ENKRIPSI DAN DEKRIPSI MENGGUNAKAN VIGENERE CIPHER ASCII JAVA

Mahmud Hidayatulloh , Entik Insannudin

Teknik Informatika UIN Bandung
email : mahmudhidayatulloh@student.uinsgd.ac.id; insan@if.uinsgd.ac.id

Abstraksi
Kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan yang meliputi aspek keamanan pesan seperti kerahasiaan, integritas data, serta otentikasi. Salah satu metode yang dapat digunakan untuk meyandikan pesan adalah Vigenere cipher. Pada jurnal ini, kami membahas kode program dan algortima Vigenere cipher berdasarkan tabel ASCII menggunakan bahasa pemrograman JAVA.

***Kata Kunci* :**
Kritografi, Vigenere, JAVA, Integritas

Pendahuluan

Kriptografi berasal dari bahasa Yunani, crypto dan graphia. Crypto berarti secret (rahasia) dan graphia berarti writing (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain.

Untuk berbagai alasan, keamanan dan kerahasiaan sangat kita butuhkan dalam komunikasi data. Ada berbagai cara untuk menjamin keamanan dan kerahasiaan komunikasi data kita di antaranya adalah dengan seni pengacakan data atau disebut juga Kriptografi [1]. Pada jurnal ini akan membahas teknik enkripsi dan dekripsi pada algoritma vigenere cipher berdasarkan tabel ACII menggunakan bahasa pemrograman java.

Algoritma vigenere cipher termasuk kriptografi simetri klasik. Dan Termasuk ke dalam cipher abjad-majemuk *(polyalpabetic substitution cipher)*. Pertamakali dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis, Blaise de Vigènere pada abad 16 (tahun 1586). Tetapi sebenarnya Giovan Batista Belaso telah menggambarkannya pertama kali pada tahun 1553 seperti ditulis di dalam bukunya *La Cifra del Sig. Giovan Batista Belaso*. [2]

Algoritma tersebut baru dikenal luas 200 tahun kemudian yang oleh penemunya *cipher* tersebut kemudian dinamakan vigenere cipher. Cipher ini berhasil dipecahkan oleh Babbage dan Kasiski pada pertengahan Abad 19. Vigenere cipher digunakan oleh Tentara Konfiderasi (*Confederate Army*) pada Perang Sipil Amerika (*American Civil war*). [2]

Vigenere cipher menggunakan bujursangkar vigenere *(Vigenere Square)* untuk melakukan enkripsi. Setiap baris di dalam bujursangkar menyatakan huruf-huruf cipherteks yang diperoleh dengan Caesar cipher. Namun pada jurnal ini kami menggunakan tabel ACII, dimana *key*-nya sebanyak 256 karakter. Sehingga hasil enkripsinya relatif lebih aman dibanding dengan vigenere alfabet biasa (26 karakter).

Ciri-ciri kriptografi kunci simetri klasik:

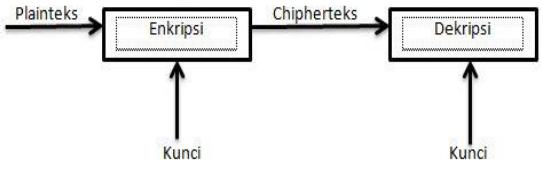
- Kunci enkripsi = kunci dekripsi
- Algoritma kriptografinya disebut algoritma simetri
- Contoh algoritma: caesar cipher, vigenere cipher, playfair cipher, DES, 2DES, 3DES, AES, RC2, RC3, RC4, RC5, RC6, Blowfish, GHOST, LOKI, IDEA, dll. [1]

Tinjauan Pustaka

Setelah melakukan pencarian di internet, kami menemukan beberapa algoritma vigenere cipher namun hanya berdaskan alfabet. Ada yang berdasarkan ACII, namun menggunakan bahasa pemrograman lain (C++). Jika ada yang menggunakan bahasa pemrogrman java, tapi belum GUI (*Graphical user interface*). Dalam jurnal ini, kami akan mengimplementasikan algoritma vigenre cipher ACII menggunakan bahasa pemrograman java GUI.

Plaintext merupakan pesan yang belum disandikan, sedangkan ciphertext merupakan pesan yang sudah disandikan.

Enkripsi adalah proses menyandikan plaintext menjadi ciphertext, sedangkan dekripsi adalah proses mengembalikan ciphertxt menjadi plaintext atau membuka sandi.



