Modifikasi Kriptografi Hill Cipher Kunci Matriks Persegi Panjang Menggunakan Fungsi Xor Dan Fungsi Xnor

Tuti Alawiyah[1)], Agung Baitul Hikmah[2)]
[1)] Universitas BSI Bandung, [2)]AMIK BSI Purwokerto
[1)] tuti.tah@bsi.ac.id, [2)]agung.abl@bsi.ac.id

Abstract *- Cryptography is needed to store or transmit information/data. There are two important things in cryptography, which are, encryption (change an information into a form that will not be understood), and description (change the encryption back into the original message). The original message called as plaintext while the encryption form called ciphertext. Hill cipher is one of the cryptography algorithms that keeps developing. One of the development is the use of rectangle matrix as the key matrix. The use of rectangle matrix makes the ciphertext be longer than the plaintext. Those makes the message be more disguised. In this paper, the writer modifies the hill cipher algorithm by using the rectangle matrix in binary operation by adds the X-OR and XNOR logic. By adding the X-OR and XNOR logic, crypto-analyst finds difficulties in the linier equation to find the plaintext and the key matrix. Binary operation makes the ciphertext be longer than the ciphertext in the rectangle key matrix only. Length of ciphertext is* l_c *=((((2 ** l_p*)/m * n) * 2)/m)*n. and ((2 ** l_p*) mod m) =0. If ((2 *** l_p*) mod m) ≠ 0 character of* l_p *must be adds till ((2 *** l_p*) mod m) =0.*

Keywords: Cryptography, Hill Cipher, X-OR, XNOR

Abstrak - Kriptografi sangat diperlukan untuk menyimpan atau mengirimkan informasi/data. Kriptografi terdiri dari 2 hal penting, yaitu enkripsi (merubah informasi menjadi bentuk yang tidak dimengerti) dan deskripsi (mengembalikan informasi dari bentuk yang tidak dimengerti menjadi informasi aslinya). Informasi atau pesan asli disebut *plaintext*, sedangkan hasil penyandiannya disebut *ciphertext*. Hill cipher merupakan salah satu algoritma kriptografi yang terus berkembang. Salah satu perkembangannya adalah penggunaan matriks persegi panjang sebagai matriks kuncinya. Penggunaan matriks persegi panjang menjadikan *ciphertext* lebih panjang dari *plaintext*, sehingga pesan menjadi lebih tersamarkan. Pada tulisan ini, penulis memodifikasi algoritma hill cipher dengan matriks persegi panjang menggunakan operasi biner dengan menambahkan fungsi logika X-OR dan XNOR. Dengan penambahan fungsi logika X-OR dan XNOR, kripnatalis sulit menemukan persamaan linier untuk menemukan *plaintext* dan matriks kuncinya. Operasi biner memungkinkan *ciphertext* yang dihasilkan lebih panjang daripada *ciphertext* yang hanya menggunakan matriks kunci persegi panjang saja. Jumlah karakter *ciphertext* yang dihasilkan adalah: l_c =((((2 * l_p)/*m* * *n*) * 2)/*m*)**n*. dengan l_p harus memenuhi syarat ((2 * l_p) mod *m*) =0. Jika tidak sama dengan nol (0), maka l_p harus ditambah sehingga memenuhi syarat ((2 * l_p) mod *m*) =0.

Kata Kunci: Fungsi X-OR, Fungsi XNOR, Hill Cipher, Kriptografi

1. PENDAHULUAN

Pertukaran informasi menjadi hal yang sangat penting pada era globalisasi saat ini. Begitu pentingnya pertukaran informasi, tentunya harus disertai dengan keamanan informasi (*information security*). Keamanan informasi yang berkaitan dengan penggunaan komputer tidak dapat dipisahkan dari kriptografi. Keamanan informasi yang dimaksud meliputi kerahasiaan, integritas data, otentikasi dan penyangkalan. Kriptografi tidak hanya menyediakan alat untuk keamanan pesan, tetapi juga sekumpulan teknik yang digunakan untuk menjamin kerahasiaan informasi tersebut.

Kriptografi hill cipher merupakan salah satu teknik kriptografi klasik yang cukup kuat karena menggunakan operasi matriks dan modulo yang cukup kompleks. Pada awalnya kunci matriks yang digunakan haruslah matriks persegi

yang memiliki *invers*, namun perkembangan pengetahuan memungkinkan penggunaan matriks persegi panjang dengan *pseudo invers* sebagai matriks kunci. Penggunaan matriks persegi panjang menghasilkan chiperteks yang lebih panjang dibandingkan plainteksnya.

Penggunaan sebuah matriks kunci memungkinkan kripnatalis untuk menemukan hubungan linier antara chiperteks dengan plainteksnya. Karena itu, Dalam penelitian ini penggunaan matriks persegi panjang akan dimodifikasi dengan menambahkan operasi biner yang diharapkan menghasilkan chiperteks yang lebih rumit serta menyulitkan kriptanalis untuk menemukan kuncinya.

Penulis melakukan penelitian terhadap salah satu algoritma kriptografi yaitu algoritma hill cipher dengan kunci matriks persegi panjang. Dari penelitian ini, penulis mengidentifikasi masalah diantaranya:

1. Apakah algoritma hill cipher dengan kunci matriks persegi panjang dapat dikombinasikan dengan operasi biner?
2. Apakah modifikasi ini dapat menghasilkan cipherteks yang lebih rumit?

Pada penelitian ini, penulis membatasi pembahasan hanya pada masalah berikut ini:

1. Plainteks berupa huruf, angka dan symbol tertentu sesuai dengan data yang ada pada table 1. korespondensi karakter dengan angka desimal.
2. Aplikasi digunakan untuk melakukan enkripsi dan deskripsi data dalam bentuk file text.
3. Data-data yang berupa huruf, angka dan simbol dikonversikan khusus pada Z_p. Untuk contoh dan aplikasi semua dioperasikan atas bilangan modulo 95.
4. Keamanan informasi yang dibahas hanya tentang kerahasiaan

Tabel 1. Korespondensi Karakter Dengan Angka Desimal

Karakter	Nilai Konversi	Karakter	Nilai Konversi	Karakter	Nilai Konversi	Karakter	Nilai Konversi	Karakter	Nilai Konversi
A	0	T	19	M	38	5	57	}	76
B	1	U	20	N	39	6	58	\	77
C	2	V	21	O	40	7	59	|	78
D	3	W	22	P	41	8	60	`	79
E	4	X	23	Q	42	9	61	~	80
F	5	Y	24	R	43	spasi	62	!	81
G	6	Z	25	S	44	,	63	@	82
H	7	a	26	T	45	<	64	#	83
I	8	b	27	U	46	.	65	$	84
J	9	c	28	V	47	>	66	%	85
K	10	d	29	W	48	/	67	^	86
L	11	e	30	X	49	?	68	&	87
M	12	f	31	Y	50	;	69	*	88
N	13	g	32	Z	51	:	70	(	89
O	14	h	33	0	52	‘	71	)	90
P	15	i	34	1	53	“	72	-	91
Q	16	j	35	2	54	[	73	_	92
R	17	k	36	3	55	{	74	=	93
S	18	l	37	4	56	]	75	+	94

2. KAJIAN LITERATUR

A. Kriptografi Hill Cipher

Hill cipher merupakan salah satu algoritma kriptografi simetris yang menggunakan matriks sebagai kuncinya. Proses enkripsi dimulai dengan mengkonversikan *plaintext* kedalam angka, selanjutnya angka-angka tersebut dikelompokkan menjadi beberapa blok, dimana masing-masing blok terdiri dari *m* anggota sesuai dengan ordo matriks kunci $K_{(mxm)}$. *Chipertext* akan didapatkan dengan menggunakan persamaan C=K*P.

