Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 1, No 3, Juli 2022
Hal 71−81

Modifikasi Algoritma Vigenere Cipher dengan Pembangkit Kunci Random Number Generator Dalam Pengamanan Citra Digital

Shafira Amalia Zebua

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: [1]sahazeb@gmail.com

Abstrak−Citra Digital merupakan suatu gambaran yang dapat diolah oleh komputer. Citra Digital yang bersifat tidak boleh diketahui orang lain atau dikatakan bersifat pribadi dan tidak diperbolehkan untuk dimanipulasi pihak lain yang tidak bertanggung jawab. Tindakan memanipulasi, membajak atau menyadap citra tersebut akan membuat kerugian bagi pemilik citra. Teknik untuk mengurangi tindakan tersebut yaitu menggunakan teknik kriptografi. Teknik kriptografi sangat terkenal untuk mengamankan suatu informasi dengan kata sandi yang hanya dimengerti oleh pemilik citra. Untuk memperkuat hasil dari enkripsi yang dilakukan Algoritma Vigenere Cipher, maka dilakukan modifikasi dengan Pembangkit Bilangan Random Number Generator. Random Number Generator yang digunakan adalah Linear Congruent Generator mempunyai sifat rekursif linear yang dikombinasi dengan modulus. Jadi penggabungan antara vigenere cipher dengan pembangkit bilangan acak Linear Congruent Generator sangat mempersulit untuk pihak lain untuk menyalahgunakan citra tersebut.

Kata Kunci: Citra Digital; Vigenere Cipher; LCG

Abstract−Digital image is an image that can be processed by a computer. Digital images that are not allowed to be known by others or are said to be private and are not allowed to be manipulated by other irresponsible parties. The act of manipulating, hijacking or tapping the image will cause harm to the owner of the image. The technique to reduce these actions is to use cryptographic techniques. Cryptographic techniques are very well known for securing information with a password that only the owner of the image understands. To strengthen the results of the encryption carried out by the Vigenere Cipher Algorithm, modifications are made to the Random Number Generator. The Random Number Generator used is Linear Congruent Generator which has linear recursive properties combined with modulus. So the combination of vigenere cipher with random number generator Linear Congruent Generator is very difficult for other parties to abuse the image.

Keywords: Digital Image; Vigenere Cipher; LCG

1. PENDAHULUAN

Masalah kerahasiaan dan keamanan file merupakan suatu aspek yang penting dari suatu informasi, data dan pesan. Hal tersebut sangat berhubungan dengan keamanan informasi, data dan pesan yang dikirim atau diterima oleh pihakyang berkepentingan. Informasi yang didistribusikan dalam berbentuk teks, audio,video maupun citra. Proses pendistribusian data bila tidak disertai dengan pengamanan, maka sangat rentang terhadap upaya penyerangan oleh pihak yang tidak bertanggung jawab. Salah satu jenis data yang sering umum digunakan dalam komunikasi adalah data dalam bentuk citra digital. Data dalam bentuk citra digital dapat digunakan sebagai media untuk mendistribusikan informasi yang bersifat *public* (umum) maupun *private* (rahasia). Tentu saja gambar rahasia tidak boleh diakses oleh sembarangan diakses oleh pihak-pihak yang tidak memiliki kepentingan pada gambar tersebut. Apabila pihak lain mendapatkan kerahasiaan dari gambar tersebut, maka gambar akan rusak bahkan dapat hilang dan akan mengakibatkan kerugian pada pemilik gambar yang memiliki kerahasiaan [1]

Teknik kriptografi merupakan salah satu teknik yang umum dan dapat digunakan dalam pengamanan data. Kriptografi merupakan cara yang digunakan untuk mengamankan data, yaitu dengan cara merubah informasi pesan menjadi kode yang tidak akan di mengerti sehingga orang lain tidak mengetahui isi dari informasi yang ada. Kriptografi bisa menjaga kerahasiaan suatu data supaya tetap terjaga kerahasiaannya berdasarkan kaidah atau algoritmanya. Berdasarkan penelitian sebelumnya, menyatakan bahwa kekuatan kriptografi terdapat padakuncinya, bukan dari hasil enkripsi ataupun *ciphertext*. Kunci kriptografi bisa dikatakan sebagai jantung dari pertahanan pengamanan data, karena kunci adalah alat untuk menjembatani proses dari enkripsi-deskripsi atau proses dari deskripsi - enkripsi [2] Salah satu algoritma dari kriprografi untuk pengamanan data yaitu Algoritma *Vigenere Cipher*.

Vigenere chiper merupakan salah satu algoritma kriptografi klasik untuk menyandikan suatu *plaintext* dengan menggunakan teknik substitusi. Sandi Vigenère terdiri dari beberapa sandi *Caesar* dengan nilai geseran yang berbeda. Proses penyandian pesan dilakukan dengan menggunakan sebuah tabel alfabet yang disebut tabel *Vigenère, tabel Vigenère* berisi alfabet yang dituliskan dalam 26 baris, masing-masing baris digeser satu urutan ke kiri dari baris sebelumnya, membentuk ke-26 kemungkinan sandi *Caesar*. Setiap huruf disandikan dengan menggunakan baris yang berbeda-beda, sesuai kata kunci yang diulang. Sandi ini dikenal luas karena cara kerjanya mudah dimengerti dan dijalankan, dan bagi parapemula sulit dipecahkan [3].