Gambar 3. *Proses enkripsi dan dekripsi simetri klasik*

Berikut rumus enkripsi dan dekripsi vigenere cipher:

a. Enkripsi

$C_{i =} (P_i + K) \bmod 256$

b. Dekripsi

$P_i = (C_i - K) \bmod 256$

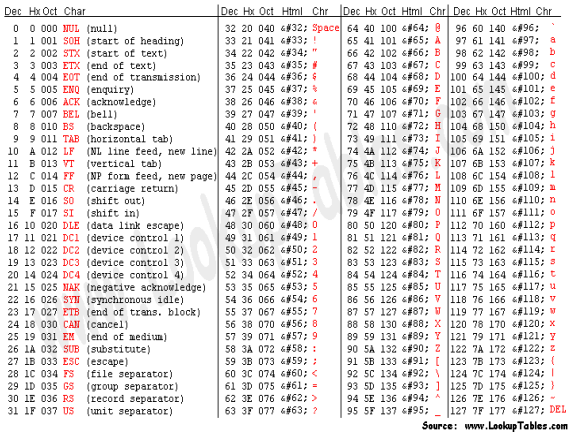
Keterangan:

C_i = nilai desimal karakter ciphertext ke-i

P_i = nilai desimal karakter plaintext ke-i

K = nilai desimal karakter kunci ke-i

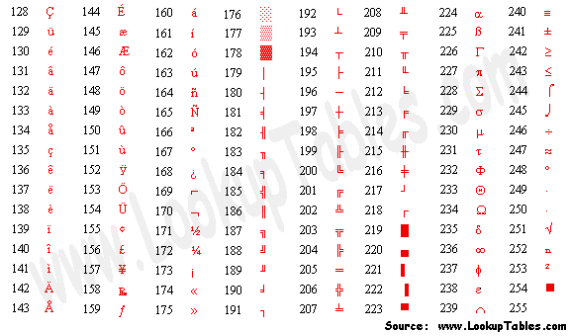
mod 256 = karena berdasarkan ASCII



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Gambar 1. *Tabel ACII (teks)*



128	Ç	144	É	160	á	176	░	192	└	208	╨	224	α	240	≡
129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	2
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	

Source: www.LookupTables.com

Gambar 2. *Tabel ACII (simbol)*

Misal Pada Vigenere Cipher kunci K adalah urutan huruf-huruf K = k……kd dimana ki didapat dari banyak penggeseran pada karakter ACII ke-i. Berikut adalah formula vigenere cipher:

Misalnya m menentukan beberapa nilai integer positif diberikan P = C = K = $(Z_{97})^m$. untuk sebuah kunci K=(k1,k2,……,km), didefinisikan :

Enkripsi:

ek (p1, p2 ... pm) = (p1 + k1, p2 + k2 ... pm + km) (mod 26)

Dekripsi:

dk (c1, c2 ... cm) = (c1-k1, c2-k2 ... cm-km) (mod 26)

Contoh:

Plaintext: HANTAM MEREKA
KEY: BOMBOMBOMBOMB
Ciphertext:)/:5/9 ;1337"

Metode Penelitian

Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital.

Pada Vigenere Cipher kunci K adalah urutan huruf-huruf K = k……kd dimana ki didapat dari banyak penggeseran pada alphabet ke-i. Berikut adalah formula vigenere cipher :

Misalnya m menentukan beberapa nilai integer positif diberikan P = C = K = (Z_{97})m. untuk sebuah kunci K=(k1,k2,……,km), didefinisikan :

eK(x1,x2,…….,xm)=(x1+k1, x2+k2,……., xm+ km)

dan

dK(y1,y2,…….,ym)=(y1-k1,y2-k2,……., ym- km)

dimana semua operasi adalah berbasiskan pada Z_{97}

Langkah Enkripsi:

Ubah kunci dan plaintext kedalam urutan bilangan integer dengan memperhatikan table konversi.

Tambahkan nilai K dan plaintext dengan mereduksikan sebagai penjumlahan modulo 97. Dan apabila ukuran plaintext lebih panjang dari pada kunci maka penjumlahan dengan K dilakukan secara periodic dalam arti bila K sudah mencapai nilai terakhir maka akan diulang kembali pada K untuk nilai dengan urutan pertama.

Konversi kembali urutan bilangan hasil penjumlahan K dan plaintext kedalam karakter dengan mengacu kembali pada tabel ACII (lihat gambar 1 dan 2).

