perangkat lunak kriptografi yang dbuat ini belum sepenuhnya sempurna. Penulis berharapkan dan menyarankan dimasa yang akan datang, agar pembaca atau programer yang lebih handal dapat mengembangkan dan menyempurnakan kekurangan-kekurangan dari perangkat lunak ini. Pembaca diharapkan dapat mengembangkan perangkat lunak kriptografi agar lebih baik lagi antara lain dengan cara :

1. Perangkat lunak dapat dikembangkan agar dapat di gunakan dengan metode kriptografi yang lain selain metode Hill Cipher dan Advanced Encryption Standard (AES) sehingga tingkat keamanan yang dihasilkan jauh lebih baik.
2. Dapat melakukan enkripsi dan dekripsi file selain file yang berformat .docx dan txt.
3. Dapat melakukan proses enkripsi/dekripsi teks kurang dari 16 karakter dan dapat peringatan bila pengguna salah dalam pengoperasian program.
4. Dapat mengoptimalkan algoritma yang sudah ada dalam perangkat lunak kriptografi ini sehingga dapat mempercepat proses enkripsi dan dekripsi.

DAFTAR PUSTAKA

[1] Rifki Sadikin. 2012. Kriptografi Untuk Keamanan Jaringan. Yogyakarta:Andi
[2] Kromodioeljo, Sentot. 2009, *Teori dan Aplikasi Kriptografi*,SPK IT Consulting.
[3] Rosa, A.S., Shalahuddin,M. 2013. Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Objek. Informatika. Bandung.
[4] Pressman, Roger S. 2012. Rekayasa Perangkat Lunak, jilid I. Yogyakarta: Andi.
[5] Rosa, A.S., Shalahuddin,M. 2013. Rekayasa Perangkat Lunak : Terstruktur dan Berorientasi Objek. Informatika. Bandung.