SUATU ALGORITMA KRIPTOGRAFI *STREAM CIPHER* BERDASARKAN FUNGSI *CHAOS*

Dwi Lestari
Jurusan Pendidikan Matematika FMIPA
Universitas Negeri Yogyakarta
E-mail: dwilestari@uny.ac.id

Muhamad Zaki Riyanto
Pendidikan Matematika JPMIPA FKIP
Universitas Ahmad Dahlan, Yogyakarta
E-mail: zakimath@gmail.com

Abstrak

Salah satu sifat fungsi *chaos* adalah sensitif terhadap nilai awal, artinya perubahan kecil pada nilai awal akan mengakibatkan perubahan besar pada nilai fungsi selanjutnya. Sifat seperti ini dapat diterapkan dalam kriptografi, terutama dalam algoritma kriptografi *stream cipher*. Pada *stream cipher* digunakan sebuah kunci yang disebut dengan kunci awal, dari kunci ini dibuat kunci yang lebih panjang dari kunci awal yang disebut dengan *key stream*. Penerapan fungsi *chaos* dalam *stream cipher* tentu menguntungkan, karena sifat sensitif pada nilai awal tersebut, sehingga diharapkan dapat meningkatkan keamanan dari *stream cipher* yang didasarkan pada fungsi *chaos*.

Kata kunci: enkripsi simetris, fungsi chaos, pembangkitan kunci rahasia

1. Pendahuluan

Perkembangan teknologi informasi dewasa ini telah berpengaruh pada hampir semua aspek kehidupan manusia, tak terkecuali dalam hal berkomunikasi. Dengan adanya internet, komunikasi jarak jauh dapat dilakukan dengan cepat dan murah. Namun di sisi lain, ternyata internet tidak terlalu aman karena merupakan jalur komunikasi umum yang dapat digunakan oleh siapapun sehingga sangat rawan terhadap penyadapan. Oleh karena itu, keamanan informasi menjadi faktor utama yang harus dipenuhi.

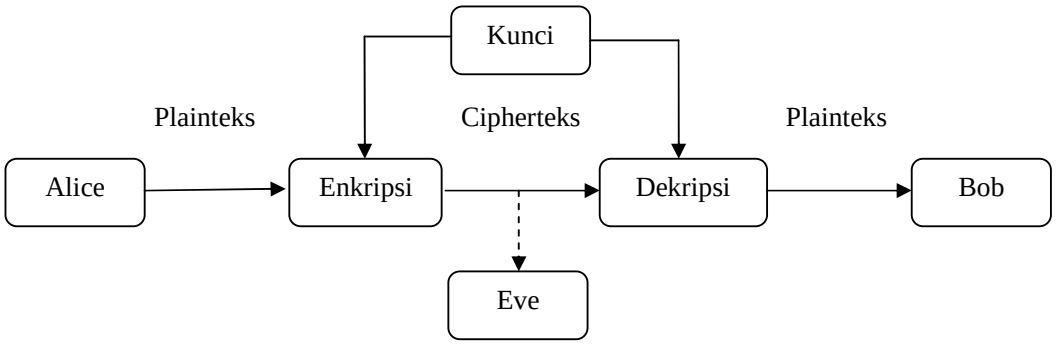
Salah satu solusinya adalah dengan menggunakan kriptografi. Kriptografi adalah suatu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data (Menezes dkk, 1996). Tetapi tidak semua aspek keamanan informasi dapat

diselesaikan dengan kriptografi. Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi pesan tersebut mungkin dapat disadap oleh pihak lain yang tidak berhak untuk mengetahui isi pesan tersebut. Untuk menjaga pesan, maka pesan tersebut dapat diubah menjadi suatu kode yang tidak dapat dimengerti oleh pihak lain. Enkripsi adalah suatu proses penyandian yang melakukan perubahan suatu pesan, dari yang dapat dimengerti, disebut dengan plainteks, menjadi suatu kode yang sulit dimengerti, disebut dengan cipherteks. Sedangkan proses kebalikannya untuk mengubah cipherteks menjadi plainteks disebut dekripsi. Proses enkripsi dan dekripsi memerlukan suatu mekanisme dan kunci tertentu.