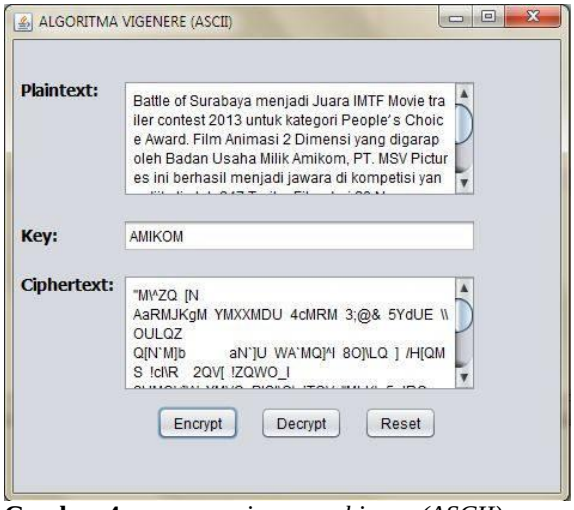
Langkah Dekripsi:

Ubah ciphertext dan kunci kedalam urutan bilangan integer dengan memperhatikan table konversi.

Pada masing-masing urutan bilangan yang merupakan ciphertext kurangkan dengan nilai K dan mereduksikan sebagai penjumlahan modulo 97. Konversikan kembali urutan bilangan kedalam karakter dengan kembali mengacu pada tabel ACII (lihat gambar 1 dan 2).

Hasil dan Pembahasan

Kami menggunakan bahasa pemograman java. Pada input Plaintext program dapat melakukan enkripsi serta dekripsi hasil menggunakan algoritma Vigenere Chiper. Berikut adalah source serta *screenshot* demo program yang telah kami buat: :



Gambar 4. *program vigenere chipper (ASCII)*

Berikut *source code* algoritma vigenere cipher ASCII:

```
/**================================**/
public class VigenereClass {

// kelas enkripsi

   public String enkripsi(String keyword, String line)
{
      String result = "";
      int offset;
      int j = 0, shift;
      for (int i = 0; i < line.length(); i++) {
         shift = ((int) keyword.charAt(j)) - 97;
         j++;
         j %= keyword.length();
         offset = ((int) line.charAt(i) + shift) % 256;
         result += (char) (offset);
      }
      return result;
   }

//kelas dekripsi
   public String dekripsi(String keyword, String line)
{
      String result = "";
      int offset;
      int j = 0, shift;
      for (int i = 0; i < line.length(); i++) {
         shift = ((int) keyword.charAt(j)) - 97;
         j++;
         j %= keyword.length();
         offset = ((int) line.charAt(i) - shift) % 256;
         if (offset < 0) {
            offset += 256;
         }
         result += (char) (offset);
      }
      return result;
   }

}
```

Penjelasan *source code*:
NB: lihat nilai desimal karakter di tabel ASCII.
Diketahui plaintext "HANTAM MEREKA", jika menggunakan nilai Z = 97 (berdasrkan *source code*), kalau di tabel ASCII yaitu huruf "a". dan kuncinya "BOM". Maka proses enkripsinya, sebagai berikut:

Rumus enkripsi: Ci = (Pi + K) mod 256

- H = 72
  - shift = nilai desimal kunci (B) – 97
    = 66 - 97 = -31
  - Ci = (72 + shift) mod 256
    = (72 - 31) mod 256
    = 41
  - nilai desimal 41 di tabel ASCII mempunyai karakter)
- A = 65
  - shift = nilai desimal kunci (O) – 97
    = 79 - 97 = -18
  - Ci = (65 + shift) mod 256
    = (65 - 18) mod 256
    = 47
  - nilai desimal 47 di tabel ASCII mempunyai karakter /
- N = 78
  - shift = nilai desimal kunci (M) – 97
    = 77 - 97 = -20
  - Ci = (78 + shift) mod 256
    = (78 - 20) mod 256
    = 58
  - nilai desimal 58 di tabel ASCII mempunyai karakter :
- T = 84

- shift = nilai desimal kunci (B) – 97
  = 66 - 97 = -31
- Ci = (84 + shift) mod 256
  = (84 - 31) mod 256
  = 53
- nilai desimal 53 di tabel ASCII mempunyai karakter 5

