Persamaan Lorenz untuk Keamanan Nomor Serial Sistem Operasi Window7

Dewanto Harwin Rohan[1], Nur Hayati[2]
Informatika, Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional
Jl. Sawo Manila No. 61 RT. 14 RW. 03 Pejaten Pasar Minggu Jakarta Selatan 12520
e-mail : [1]dewanto.hr@gmail.com, [2]nurh4y@gmail.com

Abstract

Serial number of operating system windows 7 needs to be safeguarded, so can't be used by the others. Security of the data can use by modern cryptography such as Vernam Cipher methods and classic cryptography such as Caesar Cipher methods. The security level both of this method depends on the keywords used and it will difficult to crack if the random key is used more and more. To get a random key, we can take from chaos of Lorenz equations as key-generator for encryption and description. Before utilizing chaos in the Lorenz equations, we have to find the maximum t (time) for the inverse problem solution to fit with the forward problem solution. We can use Runge-Kutta method in the Lorenz equations for forward problem solution and inverse problem solution. The solution of integral that obtained by the Runge-Kutta method can be searched by Trapezoidal method. The result of Runge-Kutta solution and Trapezoidal will be used as key-generator for encryption and description. In the simulations performed, the best orde in Runge-Kutta method is 4 and t max is 2. The encryption key is used as the initial condition of Lorenz equation, then the result is integrable by the Trapezoidal method. The result of orde 4 from Runge-Kutta method and Trapezoidal method used as a key-generator. Application of Lorenz equation as key-generator for encryption and decryption, may change the cryptography algorithms of symmetric to be asymmetric.

***Keyword** : Lorenz Equation, Runge-Kutta Method, Trapezoidal Method, Vernam Cipher Method, Caesar Cipher Method*

1. PENDAHULUAN

1.1. LATAR BELAKANG

Sistem operasi suatu komputer sangatlah mutlak dibutuhkan, karena tanpa adanya sistem operasi hardware komputer tidak dapat digunakan secara maksimal. Sistem operasi ini mulai dari memiliki fitur standar sampai lengkap dan mulai dari gratis sampai berbayar. Harga yang ditawarkan oleh vendor resmi untuk sistem operasi windows ini relatif mahal, sebagai contoh Microsoft Windows Proffesional OEM 7 yang pada saat ini dihargai sebesar Rp. 2.100.000 pada toko online www.nanokomputer.com. Tentu bukan harga yang murah bagi banyak kalangan apalagi pelajar dan mahasiswa yang belum mempunyai penghasilan. Hal ini tentu saja mengakibatkan tindakan ilegal seperti pencurian nomor serial number kerap seringkali terjadi dan bahkan diperjualbelikan dengan bebas.

Nomor serial windows 7 biasanya dapat dilihat di stiker yang ditempel, jika laptop ditempel di bawah laptop, jika komputer desktop ditempel di casing CPU, biasanya dibagian atas atau belakang. Pada gambar 1. dapat dilihat bahwa nomor serial windows 7 tidak ada pengamanan, sehingga dapat menimbulkan suatu masalah, dimana nomor serial sistem operasi windows 7 dapat dengan mudah untuk diambil oleh orang yang tidak berhak menggunakannya. Oleh karena itu perlunya suatu keamanan untuk nomor serial sistem operasi windows 7 yang ada di stiker tersebut.

Gambar 1. Nomor Serial Number di Laptop

Pengamanan suatu data dapat menggunakan kriptografi baik kriptografi modern maupun kriptografi klasik. Kriptografi modern salah satunya adalah dengan metode Vernam cipher sedangkan kriptografi klasik salah satunya dengan metode Caesar cipher. Tingkat keamanan dari kedua metode ini tergantung dari kunci yang digunakan, semakin acak kunci yang digunakan maka semakin sulit untuk di bobol. Mendapatkan sebuah kunci acak dapat memanfaatkan gejala chaos. Gejala chaos salah satunya terdapat di persamaan Lorenz. Chaos dari persamaan Lorenz ini dapat dimanfaatkan sebagai key-generator untuk keamanan nomor serial sistem operasi windows 7.

1.2. TUJUAN DAN KONTRIBUSI PENELITIAN

Tujuan dari penelitian ini adalah:

1. Mencari Runge-Kutta orde terbaik untuk solusi masalah invers pada persamaan Lorenz.
2. Mencari *t* maksimal solusi masalah invers yang sesuai dengan solusi masalah maju pada persamaan Lorenz.
3. Memanfaatkan *chaos* pada persamaan Lorenz untuk *key-generator* enkripsi dan *key-generator* dekripsi.
4. Merubah kriptografi algoritma simetris menjadi kriptografi algoritma asimetris.

Kontribusi dari penelitian ini adalah:

1. Penggunaan persamaan Lorenz yang bersifat chaos sebagai kunci enkripsi dan dekripsi memungkinkan perubahan algoritma kriptografi simetris menjadi asimetris
2. Program enkripsi dan dekripsi yang dibuat dapat mengamankan nomor serial sistem operasi windows 7, sehingga tidak dapat digunakan oleh orang lain yang tidak berhak menggunakannya.

1.3. BATASAN MASALAH

Batasan-batasan masalah dalam penelitian ini adalah:

1. Metode untuk mencari solusi persamaan diferensial biasa (persamaan Lorenz) adalah metode numerik Runge-Kutta orde 1, 2, 3, 4 dan orde lebih tinggi (*higher order*).
2. Metode untuk mencari solusi integral adalah metode numerik trapezoidal.
3. Proses enkripsi dan dekripsi menggunakan kriptografi modern metode Vernam *cipher* dan kriptografi klasik metode Caesar *cipher*.
4. Program yang dibuat adalah enkripsi dan dekripsi untuk keamanan nomor serial sistem operasi windows 7.
5. Perangkat lunak yang digunakan adalah *Scilab (5.4.0)* untuk simulasi dan *Visual Studio Express 2012 For Desktop (11.0.51106.1 Update 1)* dengan bahasa pemrograman *Visual Basic* untuk pembuatan program.

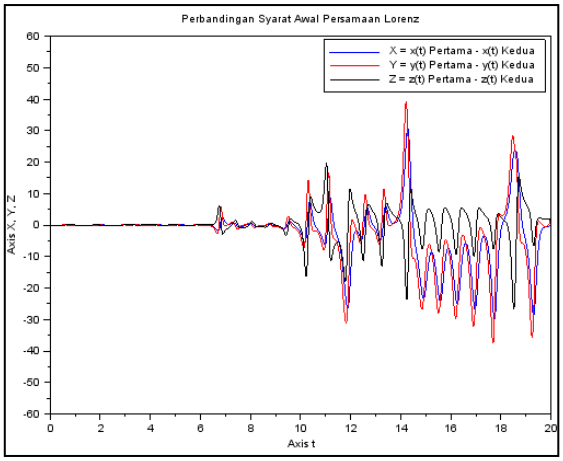
2. METODE PENELITIAN

2.1. PERSAMAAN LORENZ

Persamaan Lorenz yang dinamai sesuai penemunya, Edward N. Lorenz, merupakan persamaan diferensial biasa nonlinier masalah nilai awal. Persamaan Lorenz ini salah satu persamaan yang telah membawa orang pada pengetahuan tentang sistem yang *chaos*. Bentuk persamaan Lorenz adalah sebagai berikut[4]:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x); \\ \frac{dy}{dt} &= \rho x - y - xz; \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \tag{2.1}$$

Dimana σ, ρ dan β adalah parameter real positif, sedangkan x, y dan z adalah modus dominan dari aliran konveksi[1]. Parameter yang digunakan adalah[6] $\sigma = 10;\ \beta = \frac{8}{3};\ \rho = 28.$



Gambar 2. Solusi perbandingan syarat awal persamaan Lorenz dengan perbedaan 0,0001

2.2. MASALAH MAJU DAN MASALAH INVERS

Semua masalah sains dan teknologi yang dihadapi dapat dikategorikan menjadi tiga kelompok, yaitu masalah analisis, masalah pemodelan, dan masalah kendali[9].

