Enkripsi Modifikasi *Playfair* dengan *Vigenere Extended*

Benardi Atmadja - 13510078
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510078@std.stei.itb.ac.id

***Abstract*— Kriptografi, merupakan suatu cara untuk menjaga kerahasiaan suatu pesan. Kriptografi sendiri dibagi menjadi dua macam yaitu kriptografi klasik dan kriptografi modern. Kriptografik klasik telah ditemukan sejak dulu dan proses enkripsinya masih sangat sederhana dan biasanya dilakukan secara manual. Baru kemudian kriptografi modern ditemukan yang menggunakan perhitungan rumit dan biasanya proses enkripsi dilakukan dengan menggunakan komputer. Hingga sekarang kedua jenis kriptografi ini masih digunakan untuk menjaga kerahasiaan pesan.**

Algoritma kriptografi klasik yang cukup sulit untuk dipecahkan adalah *playfair* cipher. Algoritma ini mengupayakan agar persebaran frekuensi huruf-huruf menjadi rata sehingga sulit dipecahkan oleh kriptanalis. Akan tetapi walaupun sulit dipecahkan, algoritma ini masih memiliki beberapa kelemahan sehingga bisa dipecahkan oleh kriptanalis.

Dilain sisi terdapat algoritma *vigenere* cipher yang merupakan algoritma kriptografi klasik lainnya yang termasuk sulit dipecahkan. Kekurangan dari algoritma ini adalah rentannya terhadap analisa kasiski.

Dari deskripsi kekurangan dan kelebihan kedua algoritma diatas, makalah ini akan membahas tentang modifikasi algoritma *playfair* dan *vigenere* agar kerahasiaan pesan lebih aman dari serangan kriptanalis.

***Index Terms*—playfair cipher, vigenere chiper, kriptanalisis, kriptografi.**

I. PENDAHULUAN

Seiring dengan berkembangnya kemajuan teknologi, diperlukan suatu pengamanan terhadap persebaran pesan informasi yang ada. Kriptografi telah menjadi suatu ilmu dan seni yang amat berguna untuk mengamankan pesan yang dari pihak yang tidak bertanggung jawab. Ilmu kriptografi terus berkembang dari kriptografi klasik sejak jaman dahulu yang belum ada komputer hingga sekarang kriptografi modern yang berkaitan dengan bilangan yang amat besar dan algoritma yang amat rumit.

Perkembangan kriptografi terus berlanjut walaupun algoritma yang terkemuka dan dinilai kompleks sudah mulai bisa dipecahkan. Suatu cara mengamankan pesan boleh jadi tidak rahasia. Algoritma yang baik akan memiliki kekuatan pada keamanan kunci sehingga pihak yang tidak berwenang boleh jadi memiliki seluruh pesan chiper dan algoritmanya. Namun pihak tersebut tidak bisa mendekripsikan pesan karena kunci tidak diketahui.

Algoritma kriptografi klasik sudah dinilai *Obsolate* atau sudah tidak aman karena dapat dengan mudah dipecahkan menggunakan alat bantu komputer. Di antara algoritma yang sudah bisa dipecahkan dalam hitungan detik adalah vigenere chiper dan *playfair* chiper.

Di mana ada kriptografer, ada pula lawannya yaitu kriptanalis. Kriptanalis akan berusaha mengetahui cara kerja algoritma yang digunakan oleh kriptografer dan menyerang algoritma yang ada untuk mendekripsikan chiper teks yang ada.

Pada makalah ini akan diberikan sebuah modifikasi algoritma kriptografi klasik beserta analisis pemecahan kode serta serangan-serangan yang mungkin dilakukan kriptanalis untuk memecahkannya. Pada akhir bagian akan diberikan kesimpulan yang menyangkut kelebihan dan kekurangan algoritma tersebut.

II. DASAR TEORI

A. *Playfair* Chiper

Salah satu ciri khas algoritma kriptografi klasik adalah, sistem enkripsi dan dekripsinya dipandang lebih sederhana. Penyandiannya tidak harus memerlukan komputer dengan perhitungan yang rumit. Proses enkripsi dan dekripsinya dapat dilakukan hanya dengan perhitungan manual, bahkan hanya dengan menggunakan pena dan kertas. Selain itu, berbeda dengan algoritma modern yang basisnya adalah bit, algoritma klasik berbasiskan karakter. Terdapat dua macam algoritma klasik: sandi substitusi dan sandi transposisi.