Vigenere cipher pada dasarnya cukup rumit untuk dipecahkan, namun v*igenere cipher* memiliki kelemahan yaitu panjang kuncinya dapat diketahui dengan menggunakan metode kasiski[4]. Hal ini tejadi karena umumnya terdapat frasa yang berulang-ulang pada *ciphertext* yang dihasilkan. Salah satu solusi yang dapat dilakukan untuk memperbaiki kelemahan algoritma *Vigenere Cipher* adalahmelakukan modifikasi terhadap pembangkitan kunci yang lebih acak. Salah satu teknik pembangkit bilangan acak adalah *Random Number Generator*.

Random Number Generator (RNG) merupakan pembangkit bilangan acak secara nulerik/aritmatika menggunakan komputer yang sering digunakan untuk proses perhitungan dalam simulasi. Umumnya pembangkit bilangan acak (RNG)

harus berdistribusi *uniform* dengan nilai 0 dan 1 dimana tidak ada korelasi antar bilangan, membangkitkan dengan cepat, sehingga *storage* yang digunakan tidak besar, dan periode yang terjadi besar, karena bilangan acak dapat dibangkitkan berulang[5].

Penelitian ini menguraikan bagaimana memodifikasi algoritma *VigenereCipher* khususnya untuk melakukan pembangkitan kunci secara acak berdasarkan pembangkit kunci *Random Number Generator* (RNG). Algoritma yang telah dimodifikasi akan diimplementasikan untuk mengamankan ctra digital berwarna.

2. METODOLOGI PENELITIAN

2.1 Kriptigrafi

Seni dan ilmu menjaga keamanan pesan adalah kriptografi (cryptography) sedangkan orang yang menggunakannya disebut cryptograph[6]. Kriptanalis (cryptanalysts) adalah praktisi dari kriptanalis, seni dan ilmu untuk memecahkan ciphertext yaitu menampilkan dengan samaran. Cabang dari matematika yang mencakup kriptografi dan kriptanalis adalah kriptologi (cryptology) dan praktisinya disebut kriptologis (cryptologists). Cryptologis modern biasanya dilatih dari belajar matematika. Suatu pesan adalah plainteks (plaintext) kadang juga disebut dengan cleartext. Proses menyembunyikan pesan disebut enkripsi (encryption). Pesanyang di enkripsi adalah cipherteks (ciphertext). Proses pengembalian ciphertekske plainteks disebut dekripsi (decryption). Plainteks ditandai dengan M untuk pesan, atau P untuk plainteks. Ini bisa merupakan suatu aliran bit, file teks,bitmap, aliran suara digital, dan citra video digital, sedangkan M adalah databiner. Cipherteks ditandai dengan C. Ini juga data biner, kadang-kadang ukurannya sama seperti M. Fungsi enkripsi E, beroperasi pada M untuk menghasilkan C[7].

2.2 Algoritma Vigenere Cipher

Sandi *Vigenere* awalnya digambarkan oleh Giovan Battista Bellaso dalam bukunya " *La cifra del. Sig. Giovan Battista Bellaso*" pada tahun 1553. Ia membangun sandi atas *tabula recta Trithemius*, tetapi menambahkan sebuah kunci perulangan untuk menukar setiap huruf abjad sandi. Sandi *Vigenère* sebenarnya merupakan pengembangan dari sandi *Caesar*[3]. Setiap huruf teks terang pada sandi *Caesar* digantikan dengan huruf lain yang memiliki perbedaan tertentu pada urutan alfabet. Misalnya pada sandi *Caesar* dengan geseran 3, A menjadi D, B menjadi E and dan seterusnya. Sandi *Vigenère* terdiri dari beberapa sandi *Caesar* dengan nilai geseran yang berbeda. penyandian suatu pesan, digunakan sebuah tabel alfabet yang disebut tabel *Vigenère*, tabel *Vigenère* berisi alfabet yang dituliskan dalam 26 baris, masing-masing baris digeser satu urutan kekiri dari baris sebelumnya, membentuk ke-26 kemungkinan sandi *Caesar*. Setiap huruf disandikan dengan menggunakan baris yang berbeda-beda, sesuai kata kunci yang diulang. Sandi ini dikenal luas karena cara kerjanya mudah di- mengerti dan dijalankan, dan bagi para pemula sulit dipecahkan[3]. Teknik dari *Vigenere Cipher* dapat dilakukan dengan 2 cara yaitu Angka dan Huruf [7].