Masalah analisis (masalah maju), maksud dari masalah ini adalah untuk mengetahui bagaimana atau hasil apa yang dikeluarkan dari suatu sistem, jika diberikan masukan pada sistem tersebut.

Masalah pemodelan (masalah invers 1), pada masalah ini adalah bila diketahui masukan dan keluaran pada suatu sistem yang tidak diketahui, maka akan dicari atau dibuat suatu sistem yang saling keterkaitan antara masukan dan keluaran.

Masalah kendali (masalah invers 2), di tahap ini masalahnya adalah dicari masukan apa agar keluaran dari suatu sistem sesuai, dimana sistem dan keluaran diketahui. Dapat diartikan juga untuk masalah invers ini dimana akibat dari masalah diketahui tetapi sebabnya tidak diketahui.

Masalah yang ada pada persamaan diferensial biasa masalah nilai awal (PDB MNA) adalah masalah analisis (masalah maju) jika yang diketahui adalah syarat awal, dan masalah kendali (masalah invers 2) jika diketahui adalah syarat akhir. Sedangkan untuk masalah pemodelan (masalah invers 1) tidak ada karena sistem selalu diketahui yaitu persamaan diferensial biasa itu sendiri.

Bentuk umum persamaan diferensial biasa masalah nilai awal sebagai berikut[9]:

$$u' = \frac{du}{dt} = f(t,u), \text{ dengan syarat awal } u(t_0) = u_0$$

Jika syarat awal $u(t_0) = u_0$ sebagai masukan dan persamaan diferensial biasa dianggap sebagai sistem maka masalah majunya adalah mencari nilai u_n yang memenuhi syarat awal sebagai hasil keluaran. Peubah t biasanya adalah waktu. Penyelesaian persamaan diferensial biasa secara numerik berarti menghitung nilai u di $t_{i+1} = t_i + h$ dengan $u(t_i)$ sebagai masukan dan h adalah jarak langkah setiap iterasi.

Masalah invers pada persamaan diferensial biasa adalah jika diketahui nilai keluaran $u(t_n) = u_n$ pada sistem persamaan diferensial biasa, yaitu nilai u pada titik $t = t_n$. Masalahnya adalah mencari nilai penyelesaian persamaan diferensial biasa secara numerik u di $t_{i-1} = t_i - h$, sehingga bentuk persamaan diferensial biasanya menjadi sebagai berikut[9]:

$$u' = \frac{du}{dt} = f(t,u), \text{ dengan syarat awal } u(t_n) = u_n$$

Jadi jika nilai $u(t_n) = u_n$ sebagai syarat awal diketahui dan persamaan diferensial biasa dianggap sebagai sistem juga diketahui, sedangkan $u(t_0) = u_0$ tidak diketahui dan harus dicari, maka kita menghadapi masalah invers.

2.3. SOLUSI PDB MNA DENGAN RUNGE-KUTTA

Metode Runge-Kutta adalah alternatif lain dari metode Taylor yang tidak membutuhkan perhitungan turunan. Metode ini berusaha mendapatkan derajat ketelitian yang lebih tinggi. Bentuk umum metode Runge-Kutta orde-n ialah[8]:

$$y_{i+1} = y_i + a_1k_1 + a_2k_2 + a_3k_3 + \cdots + a_nk_n \quad (2.2)$$

Dengan a_1 , a_2 , a_3 ... , a_n adalah tetapan, dan

$k_1 = hf(t_i , y_i)$

$$k_2 = hf\left(t_i + p_1 h\,, y_i + q_{1,1}k_1\right)$$
$$k_3 = hf\left(t_i + p_2 h\,, y_i + q_{2,1}k_1 + q_{2,2}k_2\right)$$
$$\vdots$$
$$k_n = hf\left(t_i + p_{n-1} h\,, y_i + q_{n-1,1}k_1 + q_{n-1,2}k_2 + \cdots + q_{n-1,n-1}k_{n-1}\right)$$

Nilai $a_i\,, p_i\,, q_{ij}$ dipilih sedemikian rupa sehingga menimbulkan galat per langkah.

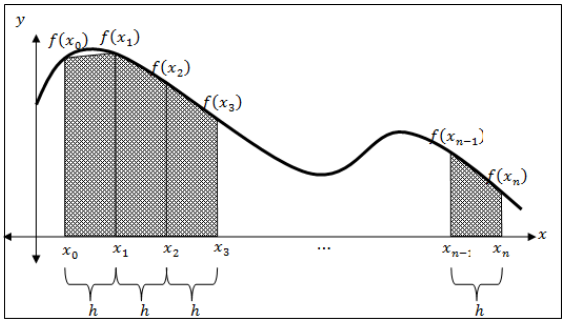
Metode Runge-Kutta ini selain dapat menyelesaikan masalah maju juga dapat menyelesaikan masalah invers. Untuk masalah invers diselesaikan dengan y_{i-1} dimana delta negatif $(-h)$, maka persamaan (2.3) menjadi:

$$y_{i-1} = y_i + a_1k_1 + a_2k_2 + a_3k_3 + \cdots + a_nk_n \tag{2.3}$$

Dengan $a_1\,, a_2\,, a_3\ \ldots\,, a_n$ adalah tetapan, dan
$k_1 = -hf(t_i\,, y_i)$
$k_2 = -hf\left(t_i - p_1 h\,, y_i + q_{1,1}k_1\right)$
$k_3 = -hf\left(t_i - p_2 h\,, y_i + q_{2,1}k_1 + q_{2,2}k_2\right)$
$\vdots$
$k_n = -hf\left(t_i - p_{n-1} h\,, y_i + q_{n-1,1}k_1 + q_{n-1,2}k_2 + \cdots + q_{n-1,n-1}k_{n-1}\right)$

2.4. SOLUSI INTEGRAL FUNGSI DENGAN *METODE* TRAPEZOIDAL

Metode trapezoidal tidak membutuhkan perhitungan yang rumit untuk mencari pendekatan integral dari suatu fungsi. Dasar dari metode ini adalah dengan cara mencari luas trapesium[8].