- A = 65
  - shift = nilai desimal kunci (O) – 97
    = 79 - 97 = -18
  - Ci = (65 + shift) mod 256
    = (65 - 18) mod 256
    = 47
  - nilai desimal 47 di tabel ASCII mempunyai karakter /
- M = 77
  - shift = nilai desimal kunci (M) – 97
    = 77 - 97 = -20
  - Ci = (77 + shift) mod 256
    = (77 - 20) mod 256
    = 57
  - nilai desimal 57 di tabel ASCII mempunyai karakter 9

- Spasi = 32
  - shift = nilai desimal kunci (B) – 97
    = 66 - 97 = -31
  - Ci = (32 + shift) mod 256
    = (32 - 31) mod 256
    = 1
  - nilai desimal 1 di tabel ASCII mempunyai karakter SOH

- M = 77
  - shift = nilai desimal kunci (O) – 97
    = 79 - 97 = -18
  - Ci = (77 + shift) mod 256
    = (77 - 18) mod 256
    = 59
  - nilai desimal 59 di tabel ASCII mempunyai karakter ;
- E = 69
  - shift = nilai desimal kunci (M) – 97
    = 77 - 97 = -20
  - Ci = (69 + shift) mod 256
    = (69 - 20) mod 256
    = 49
  - nilai desimal 49 di tabel ASCII mempunyai karakter 1
- R = 82
  - shift = nilai desimal kunci (B) – 97
    = 66 - 97 = -31
  - Ci = (82 + shift) mod 256
    = (82 - 31) mod 256
    = 51
  - nilai desimal 51 di tabel ASCII mempunyai karakter 3

- E = 69
  - shift = nilai desimal kunci (O) – 97
    = 79 - 97 = -18
  - Ci = (69 + shift) mod 256
    = (69 - 18) mod 256
    = 51
  - nilai desimal 51 di tabel ASCII mempunyai karakter 3
- K = 75
  - shift = nilai desimal kunci (M) – 97
    = 77 - 97 = -20
  - Ci = (75 + shift) mod 256
    = (75 - 20) mod 256
    = 55
  - nilai desimal 49 di tabel ASCII mempunyai karakter 7
- A = 65
  - shift = nilai desimal kunci (B) – 97
    = 66 - 97 = -31
  - Ci = (65 + shift) mod 256
    = (65 - 31) mod 256
    = 34
  - nilai desimal 34 di tabel ASCII mempunyai karakter "

Jadi, Plaintext "HANTAM MEREKA" dengan kunci "BOM" mempunyai ciphertext)/:5/9 ;1337" . Untuk melakukan proses dekripsi, bisa menggunkan rumus dekripis vigenere cipher.

Kesimpulan dan Saran

Implementasi enkripsi-dekripsi dengan algoritma vigenere cipher pada kode ASCII memberikan kemungkinan yang luas dan lebih banyak karakter yang tercakup, tidak hanya terbatas pada 26 alfabet, tetapi juga mencakup karakter-karakter dan simbol seperti . ,", ',= dan sebagainya.

Vigenere cipher yang masih tergolong kriptografi simetri klasik sudah di pecahkan oleh beberapa peneliti dan dapat di simpulkan bahwa Algoritma Vigenere cipher bisa dikatakan sudah *jadul* dan tidak aman lagi jika digunakan, sehingga hampir tidak ada yang menggunakan tersebut.

Saran kami, jika ingin lebih aman maka gunakan *Hybrid Cryptosystem*, yaitu gabungan dua algoritma atau lebih (asimetri dan simetri).

Daftar Pustaka

[1] Hidayat, Akik, 2009, *KRIPTOGRAFI DAN STENOGRAFI MENGGUNAKAN ALGORITMA VIGENERE DAN TEA (TINY ENKRIPSIION ALGORITHM)*, Repositoy UNPAD, Indonesia.

[2] Hartatik, 2005, Materi Kriptografi, STMIK AMIKOM, Indonesia.

[3] Unknow, 10 Desember 2013, *Implementing Vigenere Cipher Poly-alphabetic Substitution In Java*, http://pjcodingcenter.blogspot.com/2013/03/implementing-vigenere-cipher-poly.html

[4] Kester, Quist-Aphetsi., 2013, *A HYBRID CRYPTOSYSTEM BASED ON VIGENÈRE CIPHER AND COLUMNAR TRANSPOSITION CIPHER*, Proceeding IJATER. Accra North, Ghana

[5] Unknow, 12 Januari 2014, *ASCII Table and Description*, http://www.asciitable.com

[6] Khannedy, Eko Kurniawan, 2012. "Pemrograman GUI menggunakan Java dan NetBeans