1. Angka
   Teknik ini hampir sama dengan teknik geser atau teknik subsitusi denganmenggantikan huruf dengan angka
2. Huruf
   Metode lain untuk melakukan proses enkripsi dengan metode *vigènere cipher* yaitu menggunakan *tabula recta* (disebut juga bujursangkar *vigènere*). Berikut adalah cara *Vigenere Cipher* dengan menggunakan huruf, dengan bantuan tabeldibawah ini, kita bisa melihat untuk menentukan *Ciphertext*. Pada posisi vertikal adalah KUNCI, dan posisi horizontal adalah *PLAINTEXT*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 1. *Tabula recta*

Langkah-langkah enkripsi berdasarkan *Vigenere Cipher*[7] adalah:

1. Gunakan tabel *Vigenere Cipher*
2. Tabel yang memuat abjad yang ditulis 26 kali dalam baris yang berbeda,dengan tiap-tiap abjad digeser memutar kekiri terhadap abjad sebelumnya,bersesuaian dengan 26 kemungkinan *Caesar Cipher*.
3. Pada tiap huruf berbeda dalam proses enkripsi, pembuat kode menggunakan abjad berbeda dari salah satu baris.
4. Abjad yang digunakan pada setiap huruf disesuaikan kepada kata kunci yang berulang

Langkah-langkah deskripsi berdasarkan *Vigenere Cipher* [7] adalah:

1. Huruf pertama *cipherteks* dan kita sambungkan pada huruf pertama kata kunci pada baris.
2. Tarik ke samping kanan sampai ketemu dengan *cipherteks*.
3. Tarik ke atas sehingga kita mendapatkan huruf baru dari tabel tersebut, dan huruf itu merupakan huruf pertama pada *plaintext* dan begitu seterusnya.

Teknik *Vigenere Cipher* ini melakukan enkripsi dengan bantuan tabel *Vigenere*. Kelebihan dari *Vigenere Cipher* dari metode klasik yang sebelumnya yaitu subsitusi untuk *Cipher* nya mempunyai sifat *polyalphabetic* yaitu satu karakter yang sama dapat mempunyai karakter subsitusi yang berbeda. Tetapi teknik ini mempunyai kelemahan yaitu jumlah karakter di plainteks akan mendapatkan jumlah karakter kunci yang sama, karena jarak antara kedua jumlah kata tersebut adalah kelipatan dari jumlah kunci yang digunakan[4].

2.3 Linear Congruent Generator (LCG)

Linear Congruent Generator adalah salah satu dari metode *Random Number Generator* mempunyai sifat rekursif linear yang dikombinasi denganmodulus[8]. Berdasarkan kebutuhan untuk pengamanan yang menggunakan algoritma *Vigenere Cipher*, maka dilakukan modifikasi dengan *Linear Congruent Generator* diharapkan dapat membangkitkan kunci yang lebih kuat. Metode membangkitkan bilangan acak Linear menggunakan rumus:

$$Zi = (a * Zi -1 + c) \ mod \ m \quad (1)$$

Rumus di atas dibutuhkan pembangkit yang disebut dengan umpan (*seed*) yang merupakan kunci pembangkitnya adalah Z_0.

3. HASIL DAN PEMBAHASAN

Penggunaan citra *digital* pada era serba *digital* sangat luas dan penting, selain itu perangkat untuk membuat citra *digital* sudah sangat murah, terlebih dengan hadirnya perangkat *smartphone* yang hampir seluruhnya didukung dengan*camera.* Faktor yang lain yang menjadikan citra *digital* menjadi sangat penting adalah bahwa citra *digital* dapat digunakan untuk menyampaikan informasi baik informasi penting dan rahasia. Bahkan hampir bisa disimpulkan bahwa semua aplikasi sosial media, toko-toko *online* dan aplikasi *chating* menyediakan *button* untuk pendistribusian citra *digital*.

Berdasarkan permasalahan, maka pengamanan dalam pendistribusian citra *digital* harus benar-benar diperhatikan, sebab bila citra tidak aman pada saatpendistribusian dapat terjadi hal-hal yang merugikan bagi pihak pengirim dan juga penerima dan pihak lain. Penyerangan dapat saja dilakukan pada saat pendistribusian seperti pencurian, manipulasi dan pemantauan distribusi oleh pihak-pihak yang tidak bertanggungjawab.

Kemajuan teknologi dan pengetahuan saat ini menyebabkan banyak dikembangkan berbagai cara untuk melakukan penyerangan terhadap data yang telah diamankan berdasarkan algoritma dari teknik kriptografi, sehingga para penyerang dapat mengambil dan mengetahui makna dari data asli yangdidistribusikan, termasuk citra digital.

Kekuatan teknik kriptografi sepenuhnya bergantung pada kekuatan kuncinya, bukan metodenya atau algoritma yang digunakan. Bila kekuatan suatu metode atau algoritma mudah dipecahkan atau diserang, maka algoritma tersebut sudah tidak aman lagi untuk digunakan, namun masih dapat digunakan bila metode tersebut dimodifikasi untuk memperkuat kuncinya.

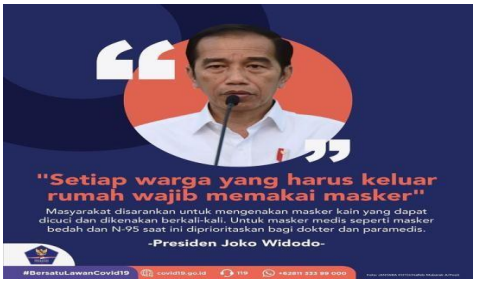
Teknik kriptografi mengamankan data dengan merubah karakter data asli menjadi karakter atau simbol-simbol yang sulit dipahami lagi makananya sehingga membutuhkan waktu yang sangat lama dan rumit untuk memecahkannya.