Gambar 3. Integral dengan metode trapezoidal banyak pias

Berdasarkan gambar (2.10) didapat pendekatan integral dengan banyak pias adalah:

$$\int_{x_0}^{x_n} f(x)\,dx \approx \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right) \tag{2.4}$$

2.5. PENGAMANAN DATA DENGAN KRIPTOGRAFI

Salah satu cara pengamanan suatu informasi adalah dengan metode Vernam *cipher*. Metode Vernam *cipher* ini merupakan algoritma simetri dimana kunci enkripsi dan dekripsi sama. Metode ini merubah plainteks menjadi biner. Enkripsi atau menentukan cipherteks pada metode Vernam *cipher* adalah [7]:

$$c_i = (p_i + k_i)\ mod\ 2 \tag{2.5}$$

Atau

$$c_i = p_i \oplus k_i \tag{2.6}$$

Dekripsi atau menentukan plainteks pada metode Vernam *cipher* adalah [7]:

$$p_i = (c_i + k_i)\ mod\ 2 \tag{2.7}$$

atau

$$p_i = c_i \oplus k_i \tag{2.8}$$

Tabel 1. Daftar perubahan plainteks menjadi real dan menjadi biner

Karakter	Real	Biner	Karakter	Real	Biner
A	0	000000	g	32	100000
B	1	000001	h	33	100001
C	2	000010	i	34	100010
D	3	000011	j	35	100011
E	4	000100	k	36	100100
F	5	000101	l	37	100101
G	6	000110	m	38	100110
H	7	000111	n	39	100111
I	8	001000	o	40	101000
J	9	001001	p	41	101001
K	10	001010	q	42	101010
L	11	001011	r	43	101011
M	12	001100	s	44	101100
N	13	001101	t	45	101101
O	14	001110	u	46	101110
P	15	001111	v	47	101111
Q	16	010000	w	48	110000
R	17	010001	x	49	110001
S	18	010010	y	50	110010
T	19	010011	z	51	110011
U	20	010100	0	52	110100
V	21	010101	1	53	110101
W	22	010010	2	54	110110
X	23	010111	3	55	110111
Y	24	011000	4	56	111000
Z	25	011001	5	57	111001
a	26	011010	6	58	111010
b	27	011011	7	59	111011
c	28	011100	8	60	111100
d	29	011101	9	61	111101
e	30	011110	#	62	111110
f	31	011111	&	63	111111

Metode Caesar *cipher* juga salah satu cara untuk mengamankan data. Caesar *cipher* yang digunakan kaisar Romawi, Julius Caesar ini adalah algoritma substitusi. Metode ini mengganti satu huruf plainteks dengan satu huruf cipherteks yang berbeda.
Enkripsi atau menetukan cipherteks pada metode Caesar *cipher*[7]:

$$c = (p + k)\ mod\ 64 \tag{2.9}$$

Dekripsi atau menetukan plainteks pada metode Caesar *cipher*[7]:

$$p = (c - k)\ mod\ 64 \tag{2.10}$$

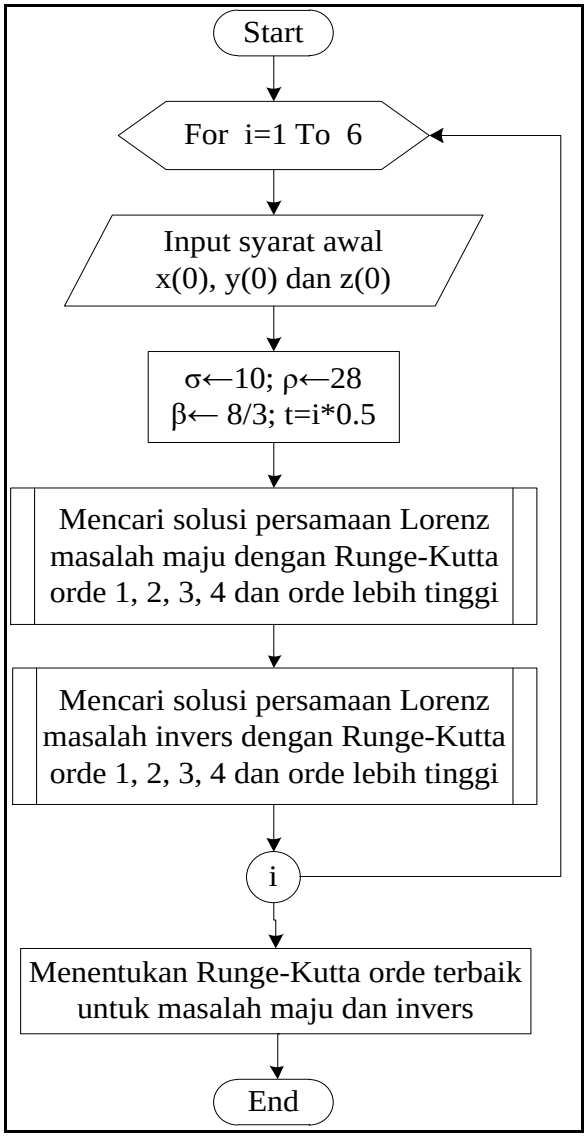
3. METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam penulisan ini adalah studi literatur dan simulasi, yaitu memberikan ilustrasi tentang alur dari proses enkripsi dan proses dekripsi untuk pengamanan nomor serial sistem operasi windows 7. Untuk mencapai tujuan penelitian ini, penulis melakukan langkah-langkah sebagai berikut:

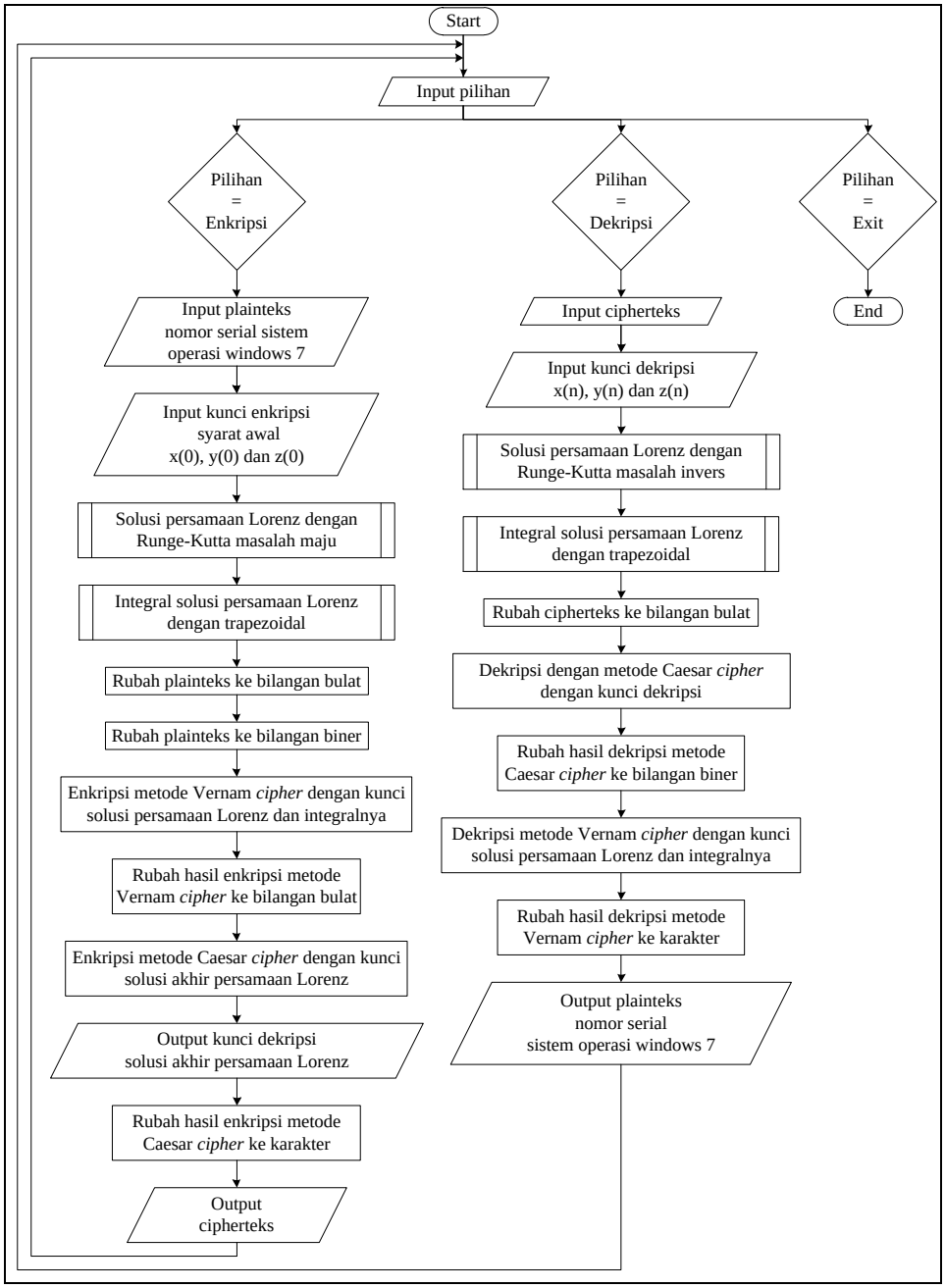
1. Studi literatur:
   a. Persamaan Lorenz.
   b. Masalah maju dan invers.
      i. Masalah maju pada PDB MNA.
      ii. Masalah invers pada PDB MNA.
   c. Solusi PDB MNA dengan metode Runge-Kutta.
      i. Runge-Kutta orde 1.
      ii. Runge-Kutta orde 2.
      iii. Runge-Kutta orde 3.
      iv. Runge-Kutta orde 4.
      v. Runge-Kutta orde lebih tinggi.
   d. Solusi integral dengan metode trapezoidal.
   e. Pengamanan data dengan kriptografi.
      i. Kriptografi metode Vernam *cipher*.
      ii. Kriptografi metode Caesar *cipher*.
2. Studi simulasi:
   a. Simulasi solusi persamaan Lorenz masalah maju dan masalah invers. Menggunakan metode Runge-Kutta orde 1, 2, 3, 4 dan orde lebih tinggi.
      Berikut adalah algoritma simulasi solusi persamaan Lorenz:
      i. Langkah 1: Masukkan syarat awal persamaan Lorenz $x(0), y(0), z(0)$ dan parameter $\sigma,\ \rho,\ \beta$ dan t akhir.
      ii. Langkah 2: Mencari solusi masalah maju dan solusi masalah invers dengan metode Runge-Kutta orde 1, 2, 3, 4 dan orde lebih tinggi.
      iii. Langkah 3: Membandingkan hasil masalah maju dengan masalah invers.
      iv. Langkah 4: Langkah 1, 2 dan 3 dilakukan berulang dengan *t* yang berbeda.
      v. Langkah 5: Menetukan Runge-Kutta dengan orde berapa yang terbaik untuk masalah maju dan invers.
   b. Simulasi integral dari solusi persamaan Lorenz. Menggunakan metode trapezoidal.
      Berikut adalah algoritma simulasi solusi persamaan Lorenz:
      i. Langkah 1: Mencari solusi masalah maju dan masalah invers dengan Runge-Kutta orde terbaik pada persamaan Lorenz.
      ii. Langkah 2: Masukkan batas bawah dan batas atas intergral.
      iii. Langkah 3: Mencari nilai integral (*t*, *x*), (*t*, *y*), dan (*t*, *z*) dari batas bawah sampai batas atas dengan metode trapezoidal.
3. Pengambilan data nomor serial sistem operasi windows 7.
   Data yang digunakan adalah data palsu yang diambil dari 25 x 10 bilangan acak dari 0 sampai 35. Bilangan tersebut dirubah kedalam bentuk karakter.
   Berikut adalah algoritma proses enkripsi:
   a. Langkah 1: Perulangan *i* dari 1 sampai dengan 10.
      i. Perulangan *j* dari 1 sampai dengan 25.
         i. Membuat bilangan acak dari 0 sampai dengan 35.
         ii. Merubah bilangan acak menjadi karakter.
4. Pembuatan program enkripsi nomor serial sistem operasi windows 7.
   Berikut adalah algoritma proses enkripsi:
   a. Langkah 1: Masukkan 25 nomor serial sistem operasi windows 7, ini sebagai plainteks.
   b. Langkah 2: Mencari solusi persamaan Lorenz masalah maju.
   c. Langkah 3: Mencari integral solusi persamaan Lorenz.
   d. Langkah 4: Enkripsi plainteks dengan metode Vernam *cipher*.
   e. Langkah 5: Enkripsi kembali dengan metode Caesar *cipher*.
   f. Langkah 6: Cetak solusi akhir persamaan Lorenz, ini sebagai kunci dekripsi.
   g. Langkah 7: Cetak cipherteks.

5. Pembuatan program dekripsi nomor serial sistem operasi windows 7.
   Berikut adalah algoritma proses dekripsi:
   a. Langkah 1: Masukkan 25 cipherteks plainteks.
   b. Langkah 2: Masukkan kunci dekripsi.
   c. Langkah 3: Mencari solusi persamaan Lorenz masalah invers.
   d. Langkah 4: Mencari integral solusi persamaan Lorenz.
   e. Langkah 5: Dekripsi cipherteks dengan metode Caesar *cipher*.
   f. Langkah 6: Dekripsi kembali dengan metode Vernam *cipher*.

g. Langkah 7: Cetak plainteks, ini sebagai nomor serial sistem operasi windows 7.



Gambar 4. Flowchart simulasi solusi persamaan Lorenz masalah maju dan masalah invers



Gambar 5. Flowchart enkripsi dan dekripsi nomor serial sistem operasi windows 7

4. HASIL DAN PEMBAHASAN

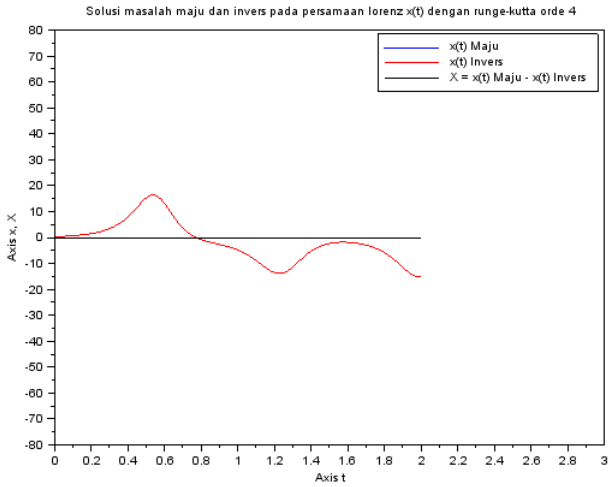
4.1. SOLUSI DENGAN METODE RUNGE-KUTTA UNTUK PERSAMAAN LORENZ

Persamaan Lorenz ini merupakan persamaan diferensial biasa orde satu sistem masalah nilai awal. Sehingga perlu disesuaikan untuk menyelesaikan persamaan Lorenz, jadi setiap iterasi delta (h) harus mencari tiga solusi $x(i+h)$, $y(i+h)$ dan $z(i+h)$ untuk masalah maju dan $x(i-h)$, $y(i-h)$ dan $z(i-h)$ untuk masalah invers.