Ada berbagai mekanisme enkripsi-dekripsi yang telah dikenal luas dalam kriptografi, seperti mekanisme XOR, Substitution-Permutation Network, Feistel Network, dan sebagainya. Operasi yang terlibat di dalamnya dapat berupa penjumlahan dan perkalian matriks, penjumlahan vektor, perkalian skalar, dan sebagainya. Pada makalah ini dibahas mengenai penerapan fungsi chaos yang digunakan untuk mendapatkan kunci untuk enkripsi-dekripsi. Devaney (1992) telah menjelaskan mengenai fungsi chaos, termasuk sifat-sifat dan cara untuk membuktikan suatu fungsi itu bersifat chaos atau tidak. Salah satu sifatnya adalah sensistif terhadap nilai awal. Hal ini dapat dimanfaatkan untuk kepentingan kriptografi dalam pembuatan kunci rahasia karena sifatnya tersebut. Metode enkripsi yang digunakan pada makalah ini adalah algoritma *stream cipher* yang akan diberikan pada bab selanjutnya.

2. Algoritma Kriptografi Stream Cipher

Algoritma kriptografi (sistem kriptografi) atau sering disebut dengan cipher merupakan suatu sistem atau kumpulan aturan-aturan yang digunakan untuk melakukan enkripsi dan dekripsi. Algoritma kriptografi simetris adalah sistem kriptografi yang menggunakan kunci enkripsi dan dekripsi yang sama. Sistem ini mengharuskan dua pihak yang berkomunikasi menyepakati suatu kunci rahasia yang sama sebelum keduanya saling berkomunikasi. Keamanan dari sistem ini tergantung pada kunci, membocorkan kunci berarti bahwa orang lain yang berhasil mendapatkan kunci dapat mendekripsi cipherteks. Algoritma kriptografi ini sering disebut dengan algoritma kriptografi kunci rahasia, seperti dijelaskan pada gambar berikut ini.



Gambar 1. Algoritma Kriptografi Simetris

Ada dua jenis algoritma kriptografi simetris yang saat ini, yaitu *block cipher* dan *stream cipher*. Pada algoritma kriptografi *block cipher*, metode enkripsi-dekripsi dilakukan dengan memotong plainteks menjadi blok-blok, dan masing-masing blok dilakukan enkripsi menggunakan kunci yang sama. Contoh *block cipher* yang dikenal luas saat ini adalah DES dan AES. Selanjuntya, pada algoritma kriptografi stream cipher, plainteks tidak dipotong menjadi blok-blok, akan tetapi enkripsi dilakukan secara mengalir menggunakan enkripsi dengan kunci yang mengalir juga. Algoritma kriptografi *stream cipher* sering juga disebut dengan sandi aliran. *Stream cipher* saat ini digunakan secara luas di internet dan di telepon seluler. Hal ini terjadi karena proses enkripsi-dekripsi yang relatif lebih cepat daripada *block cipher*. Salah satu keuntungan dari *stream cipher* adalah tidak dibatasi oleh panjang plainteks, sehingga *stream cipher* cocok untuk digunakan pada enkripsi suatu komunikasi yang berlangsung secara berkelanjutan, seperti komunikasi melalui telepon. Salah satu contoh kriptografi klasik yang menggunakan algoritma *stream cipher* adalah Vigenere *cipher* atas grup $Z_{26} = \{0,1,2,...,25\}$ yang dijelaskan sebagai berikut.

Diberikan plainteks $x_1, x_2, x_3, ...$ dengan $x_i \in Z_{26}$ dan kunci $k_1, k_2, ..., k_n$ dengan $k_i \in Z_{26}$. Cipherteks $y_1, y_2, y_3, ...$ diperoleh dengan proses enkripsi sebagai berikut

$$y_1 = x_1 + k_1, y_2 = x_2 + k_2, ..., y_n = x_n + k_n, y_{n+1} = x_{n+1} + k_1, y_{n+2} = x_{n+2} + k_2, ... \pmod{26}$$

Dapat dilihat bahwa proses enkripsi berlangsung secara mengalir. Diberikan korespondensi huruf abjad dengan bilangan sebagai berikut, a dengan 0, b dengan 1, c dengan 2, dan seterusnya sampai z dengan 25 seperti diberikan pada tabel di bawah ini.

Tabel 1. Korespondensi karakter dengan bilangan

0 ↔ a	1 ↔ b	2 ↔ c	3 ↔ d	4 ↔ e
5 ↔ f	6 ↔ g	7 ↔ h	8 ↔ i	9 ↔ j
10 ↔ k	11 ↔ l	12 ↔ m	13 ↔ n	14 ↔ o
15 ↔ p	16 ↔ q	17 ↔ r	18 ↔ s	19 ↔ t
20 ↔ u	21 ↔ v	22 ↔ w	23 ↔ x	24 ↔ y
25 ↔ z				

Sebagai contoh, misalkan dipunyai pesan “pesan rahasia” dan kunci rahasia “kunci”. Proses enkripsi diberikan dalam tabel di bawah ini.