3.1.1 Penerapan Algoritma Vigenere Cipher dan Linear Congruent Generator (LCG)

Agar prosedur penerapan metode *LCG* tersebut bekerja mengoptimalkan kekuatan kunci algoritma *vigenere cipher*, maka dilakukan pengamanan terhadap citra *digital RGB* dengan format *Jpg*. Adapun ukuran citra yang akan digunakan menjadi sampel dalam penelitian ini adalah ukuran 256 x 256, kemudian diambil 10 x 10 pixel sebagai bahan untuk melakukan proses enkripsi dan dekripsi. Berikut ini akan diuraikan bagaimana kedua metode tersebut bekerja sama untuk mengamankan citra *digital.*

1. Defenisikan citra yang akan dienkripsi.

Adapun objek yang diteliti adalah citra *RGB* berformat *.jpg.* Berikut ini adalah citra yang akan dimankan yang ditampilkan dengan *Matlab* 2013. Adapun perintah untuk membaca gambar dari direktori komputer adalah sebagai berikut:



Gambar 1. *Plain Image*

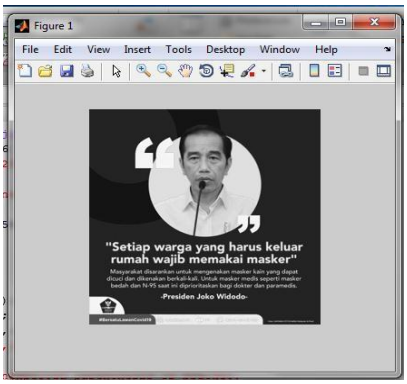
2. Mengubah ukuran citra

Ukuran citra ini perlu diketahui lebih dahulu agar dapat mengetahui jumlah kunci yang akan dibangkitkan, seperti telah diuraikan di atas bahwa *vigenere cipher* beroperasi bila setiap elemen memiliki pasangan kunci, ukuran citra yang akan diproses adalah berukuran 256 x 256, jadi berapapun ukuran citranya maka ukuran citra tersebut harus diubah lebih dahulu menjadi 256 x 256, adapun perintah untuk mengubah citra ukuran citra menjadi 256 x 256 dengan *matlab* 2013 adalah:

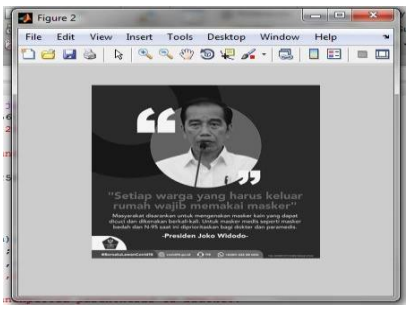
Gambar 2. *Plain image* Setelah Ukurannya Diubah

3. Isolasi setiap komponen warna *RGB* menjadi *array* tersendiri

Proses yang dilakukan untuk mendefeniskan nilai *RGB* citra safira.jpg, maka warna citra komponen RGB citra diuraikan ke dalam tiga buah variabel masingvariabel *Red*, *Green*, *Blue.* Berikut ditampilkan tabel distribusi warna RGB citra yang ditampilkan dengan menggunakan *matlab* 2013. Berikut akan ditampilkan citra dan nilai piksenya setelah setiap gambar diisolasi berdasarkan warna *RGB*-nya, yang ditampilkan dengan *matlab* 2013.



Gambar 3. *Plain Image* merah (variabel *Red*)



Gambar 4. *Plain Image* hijau (variabel *Green*)

Gambar 4. *PlainImage* biru(variabel *Blue*)

Nilai masing-masing piksel setelah diisolasi berdasarkan warna *RGB* ditampilkan dalam matriks 10 x 10 sebagai sampel pada tabel berikut yang dibaca dengan menggunakan *matlab* 2013.

Tabel 1. nilai piksel merah

index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	33	35	35	35	35	35	35	35	35	35
1	35	36	35	35	35	35	35	35	35	35
2	35	35	35	35	35	35	35	35	35	35
3	35	35	35	35	35	35	35	35	35	35
4	35	35	35	35	35	35	35	35	35	35
5	35	35	35	35	35	35	35	35	35	35
6	35	35	35	35	35	35	35	35	35	35
7	35	35	35	35	35	35	35	35	35	35
8	35	35	35	35	35	35	35	35	35	35
9	35	35	35	35	35	35	35	35	35	35

Tabel di atas adalah tabel distribusi warna merah pada citra safira.jpg yangditampilkan dengan menggunakan *matlab* 2013.