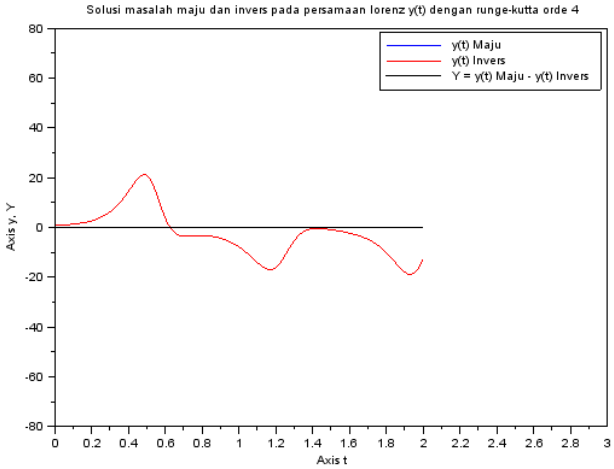
$$\begin{aligned} f_1(t,x,y,z) &= \frac{dx}{dt} = \sigma(y-x); \\ f_2(t,x,y,z) &= \frac{dy}{dt} = \rho x - y - xz; \\ f_3(t,x,y,z) &= \frac{dz}{dt} = xy - \beta z \end{aligned} \tag{3.1}$$

4.2. MASALAH MAJU DAN INVERS PADA PERSAMAAN LORENZ DENGAN RUNGE-KUTTA

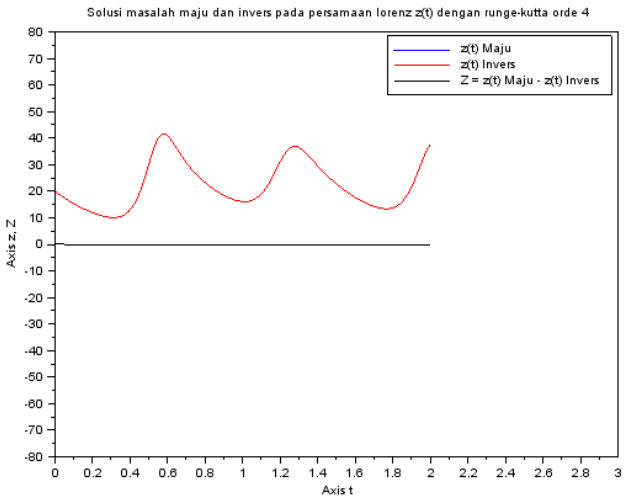
Diperlukannya penelitian untuk simulasi perbandingan solusi masalah invers dengan solusi masalah maju, karena jika terlalu jauh $\boldsymbol{t}$ solusi invers yang didapat tidak akan sama dengan solusi masalah maju. Menggunakan metode Runge-Kutta orde 1, 2, 3, 4 dan orde lebih tinggi, didapat orde 4 yang paling evisien, sedangkan $\boldsymbol{t}$ maksimal adalah 2.



Gambar 6. Perbandingan solusi masalah maju dan masalah invers pada persamaan Lorenz $t(x)$ untuk $t = 2{,}0$ dan orde 4

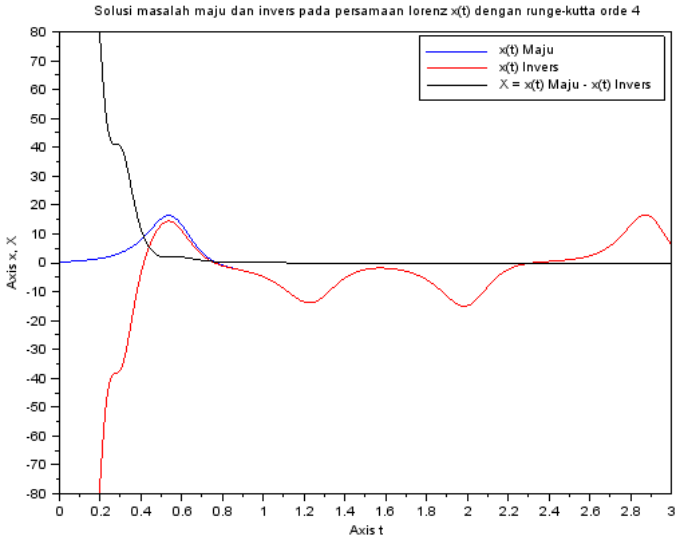


Gambar 7. Perbandingan solusi masalah maju dan masalah invers pada persamaan Lorenz $t(y)$ untuk $t = 2{,}0$ dan orde 4

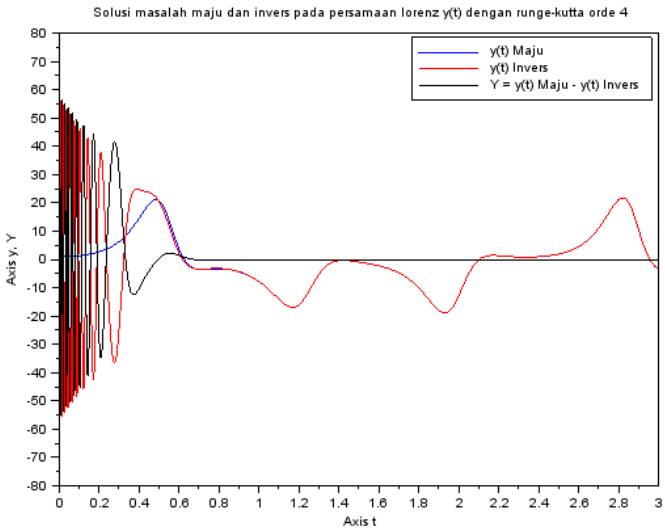


Gambar 8. Perbandingan solusi masalah maju dan masalah invers pada persamaan Lorenz $t(z)$ untuk $t = 2{,}0$ dan orde 4