Proses deskripsi diawali dengan mengkonversikan *chipertext* kedalam angka. Seperti halnya pada proses enkripsi, angka-angka tersebut dikelompokkan menjadi beberapa blok dengan anggota masing-masing blok sebanyak *m*, lalu dicari *plaintext* dengan persamaan $P = K^{-1} * C$

Matriks kunci yang dipilih haruslah matriks persegi yang invertible terhadap modulo *p*, karena matriks inversnya akan digunakan pada proses deskripsi. Jika tidak memiliki invers, maka *chipertext* tidak dapat dikembalikan kedalam bentuk

plaintext. Karena itu, algoritma hill cipher hanya bisa menggunakan kunci matriks persegi. Seiring dengan perkembangan pengetahuan, matriks persegi panjang dapat digunakan sebagai kunci jika matriks persegi panjang tersebut memiliki *pseudo-invers* (invers semu).

B. Aritmatika Modulo

Aritmatika Modulo merupakan operasi yang menghasilkan sisa bagi bulat. Contoh x sebuah bilangan bulat dan p bilangan bulat yang lebih besar dari 0. Operasi x modulo p menghasilkan sisa, jika x dibagi p. bilangan p disebut modulus atau modulo.

$x \bmod p = r$ sedemikian sehingga:

$x = p * q + r$ dengan $0 \leq r < p$

$r = x - p * q = x \bmod p$

Jika x dan y adalah bilangan bulat dan p bilangan bulat > 0, maka x kongruen dengan y dalam modulo p, jika p habis membagi $x - y$

$x \equiv y \pmod p$ jika $(x - y) \bmod p = 0$

sehingga didapatkan hubungan
$x = y + z * p$ dimana z merupakan bilangan bulat

Berdasarkan definisi diatas, didapatkan:
$x \bmod p = r$ sebagai $x \equiv r \pmod p$ (2.1)

Sebuah bilangan bulat terbesar yang membagi habis beberapa bilangan disebut faktor persekutuan Terbesar (*fpb*) atau *Greatest Common Divisor* (*gcd*). Jika z adalah pembagi bulat terbesar dua buah bilangan x dan y, maka dinyatakan bahwa $fpb(x,y) = z$.

fpb dari dua buah bilangan dapat dicari menggunakan Algoritma euclide. Misalkan untuk menghitung $fpb(x,y)$ dengan $x \geq y$

1. Jika $y = 0$ maka $fpb(x,y) = x$
   Jika $y \neq 0$ lanjut langkah kedua
2. Bagilah x dengan y dan r sebagai sisa bagi bulatnya
3. Ganti x dengan y, dan ganti y dengan r lalu kembali ke langkah pertama

$x = y * q + r, \quad 0 \leq r < a$

dari algoritma tersebut didapat:

$$
\begin{aligned}
fpb(r_0,r_1) \rightarrow\ & r_0 = r_1 * q_1 + r_2 \\
& r_1 = r_2 * q_2 + r_3 \\
& r_2 = r_3 * q_3 + r_4 \\
& \vdots \\
& r_{m-2} = r_{m-1} * q_{m-1} + r_p \\
& \mathbf{r_{m-1} = r_m * q_m + 0} \quad \mathbf{(2.2)}
\end{aligned}
$$

Sehingga didapat

$$
\begin{aligned}
\mathbf{fpb(r_0,r_1)} &= \mathbf{fpb(r_1,r_2)} \\
&= \mathbf{fpb(r_2,r_3)} \\
&= \mathbf{\ldots = fpb(r_{m-1},r_m)} \\
&= \mathbf{r_m} \qquad \mathbf{(2.3)}
\end{aligned}
$$

Dua bilangan yang mempunyai *fpb* = 1 disebut relatif prima. Sebuah bilangan x mempunyai invers perkalian x^{-1} sehingga $x * x^{-1} = 1$. bilangan x mempunyai invers terhadap modulo p jika $fpb(x,p) = 1 = 1 \bmod p$ (x dan p dikatakan relatif prima). Misalkan $x \in Z_p$. Invers terhadap pergandaan dari x adalah $x^{-1} \in Z_p$ sedemikian hingga

$$\mathbf{x * x^{-1} \equiv 1 \pmod p = 1} \qquad \mathbf{(2.4)}$$

Nilai *fpb* dari dua buah bilangan dapat ditentukan menggunakan algoritma euclide, juga dapat menentukan apakah suatu bilangan memiliki invers terhadap Z_p atau tidak, tapi belum dapat menentukan nilai inversnya. Berdasarkan algoritma euclide dengan $fpb(x,p) = 1$ dan persamaan (2.2) maka didapat:

$t_0 = r_{m+1} = 0$

$t_1 = r_m = 1 \rightarrow r_{m-2} = r_{m-1} * q_{m-1} + r_m$

$\Leftrightarrow r_m = r_{m-2} - r_{m-1} * q_{m-1}$

$$\mathbf{t_n = t_{n-2} - t_{n-1} * q_{n-1} \pmod{r_o}} \qquad \mathbf{(2.5)}$$

jadi untuk $0 \leq n \leq m$,

$r_n \equiv t_n * r_1 \pmod{r_0}$

Hal ini dapat dibuktikan dengan induksi matematika. $n = i$-1 dan $n = i$-2, untuk $i \geq 2$ akan dibuktikan pernyataan ini benar untuk $n = i$

$r_{i-2} \equiv t_{i-2} * r_1 \pmod{r_0}$

$r_{i-1} \equiv t_{i-1} * r_1 \pmod{r_0}$ sehingga didapatkan:

$$
\begin{aligned}
r_1 &\equiv r_{i-2} - r_{i-1} * q_{i-1} \\
&\equiv t_{i-2} * r_1 - t_{i-1} * r_1 * q_{i-1} \pmod{r_0} \\
&\equiv (t_{i-2} - t_{i-1} * q_{i-1}) * r_1 \pmod{r_0} \\
&\equiv t_i * r_1 \pmod{r_0}
\end{aligned}
$$

Dari pembuktian ini dapat disimpulkan pernyataan terbukti untuk semua n, Jika $fpb(r_0,r_1) = 1$, maka $r_m = 1$ sehingga didapat: $r_m = 1 \equiv t_m * r_1 \pmod{r_0}$. Dengan melihat bentuk $1 \equiv t_m * r_1$, berarti

$$\mathbf{t_m = r_1^{-1} \bmod r_0} \qquad \mathbf{(2.6)}$$

C. Matriks

A adalah matriks $m \times r$ dan B adalah matriks $k \times n$. Hasil kali AB

terdefinisi jika $r = k$, dan *AB* matriks $m \times n$. Secara umum perkalian matriks tidak bersifat komutatif

Jika *AB* = *C*, maka: $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{in}b_{nj}$

$$a_{ij} = \sum_{t=1}^{n} a_{it}b_{tj} \quad \begin{matrix} \text{dimana } i = 1,2,\ldots,m \\ j = 1,2,\ldots,n \end{matrix} \qquad (2.7)$$

Diberikan matriks *A*,*B*, dan *C*. Dengan menganggap bahwa ukuran-ukuran matriks adalah sedemikian sehingga operasi perkalian matriks dapat dilakukan, maka berlaku

A(BC) = (AB)C ➔ *Sifat assosiatif*

Matriks *A* berukuran $m \times n$, maka transpos dari matriks *A* didefinisikan dengan matriks $n \times m$ dinotasikan dengan A^T yang setiap kolom dari matriks *A* menjadi baris dari matriks A^T .

$$A^T = [a_{ji}] = \begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix}$$

Konjugat dari *A*, ditulis $\overline{A}$, merupakan matriks yang dibentuk dengan menegasikan bagian imajiner setiap entri *A*, jadi $\overline{A} = [\overline{a}_{ij}]$. **Transpos konjugat** dari *A* didefinisikan $A^H = (\overline{A})^T$. Matriks A dikatakan **hermitian** jika $A^H = A$.

Diberikan dua matriks $A = [a_{ij}]$ dan $B = [b_{ij}]$, dan skalar α dan β , maka berlaku:

a) $(\alpha A + \beta B)^T = \alpha A^T + \beta B^T$

b) $(AB)^T = B^T A^T$

c) $(A^T)^T = A$

d) $(\alpha A + \beta B)^H = \alpha A^H + \beta B^H$

e) $(AB)^H = B^H A^H$

f) $(A^H)^H = A$

Determinan adalah suatu fungsi tertentu yang menghubungkan suatu bilangan real dengan suatu matriks bujur sangkar.