Tabel 2. nilai piksel hijau

index	0	1	2	3	4	5	6	7	8	9
0	40	42	42	42	42	42	42	42	42	42
1	42	43	42	42	42	42	42	42	42	42
2	42	42	42	42	42	42	42	42	42	42
3	42	42	42	42	42	42	42	42	42	42
4	42	42	42	42	42	42	42	42	42	42
5	42	42	42	42	42	42	42	42	42	42
6	42	42	42	42	42	42	42	42	42	42
7	42	42	42	42	42	42	42	42	42	42
8	42	42	42	42	42	42	42	42	42	42
9	42	42	42	42	42	42	42	42	42	42

Tabel di atas adalah tabel distribusi warna hijau pada citra safira.jpg yangditampilkan dengan menggunakan *matlab* 2013

Tabel 3. nilai piksel biru

index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	94	96	96	96	96	96	96	96	96	96
1	96	97	96	96	96	96	96	96	96	96
2	96	96	96	96	96	96	96	96	96	96
3	96	96	96	96	96	96	96	96	96	96
4	96	96	96	96	96	96	96	96	96	96
5	96	96	96	96	96	96	96	96	96	96
6	96	96	96	96	96	96	96	96	96	96
7	96	96	96	96	96	96	96	96	96	96
8	96	96	96	96	96	96	96	96	96	96
9	96	96	96	96	96	96	96	96	96	96

Tabel di atas adalah tabel distribusi warna biru pada citra safira.jpg yangditampilkan dengan menggunakan *matlab* 2013

1. Memilih tiga buah bilangan secara acak untuk melakukan pembangkitan kunci berdasarkan *LCG*.

a = 3, c =7, z = 5.

2. Faktor pembagi m = 753.
   Alasan utama pemilihan 753 menjadi faktor pembagi adalah memenuhi syarat keamanan *LCG* yaitu c dan m relatif prima karena *GCD*(7, 753)= 1, berikut uraiannya.
   Faktor 7 = 1, 7.
   Faktor 753 = 1, 3, 251, 753.
   Jadi m=753 adalah memenuhi syarat untuk menjadi faktor pembagi, karena 7 dan 753 adalah bilangan yang relatif prima. Selain itu dengan menggunakan faktor pembagi bilangan besar dapat meminimalisasi periode kemunculan angka, walau masih memungkinkan menggunakan yang lebih besar, namun diyakini bilangan ini sudah cukup aman karena periode pengulangan maksimal dengan *LCG* adalah m-1, jadi dengan menggunakan 753, maka memungkinkan angka akan kembali dibangkitkan setelah periode tersebut tercapai, dengan demikian akan mempersulit memprediksi kemunculan kunci.
3. Membangkitkan kunci dengan *LCG*.
   Berdasarkan uraian sebelumnya, telah diuraikan ukuran citra adalah 256 x 256, maka banyaknya kunci yang akan dibangkitkan adalah sebanyak 256 x 256. Pembangkitan kunci dilakukan hanya satu kali, karena kunci yang sama akan akan digunakan untuk mengamankan elemen *RGB Red*, *Green* dan *Blue*. Sebelum menerapkan pada sistem, berikut ini akan diuraikan secara manual pembangkitan bilangan acak dengan metode *LCG* dengan membangkitkan sebanyak enam buah bilangan acak dengan variabel yang telah didefensikan di atas.
   a = 3, c =7, z_0 = 5, m=753.
   Persamaan *LCG*: zi=(a***z**$_{(i-1)}$+c) *mod* m
   Kunci *pixel* ke 0 x 0:
   z_1=(a*z_0+c) *mod* 753
   z_1=(3*****5+7) *mod* 753
   z_1= (15+7) *mod* 753
   z_1= 22 *mod* 753
   z_1= 22
   Kunci *pixsel* ke 0 x 1
   z_2=(a*z_1+c) *mod* 753
   z_2=(3*22+7) *mod* 753
   z_2=(66*7) *mod* 753
   z_2=73 *mod* 753z_2=73
   Kunci *pixel* ke 0 x 2
   z_3=(a*z_2+c) *mod* 753
   z_3=(3*73+7) *mod* 753
   z_3= (219 + 7) *mod* 753
   z_3= 226 *mod* 753
   z_3= 226
   Kunci *pixel* ke 0 x 3
   z_4=(a*z_3+c) *mod* 753 z_4= (3*226+7) *mod* 753z_4= (678+7) *mod* 753
   z_4= 685 *mod* 753
   z_4= 685
   Kunci *pixel* ke 0 x 5
   z_5=(a*z_4+c) *mod* 753 z_5= (3*685+7) *mod* 753z_5=(2055+7) *mod* 753
   z_5=2062 *mod* 753
   z_5= 556
   Kunci *pixel* ke 0 x 5
   z_6=(a*z_5+c) *mod* 753
   z_6=(3*556+7) *mod* 753
   z_6=(1668+7) *mod* 753
   z_6=1675 *mod* 753
   z_6= 169

Kunci *pixel* selanjutnya dilakukan dengan cara yang sama sampai seluruh *pixel* memiliki pasangan kunci yaitu 256 x 256 kunci. Berikut ini akan ditampilkan tabel distribusi bilangan acak yang dibangkitkan *LCG* dalam matriks 10 x 10 sebagai sampel kunci, kunci sebenarnya berukuran 256 x 256, bilangan acak ini dibangkitkan dengan menggunakan *matlab* 2013

Tabel 4. kunci yang dibangkitkan

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	22	73	226	685	556	169	514	43	136	415
1	514	43	136	415	499	751	1	10	37	118
2	1	10	37	118	361	337	265	49	154	469

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
3	265	49	154	496	661	484	706	619	358	328
4	706	619	358	328	238	721	664	493	733	700
5	664	493	733	700	601	304	166	505	16	55
6	166	505	16	55	172	523	70	217	658	475
7	70	217	658	475	679	538	115	352	310	184
8	115	352	310	184	559	178	541	124	379	391
9	541	124	379	391	427	535	106	325	229	694

Tabel 4. adalah sampel distribusi kunci berukuran 10 x10 yang dibangkitkan *LC*.Nilai-nilai kunci yang disajikan pada tabel 3.4 di ataslah yang digunakan untuk melakukan proses enkripsi maupun dekripsi masing-masing nilai *pixel plainimage* baik elemen warna *red*, *green* dan *blue*.