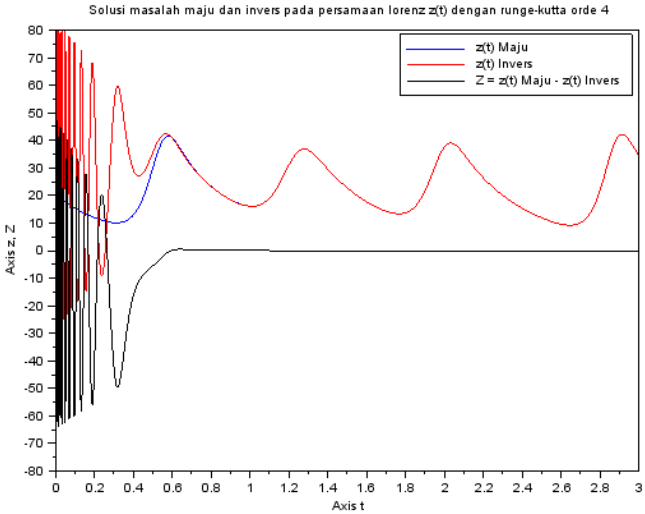
Jika digunakan t>2 maka solusi invers yang didapat tidak akan sama dengan solusi maju, sebagai berikut:



Gambar 9. Perbandingan solusi masalah maju dan masalah invers pada persamaan Lorenz $t(x)$ untuk $t = 3{,}0$ dan orde 4



Gambar 10. Perbandingan solusi masalah maju dan masalah invers pada persamaan Lorenz $t(y)$ untuk $t = 3{,}0$ dan orde 4



Gambar 11. Perbandingan solusi masalah maju dan masalah invers pada persamaan Lorenz $t(z)$ untuk $t = 3{,}0$ dan orde 4

4.3. INTEGRAL FUNGSI DENGAN METODE TRAPEZOIDAL UNTUK PERSAMAAN LORENZ

Integral fungsi yang dicari adalah fungsi dari solusi persamaan Lorenz $\boldsymbol{x(t)}$, $\boldsymbol{y(t)}$ dan $\boldsymbol{z(t)}$ dengan metode Runge-Kutta orde 4. Mencari integral dapat menggunakan persamaan (2.4) metode trapezoidal. Didapat hasil sebagai berikut:

Tabel 2. Integral masalah maju dan masalah invers

	Integral Masalah Maju	**Integral Masalah Invers**	**Kesalahan**
$x(t)$	-3,39722650394136	-3,39502261099620	0,00220389294516
$y(t)$	-4,88329565911341	-4,88493313761653	0,00163747850312
$z(t)$	44,3829035422656	44,3827566028110	0,0001469394546

Dari hasil simulasi yang dilakukan, dapat diambil kesimpulan bahwa integral untuk masalah maju dan masalah invers memiliki perbedaan yang sangat kecil.

4.4. ENKRIPSI NOMOR SERIAL SISTEM OPERASI WINDOWS 7 (PLAINTEKS)

Kunci enkripsi diambil dari hasil solusi persamaan Lorenz dan solusi integralnya. Nomor serial sistem operasi windows 7 ini akan dienkripsi dengan metode Vernam *cipher* dan Caesar *cipher*.

4.4.1 KUNCI ENKRIPSI

Kunci enkripsi ini adalah syarat awal dari persamaan Lorenz $x(0) = -88$, $y(0) = 52$ dan $z(0) = -20$.

4.4.2.KEY-GENERATOR ENKRIPSI DARI SOLUSI PERSAMAAN LORENZ

Solusi persamaan Lorenz masalah maju yaitu $\boldsymbol{x(t)}$, $\boldsymbol{y(t)}$ dan $\boldsymbol{z(t)}$, dapat digunakan untuk *key-generator* enkripsi.

$$dx \simeq |x(t_i)|$$
$$kger_{1,i} = dx \ mod \ 64$$
$$dy \simeq |y(t_i)|$$
$$kger_{2,i} = dy \ mod \ 64$$
$$dz \simeq |z(t_i)|$$
$$kger_{3,i} = dz \ mod \ 64$$

Dimana:

$i = 1, 2, 3, \cdots , 25$
$t_i = 0{,}08,\ 0{,}16,\ 0{,}24,\ \cdots,\ 2{,}00$
didapat:

Tabel 3. Kunci enkripsi dari solusi persamaan Lorenz

i	t_i	$kger_{1,i}$	$kger_{2,i}$	$kger_{3,i}$
1	0,08	42	59	39
2	0,16	4	32	17
3	0,24	27	51	39
4	0,32	16	25	58
5	0,40	9	27	45
6	0,48	12	4	51
7	0,56	3	7	39
8	0,64	3	7	32
9	0,72	5	6	28
10	0,80	5	6	25
11	0,88	6	8	23
12	0,96	8	10	23
13	1,04	10	12	26
14	1,12	11	11	30
15	1,20	10	8	31
16	1,28	8	6	30
17	1,36	6	5	27
18	1,44	6	6	24
19	1,52	7	8	22
20	1,60	8	10	23
21	1,68	10	12	26
22	1,76	11	11	30
23	1,84	10	8	31
24	1,92	8	5	29
25	2,00	6	5	26

4.4.3. *KEY-GENERATOR* ENKRIPSI DARI INTEGRAL SOLUSI PERSAMAAN LORENZ

Integral dari solusi persamaan Lorenz yaitu $\int x(t)\,dt$, $\int y(t)\,dt$ dan $\int z(t)\,dt$, dapat juga digunakan agar *key-generator* enkripsi semakin bervariasi.

$$ix \simeq \left| \left(\int_{a_i}^{b_i} x(t)\,dt \right) \times 10 \right|$$
$$ker_{4,i} = ix \bmod 64$$
$$iy \simeq \left| \left(\int_{a_i}^{b_i} y(t)\,dt \right) \times 10 \right|$$
$$ker_{5,i} = iy \bmod 64$$
$$iz \simeq \left| \left(\int_{a_i}^{b_i} z(t)\,dt \right) \times 10 \right|$$
$$ker_{6,i} = iz \bmod 64$$

dimana:

$i = 1, 2, 3, \cdots, 25$

$a_i = 0{,}00, \quad 0{,}08, \quad 0{,}16, \quad \cdots, \quad 1{,}92$
$b_i = 0{,}08, \quad 0{,}16, \quad 0{,}24, \quad \cdots, \quad 2{,}00$

didapat:

Tabel 4. Kunci enkripsi dari integral solusi persamaan Lorenz data pertama

i	a_i	b_i	$kger_{4,i}$	$kger_{5,i}$	$kger_{6,i}$
1	0,00	0,08	54	7	27
2	0,08	0,16	11	34	5
3	0,16	0,24	12	35	2
4	0,24	0,32	21	10	51
5	0,32	0,40	1	24	37
6	0,40	0,48	10	13	40
7	0,48	0,56	6	3	36
8	0,56	0,64	0	6	28
9	0,64	0,72	3	5	24
10	0,72	0,80	4	5	21
11	0,80	0,88	5	6	19
12	0,88	0,96	6	7	18
13	0,96	1,04	7	9	19
14	1,04	1,12	8	9	22
15	1,12	1,20	8	8	25
16	1,20	1,28	7	5	24
17	1,28	1,36	6	4	22
18	1,36	1,44	5	4	20
19	1,44	1,52	5	6	18
20	1,52	1,60	6	7	18
21	1,60	1,68	7	9	20
22	1,68	1,76	9	9	23
23	1,76	1,84	8	7	25
24	1,84	1,92	7	5	24
25	1,92	2,00	6	4	22

5. ENKRIPSI METODE VERNAM *CIPHER*

Sebelum menggunakan metode Vernam *cipher*, plainteks ($\boldsymbol{PTC}$) harus dirubah ke dalam bilangan biner ($\boldsymbol{PTB}$), begitu pula dengan *key-generator* enkripsi.