Playfair, salah satu algoritma kriptografi klasik subsitusi, telah ditemukan sejak lama. Tepatnya, tahun 1854 oleh Charles Wheatstone. Nama dari algoritma ini diambil dari orang yang memopulerkan sandi, Lord Playfair, seorang teman dari Wheatstone. Sandi ini sempat digunakan pihak Inggris dalam Perang Boer II dan Perang Dunia I. Bahkan pihak Australia dan Jerman juga menggunaan sandi ini dalam Perang Dunia II. Namun, sandi ini tidak lagi digunakan oleh pihak militer. Sandi *playfair* dianggap tidak lagi aman untuk menjaga kerahasiaan sejak ditemukannya program yang mampu memecahkan sandi ini dalam hitungan detik.

P	L	A	Y	F
I	R	B	C	D
E	G	H	K	M
N	O	Q	S	T
U	V	W	X	Z

Gambar 1.1 Persegi 5x5 untuk dekripsi-enkripsi *playfair*

Proses pengenkripsian pesan yang menggunakan algoritma *playfair* adalah sebagai berikut:

Ditentukan kunci 5x5 yang disepakati oleh kedua belah pihak yang akan bertukan pesan. Misalnya PLAYFAIR. Huruf yang sama tidak ditulis ulang dan huruf J tidak disisipkan pada pesan melainkan dienkripsikan sebagai huruf I.

Pesan yang akan disandikan (Plainteks) akan dipecah per dua huruf. Misalkan pesan adalah “meet me at hammersmith bridge tonight”. Pesan tersebut akan menjadi demikian: ME XE TM EA TH AM XM ER SM IT HB RI DG ET ON IG HT

Dapat dilihat bahwa diantara huruf yang sama. EE pada meet akan sisipkan huruf X agar pesan tersebut nantinya akan bisa dienkripsikan dan tidak mudah diketahui oleh kriptanalis.

Setelah kalimat pesan diubah, pesan akan dituliskan ulang dengan aturan sebagai berikut:

- Bila huruf pertama dan huruf kedua berada pada kolom yang sama maka chiper teks yang akan dituliskan adalah huruf yang berada di samping kanan dari huruf plainteks. Pada ME menjadi “EG”

- Bila huruf pertama dan huruf kedua berada pada baris yang sama yaitu pada “TM”. Maka chiperteks yang ada akan dituliskan ke bawah dari huruf plain teks tersebut berdasarkan kotak 5x5 menjadi “ZT”.

- Bila huruf pertama dan huruf kedua tidak berada pada kolom atau baris yang sama maka huruf chiperteks akan diambil berdasarkan perpotongan kolom dan baris dari huruf pertama dengan kedua. Pada “XE” dapat dilihat bahwa perpotongan baris dan kolom antara X dan E adalah huruf U dan untuk E dan X adalah K. Maka pesan menjadi UK.

Dari gambar yang ada dapat dilihat bahwa terdapat 25 huruf yang terletak pada persegi dengan posisi yang bisa bertukar di mana saja. Fakta ini membuat sandi *playfair* memiliki sekitar 25! atau 1.551121x2025 kemungkinan kunci. Namun, ada beberapa isu dalam *playfair* yang dapat dieksploitasi kriptanalis. Antara lain:

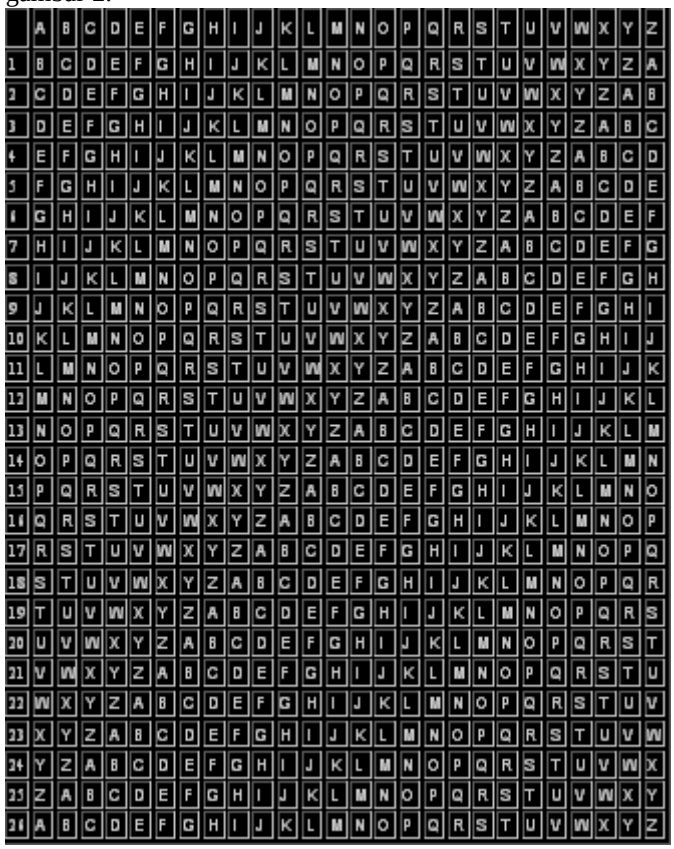
- Sebuah huruf tidak dapat mengenkripsi dirinya sendiri
- Sebuah huruf hanya dapat mengenkripsi dari satu sampai lima huruf saja
- Sebuah huruf bisa jadi dua kali lebih sering menjadi enkripsi dibanding yang lain
- Sebuah *plaintext* dan *ciphertext* saling menyandikan satu sama lain

B. *Vigenere* Chiper

Vigenere Chiper merupakan algoritma kriptografi klasik dimana huruf-huruf plainteks yang ada digeser berdasarkan kunci yang diberikan. Misalkan plainteks adalah SAYA SUKA SUSU SAPI dienkripsi dengan *vigenere* cipher berkuncikan “RAHASIA”. Karena panjang kunci tidak sepanjang plainteks, pada *vigenere* kunci tersebut akan diulang hingga panjangnya sejumlah plainteks. Proses pengenkripsian akan menghasilkan:

Plainteks: SAYA SUKA SUSU SAPI
Kunci: RAHA SIAR AHAS IARA
Chiperteks: JAFA KCKR SBSM AAGI

Proses pengenkripsian dapat juga diliat berdasarkan gambar 2.



Gambar 2. Tabel kunci enkripsi dekripsi *Vigenere*

(Sumber:http://www.simonsingh.net/The_Black_Chamber/v_square.html Waktu akses 25 Maret 17.45)

Tabel tersebut adalah penggeseran *vigenere* chiper. Di mana kunci A = 0, B=1, C=2 , … , Z= 25. Pada huruf pertama misalnya S dengan R(17), maka kita lihat hasil penggeseran akan menghasilkan huruf J.

Selain penggunaan *vigenere standard*, terdapat versi *extended* dimana *vigenere* yang dilakukan menggunakan karakter 255 ASCII sesuai gambar berikut.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	▒	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	Pts	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

Gambar 3. Tabel 255 ASCII

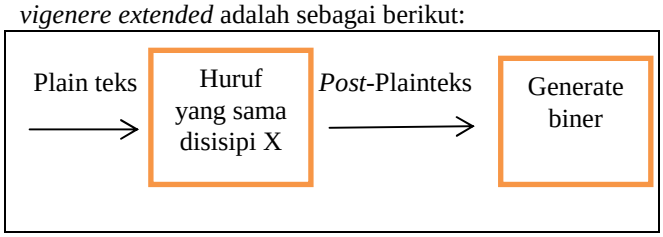
(Sumber:http://www.commfront.com/ascii-chart-table.htm Waktu akses 25 Maret 17.50)

Kode program yang menggunakan rumus dalam implementasi *Vigenere* Chiper *extended* ditunjukan pada kode dibawah ini:

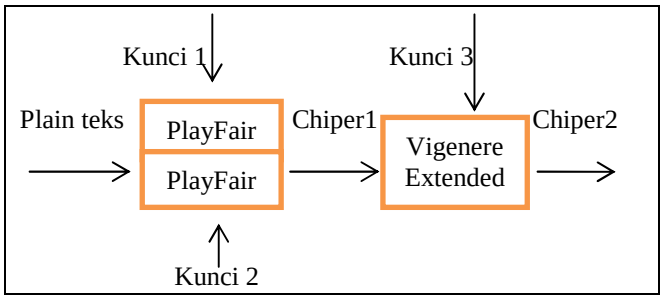
```
static String vig256(String teks, final String key, int tipe) {
        String kata = "";
        for (int i = 0, j = 0; i < teks.length(); i++) {
            char c = teks.charAt(i);

            if (tipe == 1) {
                kata += (char) ((c + key.charAt(j)) % 265);
            } else if (tipe == 2) {
                kata += (char) ((c - key.charAt(j) + 256) % 256);
            }
            j = ++j % key.length();
        }
        return kata;
    }
```