4. Proses enkripsi

   Proses enkripsi dilakukan pada semua elemen warna citra secara terpisah dengan menggunakan kunci yang sama. Adapun jumlah piksel citra yang akan dienkripsi sebagai sampel adalah baris pertama dari piksel citra sebanyak 10 piksel (mulai dari pixel 1 hingga 10) dan akan dienkripsi dengan baris pertama dari matriks kunci sebanyak 10 buah kunci. Adapun proses enkripsi akandiurakan sebagai berikut:

   a. Enkripsi elemen warna merah

      Berikut ini adalah nilai piksel elemen warna merah berdasarkan tabel 3.1 nilai piksel merah.

      Pi=33, 35, 35, 35, 35, 35, 35, 35, 35, 35.

      Ki=22, 73, 226, 685, 556, 169, 514, 43, 136, 415

Tabel 5. Distribusi Kunci Terhadap Piksel Merah

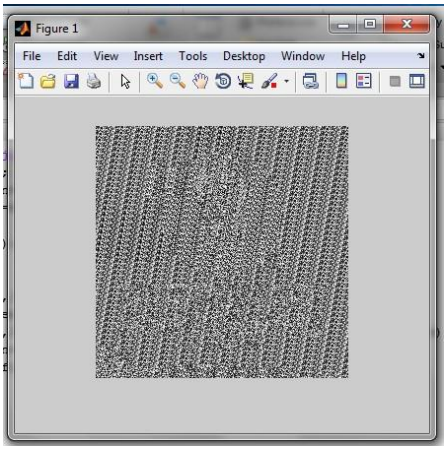
Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Ki	33	35	35	35	35	35	35	35	35	35
Pi	22	73	226	685	556	169	514	43	136	415

Karena semua *plainimage* merah sudah memiliki pasangan kunci, makaproses enkripsi dapat dilakukan, berikut diruaikan proses enkripsinya.

$C_1=(P_1+K_1)$ *mod* 256
$C_1=(33+22)$ *mod* 256
$C_1= 55$ *mod* 256
$C_1= 55$
$C_2=(P_2+K_2)$ *mod* 256
$C_2=(35+73)$ *mod* 256
$C_2= 108$ *mod* 256
$C_2= 108$
$C_3=(P_3+K_3)$ *mod* 256
$C_3=(35+226)$ *mod* 256
$C_3=261$ *mod* 256
$C_3= 5$
$C_4=(P_4+K_4)$ *mod* 256
$C_4=(35+685)$ *mod* 256
$C_4= 720$ *mod* 256
$C_4= 208$
$C_5=(P_5+K_5)$ *mod* 256
$C_5=(35+556)$ *mod* 256
$C_5= 591$ *mod* 256
$C_5= 79$
$C_6=(P_6+K_6)$ *mod* 256
$C_6=(35+169)$ *mod* 256
$C_6= 204$ *mod* 256
$C_6= 204$
$C_7=(P_7+K_7)$ *mod* 256
$C_7=(35+514)$ *mod* 256
$C_7= 549$ *mod* 256
$C_7= 37$
$C_8=(P_8+K_8)$ *mod* 256
$C_8=(35+43)$ *mod* 256
$C_8= 78$ *mod* 256
$C_8= 78$
$C_9=(P_9+K_9)$ *mod* 256

C_9=(35+136) *mod* 256
C_9= 171 *mod* 256
C_9= 171
C_{10}=(P_{10}+K_{10}) *mod* 256
C_{10}=(35+415) *mod* 256
C_{10}= 450 *mod* 256C_{10}=194

Setelah dienkripsi maka diperoleh *cipherimage* dengan nilai piksel yangbaru yaitu Ci= 55, 108, 5, 208, 79,204, 37, 78, 171, 194. Enkripsi nilai piksel yang lain dilakukan dengan cara yang sama. Adapun keadaan citra elemen merah dan nilai piksel setelah dienkripsi dapat dilihat pada gambar berikut.

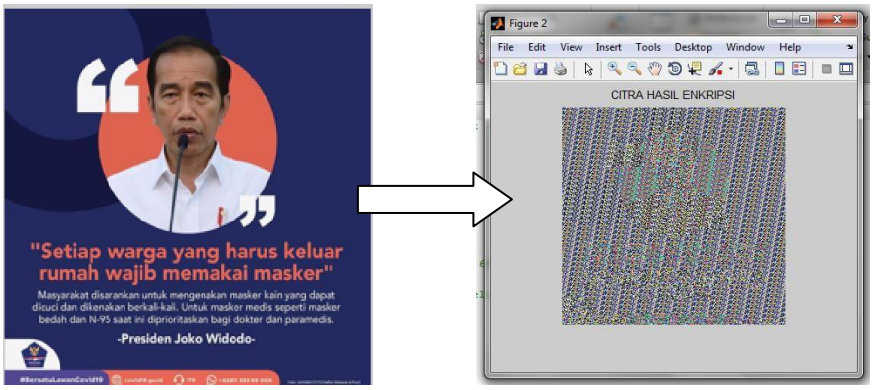


Gambar 5. *CipherImage Red* (merah)