Matriks dengan ordo lebih dari 2, determinannya dapat dicari menggunakan kofaktor. Bisa menggunakan baris atau kolom yang mana saja.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Rank dari matriks *A*, ditulis *rank(A)*, adalah banyaknya baris tak nol setelah *A* dibentuk ke dalam bentuk eselon baris. Suatu matriks $A_{m \times n}$ disebut *full column rank* jika *rank(A) = n* dan *full row rank* jika *rank(A) = m.*

Sebuah matriks persegi (*A*) disebut *invertible* (dapat dibalik) jika terdapat matriks lain (*B*) sehingga *AB* = *BA* = I dan matriks *B* disebut invers dari matriks *A*, ditulis $B = A^{-1}$ dimana I adalah matriks identitas. Matriks identitas adalah matriks persegi yang seluruh elemennya bernilai 0 kecuali pada elemen kolom dan baris yang sama bernilai 1.

$$a_{ij} = \begin{cases} \mathbf{0}, \textbf{ jika } i \neq j \\ \mathbf{1}, \textbf{jika } i = j \end{cases} \qquad \textbf{(2.8)}$$

$$A_{m\,x\,m} = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Matriks B_{nxm} disebut *pseudo invers* atau *p-invers* (matriks semu) dari matriks persegi panjang A_{mxn}, jika memenuhi syarat-syarat berikut ini:

1. *ABA = A*
2. *BAB = B*
3. $(AB)^H = AB$
4. $(BA)^H = BA$

$A^H = (\overline{A})^T$ yaitu konjugat transpos dari matrik *A*. *p-invers* dari matriks *A* dinotasikan dengan $A^{\#}$. *Pseudo invers* matriks A_{mxn} ($A^{\#}$) didapatkan dengan ketentuan sebagai berikut:

1. Jika $m \geq n$ dan matriks *A* mempunyai *full column rank*, maka
   $A^{\#} = (A^H A)^{-1} A^H$
2. Jika m < n dan matriks *A* mempunyai *full row rank*, maka
   $A^{\#} = A^H (A A^H)^{-1}$

D. Operasi Biner

Bilangan biner merupakan bilangan berbasis 2 yang terdiri dari angka 1 dan 0. Sedangkan bilangan yang biasa ditemukan sehari-hari yaitu angka 0 – 9 disebut bilangan desimal. Konversi bilangan desimal ke bilangan biner dengan cara pembagian 2 dan diambil sisa baginya. Pembagian dilakukan hingga hasil bagi 0. Untuk bilangan

binernya sendiri diambil dari sisa bagi dengan urutan paling akhir ke paling awal.

Operasi biner adalah operasi antara dua bit bilangan biner. Operasi biner yang paling banyak digunakan adalah fungsi logika NOT, AND, OR, XOR dan XNOR.

Fungsi XOR memberikan nilai true (1) jika kedua bit berbeda nilainya dan memberikan nilai false (0) jika kedua bit bernilai sama.

1 XOR 1 = 0 1 XOR 0 = 1
0 XOR 1 = 1 0 XOR 0 = 0

Fungsi XNOR kebalikan dari fungsi XOR atau NOT XOR, yaitu memberikan nilai true (1) jika kedua bit bernilai sama dan memberikan nilai false (0) jika kedua bit berbeda nilainya.

1 XNOR 1 = 1 1 XNOR 0 = 0
0 XNOR 1 = 0 0 XNOR 0 = 1

Operasi biner pada bilangan desimal dilakukan dengan cara mengkonversi bilangan desimal tersebut ke bilangan biner terlebih dahulu, karena operasi biner ini hanya dapat dilakukan pada bilangan biner.

E. Tinjauan Studi

Penelitian terdahulu tentang kriptografi hill cipher yang menjadi acuan dalam penelitian ini dapat dilihat pada Tabel 2.

Tabel 2. Penelitian Terdahulu

Peneliti, judul, Tahun Penelitian	Hasil Penelitian	Perbedaan dengan penelitian ini
V. Umakanta Sastry, N. Ravi Shankar And S. Durga Bhavani, A Modified Hill Cipher Involving Interweaving And Iteration (2010)	Proses enkripsi dilakukan dengan perkalian matriks kunci dan *plaintext* secara berulang-ulang.	Matriks kunci yang digunakan tidak terbatas pada matriks persegi saja, tapi juga matriks persegi panjang.
M. Nordin A. Rahman, A. F. A. Abidin, Mohd Kamir Yusof, N. S. M. Usop, Cryptography: A New Approach Of Classical Hill Cipher (2013)	Proses deskripsi tidak menggunakan invers matriks karena proses enkripsi dan deskripsi menggunakan kunci matriks yang sama. Pada paper ini juga digunakan kunci matriks acak pada proses enkripsi dan deskripsinya	Peneliti menggunakan matriks *invers* atau *pseudo invers* untuk proses deskripsi
Alz Danny Wowor, Modifikasi Kriptografi Hill Cipher Menggunakan Convert Between Base (2013)	Proses enkripsi dan deskripsi menggunakan n-matriks kunci, modulo 127 dan konversi biner pada *chipertext*	Peneliti tidak menggunakan banyak matriks kunci, karena Penggunaan n-matriks kunci menyulitkan dalam hal manajemen kunci. Semakin banyak matriks kunci yang digunakan, semakin rumit untuk mengingat atau memanajemen matriks kuncinya. Bilangan biner hanya digunakan untuk konversi *chipertext* kedalam bentuk deret biner tanpa melakukan operasi biner
Alz Danny Wowor, Penggunaan	Pembangkit kunci menggunakan kunci	Peneliti tidak menggunakan determinam matriks

Determinan Polinomial Matriks Dalam Modifikasi Kriptografi Hill Chiper (2014)	tambahan determinan dari matriks polinomial berupa persamaan polinomial. Selain itu juga menggunakan *convert between base* (CBB) untuk mengkonversi *chipertext* menjadi bilangan biner.	polynomial.meskipun peneliti menggunakan konversi bilangan biner, tapi peneliti menambahkan operasi biner
M.K. Viswanath dan M. Ranjith Kumar, A Public Key Cryptosystem Using Hill's Cipher (2015)	Kunci yang digunakan terdiri dari kunci umum (*public key)* dan kunci rahasia (*private key*). Kunci umum yang digunakan adalah matriks persegi yang memiliki nilai eigen irrational. Sedangkan kunci rahasianya terdiri dari sebuah bilangan bulat dan matriks persegi panjang.	Peneliti tidak menggunakan kunci umum, dikarenakan penggunaan kunci umum tidak menambah kerumitan pada *chipertext* yang dihasilkan. selain itu, penggunaan kunci tambahan (bilangan bulat dan kunci umum) Tidak menambah kerumitan kripnatalis dalam menemukan matriks kunci untuk mendeskripsi *chipertext*. Meskipun menggunakan beberapa kunci, namun untuk proses deskripsi hanya menggunakan sebuah matriks yang merupakan invers matriks kunci. Kunci yang didistribusikan bukan hanya kunci umum, tapi juga kunci rahasianya
Khairani puspita dan M. Rhifky Wayahdi, Analisis Kombinasi Metode Caesar Cipher, Vernam Cipher Dan Hill Cipher Dalam Proses Kriptografi (2015)	Mengkombinasikan 3 metode dimana *plaintext* diproses menggunakan metode Caesar cipher. Hasilnya diproses lagi menggunakan metode vernam cipher dan terakhir diproses menggunakan hill cipher	Peneliti tidak melakukan kombinasi dengan metode Caesar chipper. Matriks kunci yang digunakan adalah matriks persegi sedangkan peneliti menggunakan matriks persegi panjang yang memberikan varian matriks kunci lebih banyak, juga menghasilkan *chipertext* yang lebih banyak dibanding *plaintext*

3. METODOLOGI PENELITIAN

A. Analisis Kriptografi Hill Cipher

Algoritma kriptografi hill cipher terdiri dari 3 proses, yaitu proses inisialisasi kunci, proses enkripsi dan proses deskripsi. Semua proses perhitungan dioperasikan atas bilangan modulo *p*.