III. ALGORITMA ENKRIPSI

Proses pengenkripsian algoritma modifikasi *Playfair* dan *vigenere extended* adalah sebagai berikut:



Gambar 4. Tahap 1 enkripsi



Gambar 5. Tahap 2 enkripsi

Pada awalnya plain teks yang berisi pesan yang ingin dirahasiakan akan diberikan sisipan X bila terdapat huruf yang sama seperti penjelasan *Playfair* Chiper.

Plain teks akan diencrypt dengan dua buah *playfair*. Bila bilangan biner adalah 0 maka akan diencrypt dengan play fair 1. Bila bilangan biner adalah 1 maka akan diencrypt dengan *playfair* 2. Hasil chiperteks1 akan berupa gabungan enkripsi *playfair* 1 dan *playfair* 2. Kemudian hasil chiperteks akan divigenerekan.

Proses generate biner yang digunakan di sini adalah penjumlahan dua huruf. Bila genap maka akan dicatat angka 0. Sedangkan bila berjumlah ganjil akan dicatat angka 1.

Untuk implementasi lebih lanjut, disarankan menggunakan seed pada bilangan random yang disepakati kedua belah pihak agar proses pengenkripsian kedua buah algoritma menjadi sulit diketahui.

Kembali ke algoritma tersebut, Misalkan plain teks yang ingin dienkripsikan adalah:

> Malam ini bebaskanlah hatimu dari dendam terhadap orang yang melukaimu hari ini kemarin dan di masa lalu Sesungguhnya jika kau pikirkan dan ingat ingat dengan lebih teliti pasti ada orang orang yang malam ini juga sedang marah dan dendam karena kau telah menyalahi mereka baik sengaja atau tidak

Plain teks kalimat tersebut akan dipecah menjadi per 2 huruf. Huruf yang bersamaan seperti pada hh pada

bebaskanlah hatimu. Akan disisipi dengan huruf X.

MA LA MI NI BE BA SK AN LA HX HA TI MU DA RI DE ND AM TE RH AD AP OR AN GY AN GM EL UK AI MU HA RI XI NI KE MA RI ND AN DI MA SA LA LU SE SU NG XG UH NY AI XI KA KA UP IK IR KA ND AN IN GA TI NG AT DE NG AN LE BI HT EL IT IP AS TI AD AO RA NG OR AN GY AN GM AL AM IN IX IU GA SE DA NG MA RA HD AN DE ND AM KA RE NA KA UT EL AH ME NY AL AH IM ER EK AB AI KS EN GA IA XA TA UT ID AK

Post-plainteks

Kalimat yang sudah diolah tersebut akan diproses dengan menggunakan generate biner sesuai dengan aturan sebagai berikut. A= 1. B= 2. C=3. Z= 26. Bila penjumlahan kedua bigraph adalah genap dituliskan 0 sedangkan untuk ganjil dituliskan 1.
Hasil dari pemrosesan untuk MA (13 + 1) LA (12+1), .. AK(1 + 11) adalah sebagai berikut :

01101101101001010010111101010101000000001000110
01111100011000100011111100100001011110101000010
01101011000110111011111011010111100

Hasil generate biner

Dari generate bilangan biner yang kita dapatkan, maka plainteks yang sudah disisipi dengan huruf X tadi akan dienkripsikan dengan menggunakan *playfair* bergantung dari biner yang didapat.

Bila 0 akan di enkripsikan dengan menggunakan *playfair*1. Sedangkan bila 1 akan dienkripsikan dengna *playfair*2.