Tabel 6. Piksel *Cipher image* Merah

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	55	108	5	208	79	204	37	78	171	194
1	37	79	171	194	22	18	36	45	72	153
2	36	45	72	153	140	116	44	84	189	248
3	44	84	189	248	184	7	229	142	137	107
4	229	142	137	107	17	244	187	16	0	223
5	187	16	0	223	124	83	201	28	51	90
6	201	28	51	90	207	46	105	252	181	254
7	105	252	181	254	202	61	150	131	89	219
8	150	131	89	219	82	213	64	159	158	170
9	64	159	158	170	206	58	141	104	8	217

b. Penyatuan elemen RGB
Setelah citra selesai dienkripsi berdasarkan warna (RGB) pada setiap elemen masing-masing, maka citra kembali disatukan untuk melihat bentuk citra hasil enkripsi. Perintah *matlab* 2013 untuk menyatukan element RGB . Setelah citra selesai dienkripsi, citra dapat disimpan menjadi *file* baru dalam format .jpg. Berikut ini akan ditampilkan keadaan gambar tersebut sebelum dienkripsi dan bentuk gambar setelah dienkripsi.



Gambar 6. Citra Sebelum Dienkripsi dan Setelah Dienkripsi

5. Dekripsi warna hijau
Adapun nilai piksel dekripsi warna hijau adalahCi= 62, 115, 12, 215, 86, 211, 44, 85, 187, 201
Ki=22, 73, 226, 685, 556, 169, 514, 43, 136, 415.

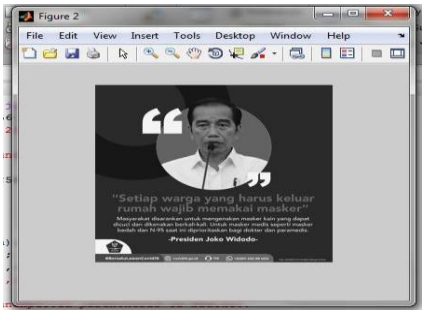
Tabel 7. distribusi kunci terhadap piksel *cipherimage* hijau

index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Ci	62	115	12	215	86	211	44	85	187	201
Ki	22	73	226	685	556	169	514	43	136	415

Karena semua *cipherimage* hijau sudah memiliki pasangan kunci, makaproses dekripsi dapat dilakukan, berikut diruaikan proses dekripsinya

$P_1=(C_1-K_1)$ *mod* 256
$P_1=(62-22)$ *mod* 256
$P_1=40$ *mod* 256
$P_1= 40$
$P_2=(C_2-K_2)$ *mod* 256
$P_2=(115-73)$ *mod* 256
$P_2= 42$ *mod* 256$P_2=42$
$P_3=(C_3-K_3)$ *mod* 256
$P_3=(12-226)$ *mod* 256
$P_3=-214$ *mod* 256
$P_3= 256-(214$ *mod* 256)
$P_3= 256-214$
$P_3= 42$
$P_4=(C_4-K_4)$ *mod* 256
$P_4=(215-685)$ *mod* 256
$P_4= -470$ *mod* 256
$P_4= 256-(470$ *mod* 256)
$P_4= 256-214$
$P_4= 42$
$P_5=(C_5-K_5)$ *mod* 256
$P_5=(86-556)$ *mod* 256
$P_5=-470$ *mod* 256
$P_5=256-(470$ *mod* 256)
$P_5= 256-214$
$P_5= 42$
$P_6=(C_6-K_6)$ *mod* 256
$P_6=(211-169)$ *mod* 256
$P_6=42$ *mod* 256
$P_6= 42$
$P_7=(C_7-K_7)$ *mod* 256
$P_7=(44-514)$ *mod* 256
$P_7=-470$ *mod* 256
$P_7= 256 (470$ *mod* 256)
$P= 256-214P_7=42$
$P_8=(C_8-K_8)$ *mod* 256
$P_8=(85-43)$ *mod* 256
$P_8=42$ *mod* 256
$P_8= 42$
$P_9=(C_9-K_9)$ *mod* 256
$P_9=(178-136)$ *mod* 256
$P_9=42$ *mod* 256
$P_9= 42$
$P_{10}=(C_{10}-K_{10})$ *mod* 256
$P_{10}=(201-415)$ *mod* 256
$P_{10}=-214$ *mod* 256
$P_{10}= 256-(214$ *mod* 256)
$P_{10}= 256-214P_{10}=42$

Setelah didekripsi maka diperoleh *plainimage* warna hijau dengan nilaipiksel yaitu Pi= 40, 42, 42, 42, 42, 42, 42, 42, 42, 42. Untuk piksel yang lain, cara kerjanya sama, adapun keadaan warna hijau dan nilai piksel setelah didekripsi kembali dapat dilihat pada gambar berikut