Tabel 2. Proses Enkripsi Vigenere Cipher

Huruf	x_i	Kunci	k_i	$y_i = x_i + k_i \pmod{26}$	Huruf
p	15	k	10	25	z
e	4	u	20	24	y
s	18	n	13	5	f
a	0	c	2	2	c
n	13	i	8	21	v
r	17	k	10	1	b
a	0	u	20	20	u
h	7	n	13	20	u
a	0	c	2	2	c
s	18	i	8	0	a
i	8	k	10	18	s
a	0	u	20	20	u

Dari perhitungan pada tabel di atas, diperoleh cipherteks “**zyfcvbuucasu**”. Bentuk kunci yang mengalir ini disebut dengan kunci aliran (*key stream*). Untuk proses dekripsi dilakukan hal yang hampir sama dengan enkripsi, yaitu dengan menjumlahkan invers dari kuncinya. Vigenere *cipher* merupakan salah satu contoh dari stream cipher. Pada saat ini, contoh stream cipher yang digunakan adalah RC4, A5/1 dan A5/2 yang digunakan pada perangkat telepon seluler.

3. Algoritma Stream Cipher Menggunakan Fungsi Chaos

Fungsi chaos dalam matematika merupakan suatu fungsi yang mempunyai sifat bahwa nilai fungsinya sensitif terhadap nilai awal, artinya perubahan kecil pada nilai awal akan mengakibatkan perubahan besar pada nilai fungsinya. Penerapan fungsi *chaos* dalam *stream cipher* tentu menguntungkan, karena sifat sensitif pada nilai awal tersebut, sehingga diharapkan dapat meningkatkan keamanan dari *stream cipher* yang didasarkan pada fungsi *chaos*. Dalam makalah ini hanya dibahas mengenai penerapan dari fungsi chaos, sedangkan pembahasan mengenai fungsi chaos dapat ditemukan pada Gulick (1992).

Salah satu fungsi chaos adalah $f(x) = rx(1-x)$. Fungsi ini dikenal sebagai fungsi logistik yang dapat menjelaskan pertumbuhan populasi suatu spesies (Devaney, 1992). Fungsi ini nantinya digunakan secara iteratif, yaitu $x_{i+1} = rx_i(1-x_i)$. Nilai *r* merupakan suatu konstanta yang dapat ditentukan berdasarkan kunci, sedangkan nilai x_i juga ditentukan oleh kunci, sehingga nantinya dapat diperoleh suatu *key stream*. Misalkan dipilih $r = 3$ dan nilai awal $k_0 = 7$. Jika digunakan modulo 100.000, dengan rumus $k_{i+1} = rk_i(1-k_i)$ mod 100.000, maka diperoleh key stream:

k_1 = 99874

k_2 = 23066

k_3 = 48130

k_4 = 88282

k_5 = 66754

k_6 = 12602

k_7 = 6594

k_8 = 77274

k_9 = 87778

k_{10} = 67962

dan seterusnya...

Bilangan-bilangan itulah yang akan digunakan sebagai kunci untuk melakukan enkripsi. Jika hasil perhitungan di atas dituliskan secara berurutan, maka diperoleh *key stream* sebagai

berikut: 998742306648130882826675412602659477274877786796 2... Misalkan dipunyai suatu pesan "**pesan rahasia untuk bob**", menggunakan *key stream* tersebut dapat dilakukan proses enkripsi dengan membuat blok-blok *key stream* dengan panjang dua seperti pada tabel di bawah ini:

Tabel 3. Enkripsi Menggunakan Fungsi Chaos (Fungsi Logistik)

Huruf	x_i	k_i	$y_i = x_i + k_i \pmod{26}$	Huruf
p	15	99	10	k
e	4	87	13	n
s	18	42	8	i
a	0	30	4	e
n	13	66	1	b
r	17	48	13	n
a	0	13	13	n
h	7	08	15	p
a	0	82	4	e
s	18	82	22	w
i	8	66	22	w
a	0	75	23	x
u	20	41	9	j
n	13	26	13	n
t	19	02	21	v
u	20	65	7	h
k	10	94	0	a
b	1	77	0	a
o	14	27	15	p
b	1	48	23	x

Berdasarkan tabel di atas, diperoleh cipherteks "**kniebnnpewwxjnvhaapx**". Apabila pihak penerima akan mendekripsi cipherteks tersebut, maka pihak penerima harus mengetahui $r = 3$ dan nilai awal $k_0 = 7$, sehingga dapat membuat *key stream* yang sama seperti pihak pengirim.

Fungsi dekripsi diberikan sebagai $x_i = y_i - k_i \pmod{26}$. Dua parameter awal tersebut harus dirahasiakan oleh kedua belah pihak agar pihak yang tidak berhak mengetahui pesan tersebut bisa mendapatkan *key stream* yang digunakan untuk enkripsi dan dekripsi. Proses dekripsi diberikan pada tabel di bawah ini.