Misalkan kita mempunyai *playfair* 1 berkunci (PLAYFAIR):

P	L	A	Y	F
I	R	B	C	D
E	G	H	K	M
N	O	Q	S	T
U	V	W	X	Z

Gambar 6. Kunci *Playfair* 1

Dan *playfair* 2 berkunci (SEPATUKACA)

S	E	P	A	T
U	K	C	B	D
F	G	H	I	L
M	N	O	Q	R
V	W	X	Y	Z

Gambar 7. Kunci *Playfair* 2

Chiper teks yang akan dihasilkan adalah

HF IT QF UE KA IB XS EQ_IT KW IP ND EZ BT BR KT TI FH SP BG TB TA QM EQ_KL EQ_HE TG XE BQ_EZ IP BR UC UE MG HF BR TI EQ_IR HF QY IT FD NK NX WN WH CF QW BQ_UC HY HY CS GB RB HY TI EQ_EU HL ND WN TS KT WN EQ_GT CR MQ_TG DN EI YQ_ND TB LQ_QT WN QM EQ_KL EQ_HE TI FH EU CU FB HL NK BT WN HF QT MB EQ_KT TI FH HY NT QE HY DS TG PI EG QW TI PI FQ_TN GM BI BQ_SX KW HL QB YP ST DS RI YH

Chiperteks 1 modifikasi *playfair*

Sesudah proses tersebut, maka chiperteks masih akan dienkripsi lagi dengan menggunakan kunci ke 3 yaitu kunci untuk mengenkripsikan secara vigenere 256 karakter ASCII. Di sini dimasukkan kunci yaitu “Vigenere” sehingga chiperteks yang ada akan menjadi:

¯‡®Â…Ã«v¾¬…¹¡’®˜‰¿¸ ªÃ…Ÿ½‡°Å…»µv·«…³¿’§ª ‰©· °Æ…ª²‡«¶…Åµv«®…Â§’¹—
‰¸² ªÃ…¡µ‡ª¿…ºªv½®…Æª’§§‰¬¿ ®Â…˜»‡º±…Çªv¶ ®…¶«’§¨‰» ªÃ…Ÿ»‡-´…Ã¾v²»…´©’³¡‰µ½ ¼À…-± ‡¨´…Ã¼v«¸…Ã¨’-¯‰¯¾ ¨Å… «‡·°…º¾v½°…³¶’ª«‰¯± ³¶…-·‡¹Á…½¹vÀµ…³¶’¬ª‰ª· ²Ã…ª°‡©¼…·®vÂ¸…¼ ©’¹˜‰³¶ ¶Æ…-·‡¶»…·¶v´³…³¶’-›‰»® «º…›¾‡¨Ã…¸§ v±³…¼”§ª‰¾³ -¸…§½‡²°…·¶v´»…Â®’« ‰¯¾ ³Æ…§ ®‡-Ç…¶¸v½®…¾®’ª ‰¸¼ ¹»…¡²‡«¿…Æ³v°´…°®’§§‰ º½ °É… µ‡¶°…Ëµv¼»…²¸’·Ÿ‰À-

Chiperteks 2 vigenere

Proses Dekripsi:

Untuk mendekripsikan chiperteks diatas diperlukan kunci vigenere extended yang akan merubah bentuk chiperteks menjadi huruf-huruf chiper *playfair*. Seperti pada chiperteks 1.

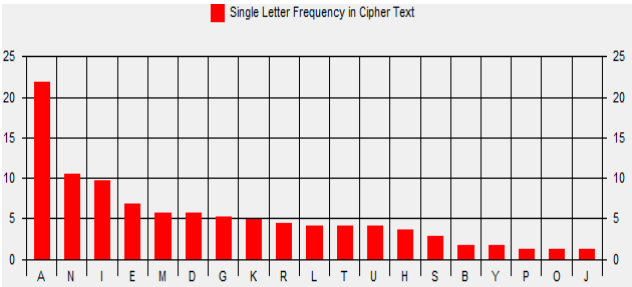
Kemudian dengan mempunyai kunci kedua buah

playfair dan bilangan biner yang ada, penerima pesan akan dapat mendekripsikan pesan yang ada.