Inisialisasi kunci dimaksudkan untuk memeriksa matriks yang digunakan sebagai kunci. Matriks kunci harus memiliki *invers* atau *pseudo invers* yang akan digunakan untuk proses deskripsi. Jika tidak memiliki *invers* atau pseudo *invers*, maka tidak dapat digunakan sebagai matriks kunci.

Proses enkripsi pada kriptografi hill cipher dengan *pseudo invers* terdiri dari beberapa tahap, yaitu:

1. Hitung panjang *plaintext l,* jika *l* mod *r* ≠ 0, maka tambahkan karakter hingga didapatkan *l* mod *r* = 0. Dimana *r* adalah rank matriks kunci.
2. Konversi *plaintext* kedalam bilangan desimal sesuai data pada tabel 1
3. Bagi *plaintext* kedalam blok-blok P_1, P_2, ... ,P_i dimana $i = 1 \ldots \frac{l}{r}$ dengan masing-masing blok terdiri dari *r* elemen.
4. Hitung C_1, C_2, ... ,C_i dengan ketentuan:
   a. $C_i = (A (P_i)^T)^T$ jika $m \geq n$

b. $C_i = P_i\ A$ jika $m < n$

5. Gabungkan $C_1, C_2, \ldots, C_i$ lalu konversikan ke dalam bentuk karakter sesuai tabel 1 sehingga didapatkan sebuah *ciphertext*.

Proses deskripsi pada algoritma hill cipher dengan *pseudo invers* terdiri dari beberapa tahap, yaitu:

1. Konversikan *ciphertext* kedalam bentuk bilangan desimal sesuai tabel 1
2. Bagi *ciphertext* kedalam blok-blok $C_1, C_2, \ldots, C_i$ dengan $i = 1 \ldots \frac{l}{j}$ dimana setiap blok terdiri dari j elemen.
   $j = m$ Jika $m \geq n$, $j = n$ Jika $m < n$
3. Hitung $P_1, P_2, \ldots, P_i$ dengan ketentuan:
   a. $P_i = (A^{\#} (C_i)^T)^T$ jika $m \geq n$
   b. $P_i = C_i\ A^{\#}$ jika $m < n$

Gabungkan $P_1, P_2, \ldots, P_i$ lalu konversikan bilangan ke dalam bentuk karakter sesuai tabel 1 sehingga didapat *plaintext*.

B. Perancangan Modifikasi Pada Kriptografi Hill Cipher

Modifikasi kriptografi hill cipher dengan *pseudo invers* dilakukan pada proses enkripsi dan deskripsi. Proses enkripsi dan deskripsi dilakukan dengan menambahkan operasi biner X-OR, XNOR dan partisi blok biner. Selain matriks kunci, pada penelitian ini menambahkan sebuah kunci bilangan bulat.

Proses enkripsi pada kriptografi hill cipher yang telah dimodifikasi terdiri dari:

1. menentukan sebuah bilangan (k), $0 \leq k \leq 255$
2. Konversi k kedalam bentuk biner 8 bit
3. Hitung panjang plainteks l, jika (l x *2)* mod $r \neq 0$, tambahkan karakter hingga didapatkan (l x *2)* mod $r = 0$. Dimana r adalah rank matriks kunci.
4. Korespondensikan plainteks kedalam bilangan desimal sesuai dengan data pada tabel 1.
5. Plainteks yang berbentuk bilangan desimal dikonversikan kedalam bentuk biner 8 bit
6. Gunakan fungsi logika X-OR untuk setiap karakter pada plainteks biner dengan bilangan biner K
7. Konversikan hasil proses 5 kedalam bilangan desimal, dimana konversi dilakukan dengan mengkonversikan 4 bit biner kedalam bilangan desimal.
8. Partisi plainteks kedalam blok-blok $P_1, P_2, \ldots, P_i$ dimana $i = 1 \ldots \frac{(l)}{r}$ dengan masing-masing blok terdiri dari r elemen.
9. Hitung $C_1, C_2, \ldots, C_i$ dengan ketentuan:
   a. $C_i = (A\ (P_i)^T)^T$ jika $m \geq n$, matriks kunci *full coloumn rank*
   b. $C_i = P_i\ A$ jika $m < n$, matriks kunci *full row rank*
10. Konversi $C_1, C_2, \ldots, C_i$ kedalam bilangan biner 8 bit.
11. Gunakan fungsi logika XNOR untuk setiap 8 bit biner hasil proses 10 dengan bilangan biner K
12. Konversikan hasil proses 11 kedalam bilangan desimal, dimana konversi dilakukan dengan mengkonversikan 4 bit biner kedalam bilangan desimal.
13. Partisi bilangan desimal kedalam blok-blok $P_1, P_2, \ldots, P_i$ dengan $i = 1 \ldots \frac{l}{j}$ dimana setiap blok terdiri dari r elemen. Jika elemen P_i kurang dari r, maka tambahkan angka pada blok pertama elemen pertama sampai elemen ke n hingga jumlah r elemen terpenuhi
14. Hitung $C_1, C_2, \ldots, C_i$ dengan ketentuan:
    a. $C_i = (A\ (P_i)^T)^T$ jika $m \geq n$, matriks kunci *full coloumn rank*
    b. $C_i = P_i\ A$ jika $m < n$, matriks kunci *full row rank*

Konversikan hasil proses 14 ke dalam bentuk karakter sesuai tabel 1 untuk mendapatkan sebuah ciphertext.

Proses deskripsi pada algoritma hill cipher dengan *pseudo invers* yang telah dimodifikasi terdiri dari beberapa tahap, yaitu:

1. Konversikan *ciphertext* kedalam bentuk bilangan desimal sesuai tabel 1.
2. Partisi bilangan desimal *ciphertext* kedalam blok-blok $C_1, C_2, \ldots, C_i$ dengan $i = 1 \ldots \frac{l}{j}$ dimana setiap blok terdiri dari j elemen.

a. $j = m$, jika matriks kunci *full coloumn rank* ($m \geq n$)
b. $j = n$, jika matriks kunci *full row rank* ($m < n$)

3. Hitung $P_1, P_2, \ldots, P_i$ dengan ketentuan:
   a. $P_i = (A^{\#} (C_i)^T)^T$ jika $m \geq n$, matriks kunci *full coloumn rank*
   b. $P_i = C_i \; A^{\#}$ jika $m < n$, matriks kunci *full row rank*
4. Hitung (l mod j), jika (l mod j) ≠ 0, maka hapus bilangan terakhir hingga (l mod j) = 0
5. Konversikan $P_1, P_2, \ldots, P_i$ hasil proses 4 kedalam bilangan biner 4 bit
6. Hasil proses 5 digabungkan dan dipartisi menjadi beberapa blok dengan masing-masing blok terdiri dari 8 bit.
7. Gunakan fungsi logika XNOR untuk setiap blok pada proses no 6 dengan bilangan biner *K*
8. Konversi bilangan biner hasil proses 7 kedalam bilangan desimal.
9. Partisi bilangan desimal hasil proses 8 kedalam blok-blok $C_1, C_2, \ldots, C_i$ dengan $i = 1 \ldots \frac{l}{j}$ dimana setiap blok terdiri dari j elemen.
   a. $j = m$, jika matriks kunci *full coloumn rank* ($m \geq n$)
   b. $j = n$, jika matriks kunci *full row rank* ($m < n$)
10. Hitung $P_1, P_2, \ldots, P_i$ dengan ketentuan:
    a. $P_i = (A^{\#} (C_i)^T)^T$ jika $m \geq n$, matriks kunci *full coloumn rank*
    b. $P_i = C_i \; A^{\#}$ jika $m < n$, matriks kunci *full row rank*
11. Konversikan $P_1, P_2, \ldots, P_i$ hasil proses 10 kedalam bilangan biner 4 bit.
12. Hasil proses 11 digabungkan dan dipartisi menjadi beberapa blok dengan masing-masing blok terdiri dari 8 bit.
13. Gunakan fungsi logika X-OR untuk setiap blok pada proses no 12 dengan bilangan biner *K*
14. Konversi bilangan biner hasil proses 13 kedalam bilangan desimal.