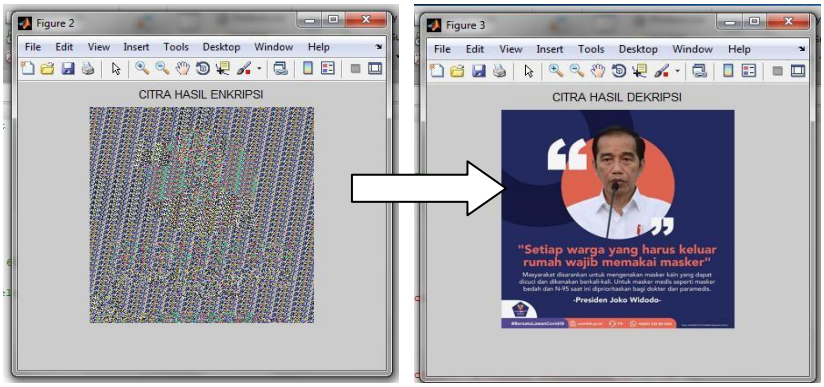
Gambar 7. Citra warna hijau setelah didekripsi

Tabel 8. nilai piksel hijau setelah didekripsi kembali

index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	40	42	42	42	42	42	42	42	42	42
1	42	43	42	42	42	42	42	42	42	42
2	42	42	42	42	42	42	42	42	42	42
3	42	42	42	42	42	42	42	42	42	42
4	42	42	42	42	42	42	42	42	42	42
5	42	42	42	42	42	42	42	42	42	42
6	42	42	42	42	42	42	42	42	42	42
7	42	42	42	42	42	42	42	42	42	42
8	42	42	42	42	42	42	42	42	42	42
9	42	42	42	42	42	42	42	42	42	42

a. Penyatuan elemen *RGB*

Setelah citra selesai didekripsi berdasarkan warna (*RGB*) pada setiap elemenmasing-masing, maka citra kembali disatukan untuk melihat bentuk citra hasil dekripsi. Adapun perintah *matlab* 2013 untuk menyatukan kembali elemen *RGB* yang telah diuraikan menjadi elemen merah, hijau dan biru adalah Adapun bentuk citra sebelum didekripsi dansetelah didekripsi dapar dilihat perbedaannya pada gambar berikut ini



Gambar 8. Keadaan citra sebelum didekripsi dan setelah didekripsi

4. KESIMPULAN

Setelah penelitian dilakukan dan hasil pengujian diperoleh, maka penulis dapat menyimpulkan garis besar dari keseluruhan rangkuman Algoritma *Vigenere Cipher* yang diterapkan pada proses pengamanan gambarbisa mengamankan gambar tetapi masih bisa dipecahkan melalui metode kasiski karena panjang kuncinya masih dapat diketahui. Pembangkit kunci *Random Number Generator* mempunyai tahap pembangkitan kunci dan proses yang sangat cepat karena menggunakan perhitungan yang sederhana. Berdasarkan pengujian yang dilakukan, maka ciphertext yang dihasilkan menggunakan kunci *vigenere cipher* yang dibangkitkan berdasarkan LCG menghasilkan *cipherimage* yang sangat berbeda dengan *plainimage* (nilai kemiripan antara citra sangat rendah), hal ini ditandai dengan nilai PSNR antara kedua citra berada di bawah 30dB, sehingga pembangkitan kunci berdasarkan LCG dapat mengoptimalkan cipher yang dienkripsi berdasarkan *vigenere cipher*.

REFERENCES

[1] Muhammad Dedi Irawan. “Implementasi Kriptografi Vigenere Cipher dengan PHP”, Jurnal Teknologi Informasi. Vol 1,No 1,2017. P-ISSN 2580-7927.

[2] Guruh Marindra Pratama dkk, "Modifikasi Algoritma Vigenere Cipher Menggunakan Metode Catalan Number dan Double Columnar Transposition",Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Adisucipto Yogyakarta. Vol 4, No 1, 2015.
[3] Efrandi dkk. "Aplikasi Kriptografi Pesan Menggunakan Algoritma Vigenere Cipher", Jurnal Media Infotama. Vol 10, No 2,2014. ISSN 1858-2680.
[4] Alfha Vionita dkk, "Penggunaan Metode Enkripsi Vigenere dan MD5 dalam Proses Pengamanan Pesan",Seminar Nasional Matematika dan Pendidikan Matemaika UNY 2016. ISBN. 978-602-73403-1-2.
[5] Fauziah dkk, "Analisis Implementasi Random Number Generator (RNG) pada Simulasi Antrian Menggunakan Aplikasi Berbasis.Net Framework",Seminar Nasional Informatika 2012. ISSN: 1979-2328.
[6] Darwis Robinson Manalu. "Penyandian dan Pengamanan Pesan Teks dengan Algoritma Misty", Majalah Ilmiah Methoda. Vol 4,No 1,2014.
[7] Janner Simarmata dkk, Kriptografi Teknik Keamanan Data dan Informasi, Yogyakarta, CV ANDI OFFSET, 2019.
[8] Arimaz Hangga dkk. "Modifikasi Linear Congruential Generator untuk Sistem Pengacakan Soal Pada Computer Based Test (CBT)", Jurnal Teknik Elektro. Vol 8.No 2. ISSN 1411-0059.