IV. ANALISIS

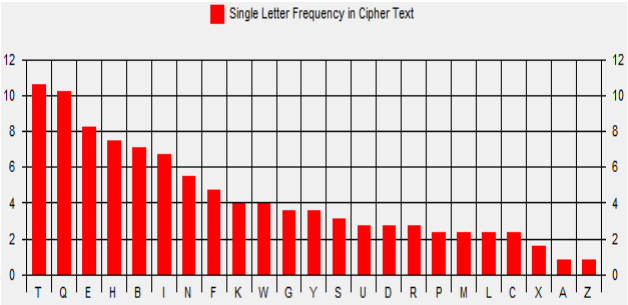
A. Persebaran Frekuensi kata *Playfair*

Dari plainteks persebaran karakter adalah sebagai berikut:



Gambar 8. Statistik Persebaran Karakter Plainteks
(Program: de.crypt.co.uk oleh Robert Marston)

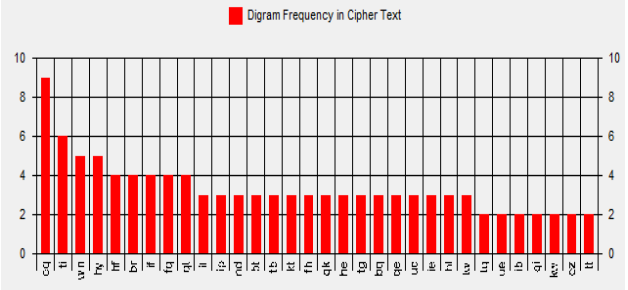
Sesudah dilakukan enkripsi hingga chiperteks 1. Persebaran karakter adalah sebagai berikut:



Gambar 9. Statistik Persebaran Karakter Chiperteks1
(Program: de.crypt.co.uk oleh Robert Marston)

Terlihat terjadi perubahan frekuensi karakter dimana huruf A yang dominan menjadi merata ke semua karakter kecuali J. (Sifat dari Playfair).

Sedangkan dari frekuensi bigraph dapat dilihat bahwa statistik dari gambar tersebut juga merata. Hal ini bisa juga disebabkan oleh panjang plainteks yang tidak terlalu besar sehingga jumlahnya juga tidak terlihat terlalu signifikan.



Gambar 10. Statistik Persebaran Karakter Plainteks
(Program: de.crypt.co.uk oleh Robert Marston)

B. Penggunaan Vigenere

Karena karakter ASCII yang digunakan extended, program tidak bisa melakukan analisis karakter. Metode kriptanalis untuk memecahkan vigenere chiper adalah menggunakan perhitungan panjang kunci dan menghitung perulangan huruf berdasarkan jarak dari kunci atau biasa dikenal dengan metode Kasiski.

Karakter chiperteks2 berupa huruf kapital dan berentang antara A sampai dengan Z. Dalam ASCII berentang antara 65-90. Bila *vigenere* dilakukan terdiri dari huruf kecil (97-122) dan huruf besar (65-90), kriptanalisis bisa mengetahui apakah kunci terdiri dari huruf besar atau huruf kecil berdasarkan cipherteks.

Pada algoritma ini menggunkan 4 buah kunci. 2 buah playfair, 1 *vigenere*, dan 1 deretan biner yang sulit diekstraksi dari chiperteks sehingga harus diberikan dengan sarana lain. Akan lebih baik bila generate biner dilakukan dengan seed sehingga cukup seed saja menjadi kunci ke 4 untuk algoritma ini.

Terlepas dari hal tersebut, algoritma ini akan menjadi kuat karena penggunaan 2 buah *playfair* akan memungkinkan chiperteks yang ada memiliki bigraph yang sama dan berlainan arti.

V. KESIMPULAN

Algoritma ini menjadi suatu cara untuk mengamankan pesan dari serangan frekuensi kata.yang baik. Kekurangan *vigenere* chiper dapat diperbaiki dengan kelebihan algoritma *playfair*. Penggunaan biner untuk membagi apakah plainteks dienkripsi dengan algoritma tertentu dapat menjadi suatu cara yang bisa digunakan pada algoritma lain.

VI. REFERENCES

[1] Munir, Rinaldi. Ir. M.T. *Diktat Kuliah IF5054 Kriptografi*

[2] http://www.simonsingh.net/The_Black_Chamber/playfair_cipher.html waktu akses 25 Maret 2013 17.05

[3] http://www.simonsingh.net/The_Black_Chamber/vigenere_cipher.html waktu akses 25 Maret 2013 17.10

[4] Department of The Army . (1990). “Field Manual”, Washington DC. (chapter 7)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 25 Maret 2013

Benardi Atmadja 13510078