Tabel 4. Dekripsi Menggunakan Fungsi Chaos (Fungsi Logistik)

Huruf	y_i	k_i	$x_i = y_i - k_i \pmod{26}$	Huruf
k	10	99	15	p
n	13	87	4	e
i	8	42	18	s
e	4	30	0	a
b	1	66	13	n
n	13	48	17	r
n	13	13	0	a
p	15	08	7	h
e	4	82	0	a
w	22	82	18	s
w	22	66	8	i
x	23	75	0	a
j	9	41	20	u
n	13	26	13	n
v	21	02	19	t
h	7	65	20	u
a	0	94	10	k
a	0	77	1	b
p	15	27	14	o
x	23	48	1	b

Dari tabel di atas, dapat dilihat bahwa proses perhitungan menghasilkan pesan semula yaitu "**pesan rahasia untuk bob**"

Diperhatikan bahwa apabila parameter awal diganti sedikit, yaitu $r = 3$ dan nilai awal $k_0 = 8$, dapat dilihat bahwa diperoleh *key stream* yang sangat berbeda saat nilai awalnya $k_0 = 7$ seperti yang telah dituliskan di atas, yaitu

$k_1 = 99832$

$k_2 = 85896$

$k_3 = 25720$

$k_4 = 21960$

$k_5 = 41080$

$k_6 = 91336$

$k_7 = 83096$

$k_8 = 50120$

$k_9 = 41752$

$k_{10} = 4040$

dan seterusnya.

Selain fungsi logistik di atas, Devaney (1992) juga memberikan contoh-contoh fungsi chaos, seperti fungsi

$$f(x)=\begin{cases} 2x & ,x\le\frac{1}{2} \\ 2-2x & ,x>\frac{1}{2} \end{cases}$$

yang bersifat chaos pada interval $[0,1]$, dan fungsi

$$f(x)=\begin{cases} 3x & ,x\le\frac{1}{3} \\ \frac{1}{3}x+\frac{3}{2} & ,x>\frac{1}{3} \end{cases}$$

yang bersifat chaos pada interval $[0,1]$. Ada beberapa cara untuk mendapatkan bilangan yang digunakan sebagai *key stream*. Salah satunya adalah dengan cara mengambil sejumlah angka pertama tidak nol pada bentuk desimalnya. Misalkan diperoleh hasil 0,007439716253... dapat diambil 7439 untuk digunakan sebagai kunci, atau diperoleh hasil 25,30115658... dapat diambil 253 sebagai kuncinya, yang terpenting adalah adanya kesepakatan antara pihak pengirim dan pihak penerima.

4. Penutup

Salah satu keuntungan dari *stream cipher* adalah tidak dibatasi oleh panjang plainteks, sehingga *stream cipher* cocok untuk digunakan pada enkripsi suatu komunikasi yang berlangsung secara berkelanjutan, seperti komunikasi melalui telepon. Fungsi chaos mempunyai sifat yang mendukung *stream cipher*, yaitu dapat diperlakukan secara iteratif, sehingga dapat dibuat *key stream* yang mengalir dan tidak terbatas. Selain itu juga sifatnya yang sensitif terhadap nilai awal, sehingga kunci rahasia berupa nilai awal apabila diganti akan menghasilkan *key stream* yang berbeda.

Untuk penelitian selanjutnya yang dapat dilakukan adalah dengan membuat algoritma yang lebih jelas dan terperinci dari *stream cipher* berdasarkan fungsi chaos tersebut, dan juga pemilihan fungsi chaos yang seperti apa agar diperoleh sistem yang aman. Oleh karena itu perlu dikaji lebih lanjut tentang kelemahan yang mungkin dapat terjadi untuk dapat dilakukan antisipasi. Pada contoh dalam makalah ini hanya digunakan grup $Z_{26} = \{0,1,2,...,25\}$. Pada penggunaan yang sebenarnya dilakukan pada bilangan-bilangan berbentuk biner yang terdiri dari 0 dan 1, yaitu menggunakan $Z_2 = \{0,1\}$, sehingga perlu juga dikembangkan cara untuk mendapatkan *key stream* dari hasil perhitungan fungsi chaos.

5. Daftar Pustaka

Devaney, Robert L., 1992, *A First Course in Chaotic Dynamical Systems: Theory and Experiment*, Addison-Wesley, Boston - Massachusetts.

Menezes Alfred J., Paul C. van Oorschot dan Scott A. Vanstone, 1996, *Handbook of Applied Cryptography*, CRC Press, USA.