Korespondensikan bilangan desimal hasil proses 14 dengan karakter sesuai tabel 1 sehingga didapatkan plainteks.

4. PEMBAHASAN

A. Simulasi Sistem

Tentukan sebuah matriks kunci $A_{(3x5)}$ terdiri dari 3 baris, 5 kolom

$$A = \begin{bmatrix} 3 & 4 & 7 & 6 & 4 \\ 7 & 5 & 8 & 0 & 1 \\ 9 & 1 & 2 & 3 & 9 \end{bmatrix}$$

jumlah baris < jumlah kolom
Rank (A)=3

Matriks A *full row rank* karena rank (A) sama dengan jumlah baris yaitu 3, maka *pseudo invers* matriks A adalah $\boldsymbol{A^{\#} = A^T (A\,A^T)^{-1}}$

$$A^T = \begin{bmatrix} 3 & 7 & 9 \\ 4 & 5 & 1 \\ 7 & 8 & 2 \\ 6 & 0 & 3 \\ 4 & 1 & 9 \end{bmatrix}$$

$$A\,A^T = \begin{bmatrix} 31 & 6 & 4 \\ 6 & 44 & 93 \\ 4 & 93 & 81 \end{bmatrix}$$

Determinan $|A^T A| = 54$
fpb (p,det$|A^T A|$) ➔ *fpb* (95,54) ➔
95 = (54*1) + 41
54 = (41*1) + 13
41 = (13*3) + 2

13 = (2*6) + 1
2 = (1*2) + 0

Fpb (95,54) = 1, berarti memiliki *invers*

Berdasarkan persamaan (2.2) didapat:
$t_0 = 0, t_1 = 1, q_1 = 1, q_2 = 1, q_3 = 3, q_4 = 6$

$t_2 = t_0 - t_1 * q_1 = 0 - 1*1 = -1$
$t_3 = t_1 - t_2 * q_2 = 1 - (-1*1) = 2$
$t_4 = t_2 - t_3 * q_3 = (-1) - (2*3) = -7$
$t_5 = t_3 - t_4 * q_4 = (2) - (-7*6) = 44$

invers 54 terhadap 95 adalah 44

$$adjoin(A\,A^T) = \begin{bmatrix} 45 & 76 & 2 \\ 76 & 25 & 86 \\ 2 & 86 & 93 \end{bmatrix}$$

$(A\,A^T)^{-1}$ adalah *invers* dari matriks $A\,A^T$,
$(A\,A^T)^{-1}$ = *invers* (*det*) * *adjoin* ($A\,A^T$)

$$(A\,A^T)^{-1} = \begin{bmatrix} 80 & 19 & 88 \\ 19 & 55 & 79 \\ 88 & 79 & 7 \end{bmatrix}$$

$A^{\#}$ adalah *pseudo invers* dari matriks kunci *A*

$A^{\#} = A^{T} * (A\,A^{T})^{-1}$

$$A^{\#} = \begin{bmatrix} 25 & 13 & 25 \\ 28 & 50 & 89 \\ 33 & 66 & 27 \\ 79 & 66 & 74 \\ 86 & 82 & 19 \end{bmatrix}$$

Syarat-syarat *p-invers*:

1. $A\,A^{\#}A = A$ dipenuhi
2. $A^{\#}A\,A^{\#} = A^{\#}$ dipenuhi
3. $(A\,A^{\#})^{*} = A\,A^{\#}$ dipenuhi
4. $(A^{\#}A)^{*} = A^{\#}A$ dipenuhi

Dari hasil perhitungan menunjukkan matriks *A* memenuhi syarat untuk digunakan sebagai matriks kunci.

Proses enkripsi pada kriptografi hill cipher yang telah dimodifikasi diawali dengan menentukan nilai sembarang *k.* Pada simulasi ini dipilih *k*=103 dan menggunakan matriks *A* sebagai matriks kunci.

plaintext: Tesis NURI 2015

k = 103 = 01100111, *rank* (*A*) = 3

panjang *plaintext l*=15, karena (*l* x 2*)* mod *r* = 0, maka tidak perlu penambahkan karakter.

Korespondensikan *plaintext* kedalam bilangan desimal sesuai table 1 sehingga didapat

P = [19, 30, 44, 34, 44, 62, 13, 20, 17, 8, 62, 54, 52, 53, 57]

Konversi bilangan desimal *plaintext* kedalam bilangan biner. Hasil konversi dapat dilihat pada Tabel 3.

Tabel 3. Konversi Bilangan Desimal *Plaintext* Kedalam Bilangan Biner

Plain text	Bilangan Biner	*Plain text*	Bilangan Biner
19	00010011	17	00010001
30	00011110	8	00001000
44	00101100	62	00111110
34	00100010	54	00110110
44	00101100	52	00110100
62	00111110	53	00110101
13	00001101	57	00111001
20	00010100		

Kemudian lakukan proses enkripsi menggunakan fungsi logika X-OR antara bilangan biner *plaintext* dengan bilangan biner *k* seperti pada Tabel 4.

Tabel 4. Proses Enkripsi Menggunakan Fungsi X-OR

Plaintext	**Kunci *k***	**X-OR**
00010011	01100111	**01110100**
00011110	01100111	**01111001**
…		
00110101	01100111	**01010010**
00111001	01100111	**01011110**

Partisi setiap blok biner hasil fungsi X-OR menjadi 4 bit dan konversi ke bilangan desimal. Hasil konversi dapat dilihat pada Tabel 5.

Tabel 5. Proses Enkripsi Konversi Bilangan Biner Hasil Fungsi XOR Kedalam Bilangan Desimal

Biner	**Desimal**
0111	7
0100	4
…	
0101	5
1110	14

Dari konversi desimal pada tabel 5 didapatkan deret bilangan desimal (*P*) sebagai berikut ini:

[7, 4, 7, 9, 4, 11, 4, 5, 4, 11, 5, 9, 6, 10, 7, 3, 7, 6, 6, 15, 5, 9, 5, 1, 5, 3, 5, 2, 5, 14]

Matriks kunci *A* merupakan matriks *full row rank*, maka digunakan persamaan $C_i = P_i\ A$

Panjang *plaintext* (*l*) = 30

rank (*A)* = 3

Deret bilangan desimal (*P*) dipartisi menjadi beberapa blok, dmana masing-masing blok terdiri dari 3 elemen (sesuai dengan jumlah baris matriks kunci *A*) sehingga didapat 10 blok yaitu P_1 - P_{10}

P_1 = [7, 4, 7]
P_2 = [9, 4, 11]
P_3 = [4, 5, 4]
P_4 = [11, 5, 9]
P_5 = [6, 10, 7]
P_6 = [3, 7, 6]
P_7 = [6, 15, 5]
P_8 = [9, 5, 1]
P_9 = [5, 3, 5]
P_{10} = [2, 5, 14]

Matriks kunci yang digunakan adalah matriks full row rank, sehingga $C_i = P_i\,A$

$$C_1 = P_1\,A = \begin{bmatrix} 7 & 4 & 7 \end{bmatrix} * \begin{bmatrix} 3 & 4 & 7 & 6 & 4 \\ 7 & 5 & 8 & 0 & 1 \\ 9 & 1 & 2 & 3 & 9 \end{bmatrix} \bmod 95$$

$$= \begin{bmatrix} 17 & 55 & 0 & 63 & 0 \end{bmatrix}$$

…

$C_{10} = P_{10} A = [2 \ 5 \ 14] * \begin{bmatrix} 3 & 4 & 7 & 6 & 4 \\ 7 & 5 & 8 & 0 & 1 \\ 9 & 1 & 2 & 3 & 9 \end{bmatrix} \text{ mod} 95$

$= [72 \ 47 \ 82 \ 54 \ 44]$

- Konversikan elemen matriks bilangan desimal kedalam bilangan biner 8 bit yang dapat dilihat pada tabel 6.