Setelah plainteks dan *key-generator* enkripsi dalam bentuk bilangan biner dilakukan proses enkripsi dengan sebanyak 6 kali dengan kunci yang ada. Berikut proses enkrips:

Enkripsi plainteks pertama:
101100-110000-001010-000111-110101-000111-000001-110101-001000-010000-000000-001010-001111-111100-000000-000101-010001-110001-110010-000101-001100-011100-001010-000010-010000

Enkripsi plainteks ke-2:
010111-010000-111001-011110-101110-000011-000110-110010-001110-010110-001000-000000-000011-110111-001000-000011-010100-110111-111010-001111-000000-010111-000010-000111-010101

Enkripsi plainteks ke-3:
110000-000001-011110-100100-000011-110000-100001-010010-010010-001111-011111-010111-011001-101001-010111-011101-001111-101111-101100-011000-011010-001001-011101-011010-001111

Enkripsi plainteks ke-4:
000110-001010-010010-110001-000010-111010-100111-010010-010001-001011-011010-010001-011110-100001-011111-011010-001001-101010-101001-011110-011101-000000-010101-011101-001001

Enkripsi plainteks ke-5:
000001-101000-110001-111011-011010-110111-100100-010100-010100-001110-011100-010110-010111-101000-010111-011111-001101-101110-101111-011001-010100-001001-010010-011000-001101

Enkripsi plainteks ke-6:
011010-101101-110011-001000-111111-011111-000000-001000-001100-011011-001111-000100-000100-111110-001110-000111-011011-111010-111101-001011-000000-011110-001011-000000-011011

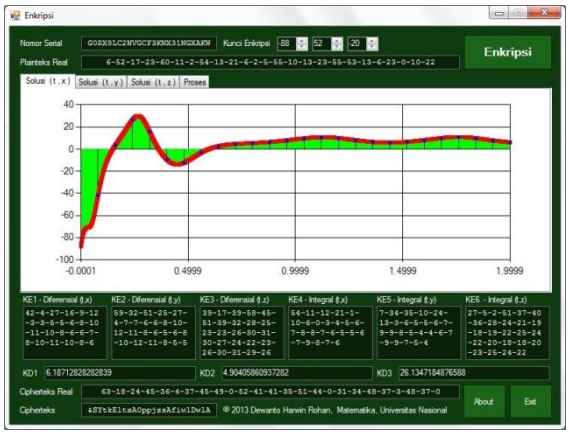
6. ENKRIPSI METODE CAESAR *CIPHER*

Enkripsi ke-6 dengan metode Vernam *cipher* dienkripsi sekali lagi dengan metode Caesar *cipher* agar hasil enkripsi akan semakin sulit untuk dipecahkan. Sebelum dienkripsi dengan metode Caesar *cipher*, hasil enkripsi Vernam *cipher* (***CTB***) harus dirubah kembali kedalam bilangan bulat (***CTR***). Kunci enkripsi Caesar *cipher* diambil dari hasil solusi persamaan Lorenz terakhir.

Proses enkripsi ke-7 menggunakan persamaan (2.9), sehingga didapat hasil pengamanan nomor serial sistem operasi windows 7 dengan penerapan persamaan Lorenz didalamnya adalah:

Cipherteks: &SYtkEltxA0ppjzsAfiwlDwlA
Kunci dekripsi 1: 6.18712828282839
Kunci dekripsi 2: 4.90405860937282
Kunci dekripsi 3: 26.1347184876588

Berikut tampilan form enkripsi program yang dibuat:



Gambar 12. Form enkripsi

4.5. DEKRIPSI NOMOR SERIAL SISTEM OPERASI WINDOWS 7 (CIPHERTEKS)

Proses dekripsi hampir sama dengan proses enkripsi dimana *key-generator* dari solusi persamaan Lorenz dan integral solusi persamaan Lorenz. *Key-generator* yang didapat, dilakukan proses dekripsi cipherteks dengan metode Caesar *cipher* dan Vernam *cipher*. Perbedaan hanya pada saat mencari *key-generator* dari solusi persamaan Lorenz, yaitu metode Runge-Kutta yang digunakan adalah untuk masalah invers dimana yang diketahui $\boldsymbol{x(2)}, \boldsymbol{y(2)}$ dan $\boldsymbol{z(2)}$.

3.5.1. KUNCI DEKRIPSI

Kunci dekripsi ini adalah syarat akhir dari persamaan Lorenz $x(2)$, $y(2)$ dan $z(2)$. Syarat awal ini diambil dari hasil enkripsi sebagai berikut:
Kunci dekripsi 1 $x(2)$: 6.18712828282839
Kunci dekripsi 2 $y(2)$: 4.90405860937282
Kunci dekripsi 3 $z(2)$: 26.1347184876588

3.5.2. *KEY-GENERATOR* DEKRIPSI DARI SOLUSI PERSAMAAN LORENZ

Solusi persamaan Lorenz masalah invers yaitu $\boldsymbol{x(t)}$, $\boldsymbol{y(t)}$ dan $\boldsymbol{z(t)}$ digunakan untuk *key-generator*, didapat hasil sebagai berikut:

Tabel 5. Kunci dekripsi dari solusi persamaan Lorenz

i	t_i	$kgdr_{1,i}$	$kgdr_{2,i}$	$kgdr_{3,i}$
1	0,08	42	59	39
2	0,16	4	32	17
3	0,24	27	51	39
4	0,32	16	25	58
5	0,40	9	27	45
6	0,48	12	4	51
7	0,56	3	7	39
8	0,64	3	7	32
9	0,72	5	6	28
10	0,80	5	6	25
11	0,88	6	8	23
12	0,96	8	10	23
13	1,04	10	12	26
14	1,12	11	11	30
15	1,20	10	8	31
16	1,28	8	6	30
17	1,36	6	5	27
18	1,44	6	6	24
19	1,52	7	8	22
20	1,60	8	10	23
21	1,68	10	12	26
22	1,76	11	11	30
23	1,84	10	8	31
24	1,92	8	5	29
25	2,00	6	5	26

3.5.3. *KEY-GENERATOR* DEKRIPSI DARI INTEGRAL SOLUSI PERSAMAAN LORENZ

Key-generator dari integral solusi persamaan Lorenz tidak ada perbedaan antara enkripsi dengan dekripsi, didapat hasil sebagai berikut:

Tabel 6. Kunci enkripsi dari integral solusi persamaan Lorenz data pertama

i	a_i	b_i	$kger_{4,i}$	$kger_{5,i}$	$kger_{6,i}$
1	0,00	0,08	54	7	27
2	0,08	0,16	11	34	5
3	0,16	0,24	12	35	2
4	0,24	0,32	21	10	51
5	0,32	0,40	1	24	37
6	0,40	0,48	10	13	40
7	0,48	0,56	6	3	36
8	0,56	0,64	0	6	28
9	0,64	0,72	3	5	24
10	0,72	0,80	4	5	21
11	0,80	0,88	5	6	19
12	0,88	0,96	6	7	18
13	0,96	1,04	7	9	19

14	1,04	1,12	8	9	22
15	1,12	1,20	8	8	25
16	1,20	1,28	7	5	24
17	1,28	1,36	6	4	22
18	1,36	1,44	5	4	20
19	1,44	1,52	5	6	18
20	1,52	1,60	6	7	18
21	1,60	1,68	7	9	20
22	1,68	1,76	9	9	23
23	1,76	1,84	8	7	25
24	1,84	1,92	7	5	24
25	1,92	2,00	6	4	22

3.5.4. DEKRIPSI METODE CAESAR *CIPHER*

Sebelum menggunakan metode Caesar *cipher* untuk dekripsi, cipherteks (CTC) harus dirubah ke dalam bilangan bulat atau real (CTR).