Tabel 6. Proses Enkripsi Konversi Bilangan Desimal Hasil Perkalian Matriks Pertama Ke Biner

C_i	Bilangan Biner
17	00010001
55	00110111
…	
54	00110110
44	00101100

- Gunakan fungsi logika XNOR untuk setiap 8 bit biner pada tabel 6. dengan bilangan biner *k.* hasil dan proses ini dapat dilihat pada Tabel 7.

Tabel 7. Proses Enkripsi Menggunakan Fungsi XNOR

C_i	Kunci *k*	XNOR
00010001	01100111	**10001001**
00110111	01100111	**10101111**
…		
00110110	01100111	**10101110**
00101100	01100111	**10110100**

- Partisi bilangan biner hasil proses XNOR menjadi 4 bit, lalu konversikan kedalam bilangan desimal yang dapat dilihat pada Tabel 8.

Tabel 8. Proses Enkripsi Konversi Biner 4 bit Hasil Fungsi XNOR Kedalam Bilangan Desimal

Biner	Desimal
1000	8
1001	9
…	
1011	11
0100	4

- Partisi bilangan desimal pada tabel 8 kedalam blok-blok $P_1, P_2, \ldots, P_i$, i = 1 … $\frac{l}{n}$
- Hitung $C_1, C_2, \ldots, C_i$, $C_i = P_i A^{\#}$
- Deret bilangan desimal [8, 9, 10, 15, 9, 8, 10, 7, …, 12, 10, 10, 14, 11, 4]
- Partisi bilangan desimal ke dalam blok-blok $P_1, P_2, \ldots P_{10}$ dengan jumlah masing-masing blok 3 elemen, jika blok terakhir kurang dari 3 elemen, maka tambahkan deret bilangan desimal awal hingga jumlah elemen blok terakhir sama dengan 3, kemudian dikalikan dengan *A*

$C_1 = [8 \ 9 \ 10]$

$C_2 = [15 \ 9 \ 8]$

…

$C_{33} = [10 \ 14 \ 11]$

$C_{34} = [4 \ 8 \ 9]$

$C_1 = P_1 A = [8 \ 9 \ 10] * \begin{bmatrix} 3 & 4 & 7 & 6 & 4 \\ 7 & 5 & 8 & 0 & 1 \\ 9 & 1 & 2 & 3 & 9 \end{bmatrix} = [82 \ 87 \ 53 \ 78 \ 36]$

$C_2 = P_2 A = [15 \ 9 \ 8] * \begin{bmatrix} 3 & 4 & 7 & 6 & 4 \\ 7 & 5 & 8 & 0 & 1 \\ 9 & 1 & 2 & 3 & 9 \end{bmatrix} = [85 \ 18 \ 3 \ 19 \ 46]$

…

$C_{33} = P_{33} A = [10 \ 14 \ 11] * \begin{bmatrix} 3 & 4 & 7 & 6 & 4 \\ 7 & 5 & 8 & 0 & 1 \\ 9 & 1 & 2 & 3 & 9 \end{bmatrix} = [37 \ 26 \ 14 \ 93 \ 58]$

$C_{34} = P_{34} A = [4 \ 8 \ 1] * \begin{bmatrix} 3 & 4 & 7 & 6 & 4 \\ 7 & 5 & 8 & 0 & 1 \\ 9 & 1 & 2 & 3 & 9 \end{bmatrix} = [54 \ 65 \ 15 \ 51 \ 10]$

- Gabungkan $C_1, C_2, \ldots, C_{34}$ untuk mendapatkan sebuah *ciphertext* dalam bentuk desimal, sehingga didapat [82, 87, 53, 78, 36, 85, 18, 3, …, 20, 65, 15, 51, 10]

Korespondensikan *ciphertext* angka kedalam bentuk karakter sesuai table 1.1. sehingga didapatkan *ciphertext* yang terdiri dari 170 karakter

@&1|k%SDTu.$x&ha%v5;=UFHo'kiiN{}mH98J~!F'#yQ8UE']/3cSE}A`jkj]$zK5l+>2lJ#K9@3(JX{+<|XYE{=?v<TK1pTA]:g>WKns{aV ZM&B<"hcWG6`p&g%7BPbSH`B,J-42vnhbQ;YC?>ur,SQ4FD:"jlaO=62.PzK

Proses deskripsi dilakukan menggunakan persamaan $P_i = C_i \ A$ karena matriks kunci yang digunakan merupakan matriks *full row rank*. Proses deskripsi diawali dengan merubah *ciphertext* kedalam bentuk angka sesuai table 1.1. Partisi matriks *C* menjadi beberapa blok matriks yang masing-masing terdiri dari 3 elemen (sesuai dengan rank (*A*)) sehingga didapat C_1 sampai C_{20}

$C_1 = [82 \ 87 \ 53 \ 78 \ 36]$

$C_2 = [85 \quad 18 \quad 3 \quad 19 \quad 46]$

…

$C_{34} = [54 \quad 65 \quad 15 \quad 51 \quad 10]$

$$P_1 = C_1A = [82 \quad 87 \quad 53 \quad 78 \quad 36] * \begin{bmatrix} 25 & 13 & 25 \\ 28 & 50 & 89 \\ 33 & 66 & 27 \\ 79 & 66 & 74 \\ 86 & 82 & 19 \end{bmatrix} = [8 \quad 9 \quad 10]$$

…

$$P_{34} = C_{34}A = [54 \quad 65 \quad 15 \quad 51 \quad 10] * \begin{bmatrix} 25 & 13 & 25 \\ 28 & 50 & 89 \\ 33 & 66 & 27 \\ 79 & 66 & 74 \\ 86 & 82 & 19 \end{bmatrix} = [4 \quad 8 \quad 9]$$

Gabungkan P_1, P_2, ... ,P_{34} sehingga didapat deret bilangan desimal sebagai berikut: [8, 9, 10, 15, 9, 8, …, 14, 11, 4, 8, 9]

- Proses deskripsi ini menghasilkan 102 elemen. Hapus elemen terakhir hingga (l mod 5) = 0, maka 2 elemen terakhir 8 dan 9 dihapus.
- Selanjutnya konversi bilangan desimal tersebut ke deret bilangan biner 4 bit. Proses ini dapat dilihat pada Tabel 9.

Tabel 9. Proses Deskripsi Konversi Bilangan Desimal Hasil Perkalian Matriks Pertama Ke Biner

Desimal	Biner
8	1000
9	1001
…	
11	1011
4	0100

- Gabungkan dua blok deret biner menjadi satu blok deret biner, kemudian operasikan fungsi logika XNOR antara blok bilangan biner *ciphertext* dengan bilangan biner *k* seperti pada Tabel 10.

Tabel 10. Proses Deskripsi Menggunakan Fungsi XNOR

Ciphertext	**Kunci *k***	**XNOR**
10001001	01100111	**00010001**
10101111	01100111	**00110111**
…		
10101110	01100111	**00110110**
10110100	01100111	**00101100**

- Konversi deret biner ke bilangan desimal. Hasil konversi dapat dilihat pada Tabel 11.