63-18-24-45-36-4-37-45-49-0-52-41-41-35-51-44-0-31-34-48-37-3-48-37-0

Kunci dekripsi yang digunakan adalah:

$$KDC = kgdr_{1,25} + kgdr_{2,25} + kgdr_{3,25}$$

Hasil proses dekripsi pertama sebagai berikut:

26-45-51-8-63-31-0-8-12-27-15-4-4-62-14-7-27-58-61-11-0-30-11-0-27

3.5.5. DEKRIPSI METODE VERNAM *CIPHER*

Sebelum menggunakan metode Vernam *cipher* untuk dekripsi, cipherteks hasil dekripsi dengan metode Caesar *cipher* (CTR) harus dirubah ke dalam bilangan biner (CTB), begitu pula dengan *key-generator* dekripsi.

Setelah cipherteks dan *key-generator* dalam bentuk bilangan biner, dilakukan proses dekripsi dengan sebanyak 6 kali dengan kunci yang ada.

Berikut proses dekripsi:

Dekripsi cipherteks ke-2:
110000-101001-101000-011000-110110-010011-000011-001011-001001-011110-001001-001100-001110-110101-000100-001111-011101-111100-111010-000011-001010-010101-000001-001000-011101

Dekripsi cipherteks ke-3:
001011-001001-011011-000001-101101-010111-000100-001100-001111-011000-000001-000110-000010-111110-001100-001001-011000-111010-110010-001001-000110-011110-001001-001101-011000

Dekripsi cipherteks ke-4:
101100-011000-111100-111011-000000-100100-100011-101100-010011-000001-010110-010001-011000-100000-010011-010111-000011-100010-100100-011110-011100-000000-010110-010000-000010

Dekripsi cipherteks ke-5:
011010-010011-110000-101110-000001-101110-100101-101100-010000-000101-010011-010111-011111-101000-011011-010000-000101-100111-100001-011000-011011-001001-011110-010111-000100

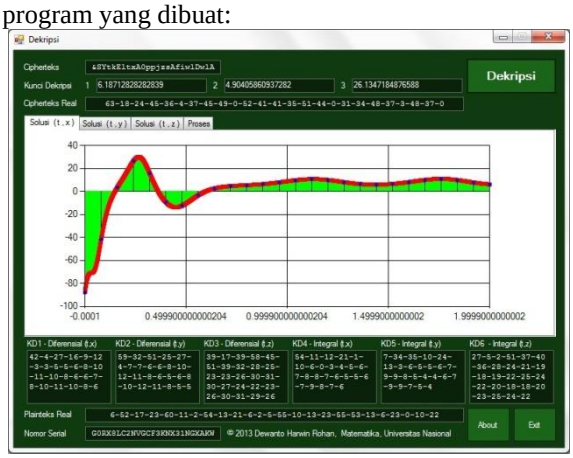
Dekripsi cipherteks ke-6:
011101-110001-010011-100100-011001-100011-100110-101010-010101-000000-010101-010000-010110-100001-010011-010101-000001-100011-100111-011111-010010-000000-011001-010010-000000

Dekripsi cipherteks ke-7:
000110-110100-010001-010111-111100-001011-000010-110110-001101-010101-000110-000010-000101-110111-001010-001101-010111-110111-110101-001101-000110-010111-000000-001010-010110

Proses dekripsi ke-7 ini adalah plainteks atau nomor serial sistem operasi windows 7 jika dirubah kedalam karakter, maka nomor serial sistem operasi windows 7 adalah:

G0RX8 − LC2NV − GCF3K − NX31N − GXAKW.

Berikut tampilan form dekripsi program yang dibuat:



Gambar 13. Form dekripsi

5. KESIMPULAN

Berdasarkan hasil yang diperoleh dari pembahasan, dapat diambil kesimpulan:

1. Metode Runge-Kutta orde 4 merupakan orde terbaik untuk mencari solusi masalah invers pada persamaan Lorenz.
2. Solusi invers pada persamaan Lorenz hanya dapat dicari dengan t maksimal adalah 2.
3. *Chaos* yang terdapat pada solusi persamaan Lorenz, dapat dimanfaatkan sebagai *key-generator* untuk enkripsi dan dekripsi.

Penerapan persamaan Lorenz sebagai *key-generator* untuk enkripsi dan dekripsi, dapat merubah kriptografi algoritma simetris menjadi kriptografi algoritma asimetris.

Daftar Pustaka

[1] Alatas, Husin, *Buku Pelengkap Dinamika Nonlinier Edisi I*, Depertemen Fisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor.
[2] Ali, Sidik M., *Implementasi Aloritma DES dan Cryptanalysis Pada Kartu ATM Eurocheque Menggunakan Teori Probabilitas*, Program Studi Matematika, Fakultas Teknik dan Sains, Universitas Nasional, Jakarta, 2012.
[3] Azizah, Siti S., *Diskretisasi Model Lorenz dengan Analogi Persamaan Beda*, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, 2012.
[4] Bunjamin, M., *Pemodelan, Komputasi, dan Simulasi dalam Sains dan Teknologi*.
[5] Chapra, Steven C. dan Canale, Raymond P., *Numerical Methods for Engineers with Software and Programing Applications Fourth Edition*, McGrawHill, New York, 2002.
[6] Lorenz, Edwar N., *Deterministic Nonperiodic Flow*, Massachusetts Institute of Technology, 1963.
[7] Munir, Renaldi, *Kriptografi*, Informatika, Bandung, 2006.
[8] Munir, Renaldi, *Metode Numerik Edisi Revisi*, Informatika, Bandung, 2006.
[9] Morisson, Marchi E., *Solusi Masalah Invers untuk Persamaan Diferensial Van Der Pol dan Laplace*, Program Studi Matematika, Fakultas Teknik dan Sains, Universitas Nasional, Jakarta, 2008.
[10] Nasution, Amrinsyah dan Zakaria, Hasballah, *Metode Numerik Dalam Ilmu Rekayasa Sipil*, ITB Bandung, Bandung, 2001.
[11] Nasution, Amrinsyah dan Zakaria, Hasballah, *Metode Numerik Dalam Ilmu Rekayasa Sipil*, ITB Bandung, Bandung, 2001.
[12] Pfleeger, Charles P., *Security in Computing Second Edition*, Prentice Hall, United States of America, 1997.