Tabel 11. Proses Deskripsi Konversi Biner Hasil Fungsi XNOR Kedalam Bilangan Desimal

Biner	Desimal
00010001	17
00110111	55
….	
00110110	54
00101100	44

- Partisi bilangan desimal pada tabel 11 kedalam blok-blok dengan masing-masing blok terdiri dari 5 elemen sehingga terbentuk 10 blok

$C_1 = [17 \quad 55 \quad 0 \quad 63 \quad 0]$

$C_2 = [59 \quad 67 \quad 22 \quad 87 \quad 44]$

…

$C_{10} = [72 \quad 47 \quad 82 \quad 54 \quad 44]$

$$P_1 = C_1A = [17 \quad 55 \quad 0 \quad 63 \quad 0] * \begin{bmatrix} 25 & 13 & 25 \\ 28 & 50 & 89 \\ 33 & 66 & 27 \\ 79 & 66 & 74 \\ 86 & 82 & 19 \end{bmatrix} \text{ mod } 95$$

$$= [7 \quad 4 \quad 7]$$

…

$$P_{10} = C_{10}A = [72 \quad 47 \quad 82 \quad 54 \quad 44] * \begin{bmatrix} 25 & 13 & 25 \\ 28 & 50 & 89 \\ 33 & 66 & 27 \\ 79 & 66 & 74 \\ 86 & 82 & 19 \end{bmatrix} \text{ mod } 95$$

$$= [2 \quad 5 \quad 14]$$

Dari perkalian ini didapat deret bilangan desimal [7, 4, 7, 9, 4, 11, 4, 5, 4, 11, 5, 9, 6, 10, 7, 3, 7, 6, 6, 15, 5, 9, 5, 1, 5, 3, 5, 2, 5, 14]

- Konversi bilangan decimal tersebut kedalam bilangan biner 4 bit. Hasil proses ini dapat dilihat pada Tabel 12.

Tabel 12. Proses Deskripsi Konversi Bilangan Desimal Hasil Perkalian Matriks Kedua Ke Biner

Desimal	Biner
7	0111
4	0100
…	
14	1110

- Gabungkan 2 blok deret biner pada table 4.10 sehingga membentuk 8 bit biner, kemudian operasikan fungsi logika XOR dengan bilangan biner *k*. Hasil operasi biner dikonversi ke

bilangan desimal. operasi ini dapat dilihat pada Tabel 13.

Tabel 13. Proses Deskripsi Menggunakan Fungsi X-OR

Ciphertext	**Kunci *k***	**X-OR**	**Bilangan Desimal**
01110100	01100111	**00010011**	**19**
01111001	01100111	**00011110**	**30**
…			
01010010	01100111	**00110101**	**53**
01011110	01100111	**00111001**	**57**

Dari Tabel 13 didapat deret bilangan desimal [19, 30, 44, 34, 44, 62, 13, 20, 17, 8, 62, 54, 52, 53, 57]

- Korespondensikan bilangan desimal tersebut dengan karakter pada tabel 1. sehingga didapat *plaintext* sebagai berikut: **Tesis NURI 2015**

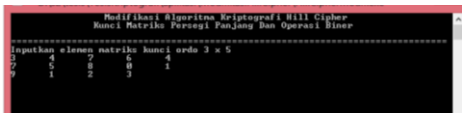
B. Simulasi Aplikasi

Proses perhitungan pada algoritma kriptografi hill cipher yang telah dimodifikasi ini cukup rumit dan membutuhkan waktu lama, karena itu penelitian ini dilengkapi dengan aplikasi untuk menggunakan algoritma kriptografi yang telah dimodifikasi ini.

Tampilan awal aplikasi menampilkan inputan matriks kunci yang akan digunakan, dimana user harus menginputkan jumlah baris dan jumlah kolom matriks kunci. Tampilan dapat dilihat pada gambar 1

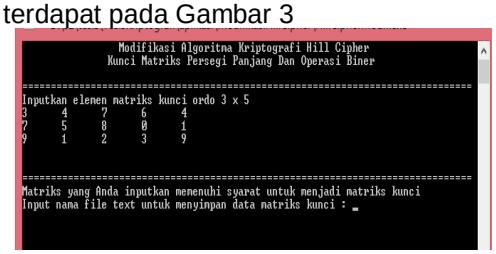
Gambar 1. Input Jumlah Baris dan Kolom Matriks Kunci

Pada tampilan berikutnya, user harus input elemen matriks kunci, setelah input sebuah elemen klik enter untuk input elemen berikutnya. Proses ini dapat dilihat pada Gambar 2.



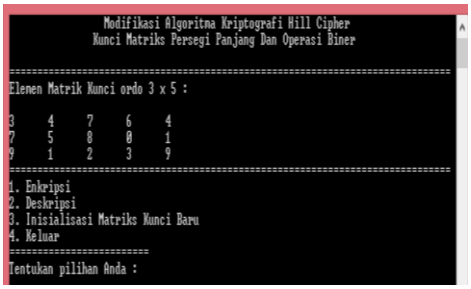
Gambar 2. Input Elemen Matriks Kunci

Selanjutnya aplikasi akan menghitung nilai *pseudo invers* matriks tersebut. Jika *pseudo invers*nya ditemukan, selanjutnya diperiksa apakah memenuhi syarat-syarat *pseudo invers* atau tidak. Jika memenuhi syarat, aplikasi meminta user menginputkan nama file text untuk menyimpan matriks kunci. Proses ini terdapat pada Gambar 3



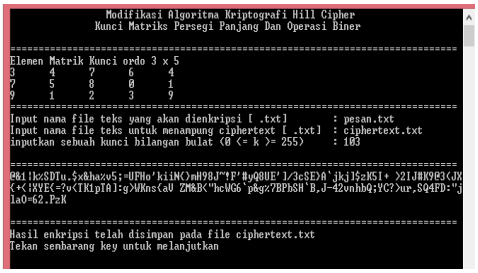
Gambar 3. Tampilan Penyimpanan Matriks Kunci

Setelah selesai proses input matriks kunci dan inisialisasi matriks kunci, selanjutnya aplikasi akan menampilkan menu utama. User dapat memilih menu dengan menginputkan nomor menu yang dipilih. Tampilan menu dapat dilihat pada Gambar 4.



Gambar 4. Tampilan Menu

1. Untuk melakukan proses enkripsi, user memilih meu 1. Data yang akan dienkripsi harus disiapkan terlebih dahulu dalam bentuk file txt. User input nama file text data *plaintext*, input nama file text untuk menampung hasil enkripsi (*ciphertext*) serta sebuah kunci bilangan bulat. Setelah klik enter, maka proses enkripsi dilakukan dan aplikasi akan menampilkan hasil proses enkripsi dan menyimpan hasil proses enkripsi pada file text yang user inputkan. Proses ini dapat dilihat pada Gambar 5.



Gambar 5. Tampilan Menu Enkripsi

Untuk melakukan proses deskripsi, user memilih menu 2. Ciphertext disiapkan terlebih dahulu dalam file text. User input nama file text tempat ciphertext berada, input file penampung hasil proses deskripsi (*plaintext*) serta sebuah kunci bilangan bulat. Setelah selesai inputan, aplikasi akan menampilkan hasil proses deskripsi. Proses ini dapat dilihat pada Gambar 6.

Gambar 6. Tampilan Menu Deskripsi

Untuk input matriks kunci baru, user memilih menu 3. Maka akan tampil kembali proses input-an baris dan kolom matriks serta elemen matriks seperti pada gambar 1. Sedangkan untuk keluar dari aplikasi, user memilih menu 4.

Berdasarkan hasil penelitian yang telah dilakukan, penggunaan matriks persegi panjang menambah varian jenis matriks yang bisa digunakan sebagai matriks kunci. Selain itu, penggunaan operasi biner X-OR dan XNOR, pembagian bit dan dua kali proses perkalian matriks, menghasilkan ciphertext yang jauh lebih panjang. Perbandingan hasil modifikasi algoritma hill cipher dengan algoritma hill cipher terdahulu dapat dilihat pada Tabel 14.

Penggunaan beberapa tahap enkripsi menjadikan *ciphertext* jauh lebih panjang dibandingkan plaintext-nya, lebih kuat dan matriks kunci pun sulit untuk diketahui dengan penyerangan persamaan linier. Tabel 15 memperlihatkan perbandingan *ciphertext* hasil proses enkripsi modifikasi hill cipher *pseudo invers* dan operasi biner dengan metode hill cipher terdahulu.

Tabel 14. Perbandingan Modifikasi Hill Cipher dengan Hill Cipher Terdahulu

Metode Hill Cipher	**Kunci**	**Panjang *Ciphertext***
Hill cipher klasik	Matriks persegi.	*Ciphertext = plaintext*
Hill cipher dengan *pseudo invers*	Matriks persegi atau matriks persegi panjang	*Ciphertext > plaintext*
Hill cipher dengan *pseudo invers* dan operasi biner X-OR dan X-NOR	Sebuah bilangan bulat dan matriks persegi atau matriks persegi panjang.	*Ciphertext > plaintext*. *Ciphertext* lebih panjang dibandingkan hill cipher dengan *pseudo invers*

Tabel 15. Perbandingan *Ciphertext* Yang Dihasilkan

Plain text	**Tesis NURI 2015**	
Cipher text	Hill cipher dengan *pseudo invers*	=~!4bSmxK[v{Aij(Q)UK)3vIO
	Hill cipher dengan *pseudo invers* dan operasi biner X-OR dan X-NOR	@&1|k%SDTu.$x&ha%v5;=UFHo'kiiN{}mH98J~!F'#yQ8UE']/3cSE}A`jkj]$zK5I+ >2IJ#K9@3(JX{+<|XYE{=?v<TK1pTA]:G>WKns{aV ZM&B<"hcWG6`p&G%7BPbSH`B,J-42vnhbQ;YC?>ur,SQ4FD:"jlaO=62.PzK

Plain text	**Pengujian sistem akan dilaksanakan hari senin, 17 Agustus 2015 Pukul 10.00!**	
Cipher Text	Hill cipher dengan *pseudo invers*	k7rR9?N_MO.Uxkmob++c0z,;QWN2V?'c,wuev;+Nvd<tXA8 YB|(Zvs(*Y]woU?Y?[EX^@&p~1>>3t= fp'2;N2WPnA={LW0@AU1K'$Bd/ap:IyFz\@T, m(O{&~n
	Hill cipher dengan *pseudo invers* dan operasi biner X-OR dan X-NOR	;/YK?PG}=\X2B&bnC'=@Qx*&aeB;)[T1A=etK%K)?`rK4mG] !%0[hHu]9FYZkUHW$r?Tt*E_,B95F[]BvaQQ@dO^>5{.T&6 *{bkicF]&.NJ$T[;/YK?W$u,"MeVHq:;Tb=w!tBep_5t<,?X!p{K #=aUO_T{_M&EsynkK/AO*\|aq1+!mo+.!]6`p&gjVHH;SH`B,_ + 5T+_ &xhkgT<!!uH eVG=5Q8MHd &xq#iz_=sO(zklh0_\|d6N^\|_xx(=6;%y!e[`sT?B.KJZ-E{Bwh?Rb<RK%N:p_5t<VL^H/mkhc}76JE,E)5;n!!uH G$ttZQ8MHd &xq#Zy))jCM*N4JRBN5#fWET&D[Hz]9FYZp4F=t@3(JXYN &!x<faiMz{h&fv~n2*\;a= r>NS>MTC&nVUE)w3cSE}l'Yw#X.UHeyI~&~{+<|XzbRK`RI `!w#@pnHpdVT\:y~AK]$zK5^P_HnRI`!wfUGN"XQ==4^/OS X=UFHo{<LkgqP-&/x&0"E)@wK'<"Ye*.$x&hX:Vk u.T)mmO)=:`%0E24)2t|zbRK`Sy#Y>u}mKp<,LqM1gYB?H#t 8ou.T)mJ*0zdoTC$7,YLHM4ZM&;ZO=c^RTDBzU\ht9=UFH o'kiiNm"e)ZnA.>8fSA!yd&wkzLN)Q.VA/&<~@u&vyUC\|`KO) Byq~mt}QWJN6c)2w6q[f$W;+,;LD$y=.BE]K8N=.E:UXKH3l <R&d~SCKi=UFHoGL#]f>"e&t4I`&Ix6H!nC[XMV4kfH"g-3q3KO)Byq~mt}rZPW%M<Mq<CB;=1teWN{IJ#K9S|jnj)@w K'mZON[-E{Bw='YqcnhbQ;

Dari Tabel 15 dapat dilihat bahwa *ciphertext* hasil enkripsi algoritma hill cipher dengan *pseudo invers* dan operasi biner jauh lebih panjang dibandingkan hasil enkripsi algoritma hill cipher terdahulu. Semakin besar matriks kunci yang digunakan, semakin panjang *ciphertext* yang dihasilkan. Jumlah karakter *ciphertext* yang dihasilkan adalah:

$$l_c = ((((2 * l_p)/m * n) * 2)/m) * n.$$

dengan l_p harus memenuhi syarat ((2 * l_p) mod m)=0. Jika tidak sama dengan nol (0), maka l_p harus ditambah sehingga memenuhi syarat ((2 * l_p) mod m) =0.

5. PENUTUP

Penelitian ini memodifikasi algoritma kriptografi hill cipher menggunakan operasi biner dan pembagian bit biner. Berdasarkan pembahasan yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Kriptografi hill cipher dapat dimodifikasi dengan menambahkan operasi biner dan pembagian bit biner.
2. Jumlah karakter *ciphertext* yang dihasilkan adalah: l_c =((((2 * l_p)/m * n) * 2)/m)*n. dengan lp harus memenuhi syarat ((2 * l_p) mod m) =0. Jika tidak sama dengan nol (0), maka l_p harus ditambah sehingga memenuhi syarat ((2 * l_p) mod m) =0. dan lebih rumit dibandingkan plaintextnya.
3. Algoritma yang dihasilkan dapat diterapkan atau disisipkan pada berbagai aplikasi sebagai pengamanan.

Penelitian ini masih memiliki banyak kekurangan. Diharapkan algoritma ini dapat dikembangkan lagi untuk mendapatkan ciphertext yang lebih kuat. Berikut ini saran-saran untuk perkembangan algoritma hill cipher

1. Sebaiknya gunakan modulus bilangan prima untuk menambah varian matriks kunci yang dapat digunakan. Karena modulus yang bukan bilangan prima menyebabkan banyak matriks yang tidak memiliki invers determinan sehingga tidak dapat digunakan sebagai matriks kunci.
2. Data yang diolah berbentuk huruf, angka dan karakter tertentu saja, diharapkan algoritma ini dapat dikembangkan dan diterapkan pada data selain karakter.

DAFTAR PUSTAKA

[1] Ariyus, Dony. (2008). Pengantar Ilmu Kriptografi Teori, Analisis dan Implementasi. Yogyakarta: Penerbit Andi

[2] Chandrasekhar, Suman.,dkk. (2013). *A Secure Encryption Technique Based on Advanced Hill Cipher For a Public Key Cryptosystem*. *IOSR Journal of Computer Engineering*, Vol 11, Issue 2(May. –Jun.2013), PP 10-14

[3] Munir, Rinaldi. (2007). Kriptografi. Bandung: Informatika

[4] Puspita, Khairani., M. Rhifky Wayahdi. (2015). Analisis Kombinasi Metode Caesar Cipher, Vernam Cipher dan Hill Cipher Dalam Proses Kriptografi. Seminar Nasional Teknologi Informasi dan Multimedia 2015.

[5] Rahman, M. Nordin A., dkk. (2013). *Cryptography: A New Approach of Classical Hill Cipher. International Journal of Security and Its Applications*, Vol. 7, No. 2

[6] Sastry, V. Umakanta., dkk. (2010). *A Modified Hill Cipher Involving Interweaving and Iteration. International Journal of Network Security*, Vol 11, No.1, PP.11-16

[7] Soulie, Juan. 2007. C++ Language Tutorial. Available at http://www.cplusplus.com/files/tutorial.pdf

[8] Supranto, J. (1997). Pengantar Matrix. Jakarta: Rineka Cipta

[9] Viswanath,M.K. (2015). *A Public Key Cryptosystem Using Hill's Cipher*. London:Taylor & Francis.

[10] Wowor, Alz Danny. (2013). Modifikasi Kriptografi Hill Cipher Menggunakan Convert Between Base. Seminar Nasional Sistem Informasi Indonesia.

[11] Wowor, Alz Danny. (2014). Penggunaan Determinan Polinomial Matriks Dalam Modifikasi Kriptografi Hill Cipher.