Analisis Efek Chaos pada Enkripsi Dekripsi Image dengan Metode One Time Pad

Emmy Apulina Br. Bangun[#1], *Irene Adisty Putri*[#3]

[#]*Departemen Teknik Informatika*

Fakultas Teknik, Institut Teknologi Harapan Bangsa JL. Dipatiukur No. 80 – 82, Bandung 40122

[1]emmy@ithb.ac.id

Abstrak— *One-Time Pad* merupakan salah satu jenis teknik kriptografi yang menggunakan metode substitusi dengan cara memberikan syarat-syarat khusus terhadap kunci yang digunakan yaitu terbuat dari nilai yang acak (kunci acak atau pad). Kunci-kunci yang digunakan untuk enkripsi dapat di generate dengan berdasar pada prinsip Chaos. Teori Chaos mempelajari perilaku sistem dinamik yang sangat sensitif terhadap kondisi awal. Kepekaan atau sensitifitas ini secara populer disebut efek kupu-kupu. Teori Chaos terbukti pada proses enkripsi, karena dengan nilai kunci yang hanya berbeda sedikit dari kunci awal, hasil gambar yang diperoleh menjadi tidak sama dengan gambar awal. Dengan fungsi pengambilan sisa kunci 24 bit atau disebut *KeyCutterBack* didapatkan *cipher* image yang lebih baik dibandingkan dengan hasil dari fungsi *Teori Chaos, One-Time Pad,efek kupu-kupu , KeyCutterFront, KeyCutterBack* yang hanya mengambil bit depan saja.

Kata kunci— *Teori Chaos, One-Time Pad,efek kupu-kupu , KeyCutterFront, KeyCutterBack*

Abstract— One-time pad is one of the cryptographic techniques using substitution method by providing special conditions to the key which is made of the value of a random (or random key pad). The keys whose are used in encryption can be generate from the other chance with Chaos teory. Chaos Theory studies the behavior of dynamical systems are very sensitive to initial conditions. This sensitivity as it is popularly called the butterfly effect. The effectiveness of the chaos theory is proven in the encryption process, because if the key value differs only slightly from the initial key, the results of obtained images will not be equal to the plain images. The function of taking the remaining 24 bits or called KeyCutterBack produced a better cipher image compared with the results of functions that only take the 24 bit starting from the front or called KeyCutterFront.

Keywords— Chaos Theory, One-Time Pad, butterfly effect, KeyCutterFront, KeyCutterBack

I. PENDAHULUAN

Pada saat ini hampir disetiap aspek kehidupan manusia melakukan pertukaran data. Proses pengiriman data tidak bisa lepas dari metode enkripsi dan dekripsi data untuk menjamin kerahasiaan data. Enkripsi dilakukan oleh pengirim dengan cara mengubah data asli menjadi suatu data lain dengan suatu kunci tertentu yang tidak bisa dipecahkan oleh orang lain.

Umumnya enkripsi dilakukan pada data teks dan jarang dilakukan pada data image. Pada prinsipnya metode yang digunakan untuk mengenkripsi teks dapat diterapkan untuk mengenkripsi image. Salah satunya adalah One-Time Pad. Metode ini merupakan salah satu metode pengenkripsian simetris, dimana kunci yang digunakan untuk dekripsi sama dengan kunci yang digunakan pada enkripsi. Jumlah kunci pada One-Time Pad sesuai dengan jumlah data. Kunci awal digunakan untuk membangkitkan kunci-kunci berikutnya melalui *key generator*. Setelah proses enkripsi, kunci-kunci tersebut dihancurkan. Sehingga data yang dikirim hanya data hasil enkripsi dan kunci awal.

Jurnal ini berfokus pada kunci yang dihasilkan dengan menggunakan teori Chaos dan efek kunci tersebut terhadap hasil enkripsi dengan metode One-Time Pad.

II. ENKRIPSI DEKRIPSI IMAGE

A. Citra

Sebuah gambar digital didefinisikan oleh array yang berisi *pixel-pixel* dan setiap *pixel* masing-masing memiliki nilai yang tertentu. Jika kita memiliki citra 512 × 512 *pixel*, itu berarti bahwa data untuk gambar berisi informasi tentang 262.144 *pixel*. Jumlah warna (*Color Space*) yang dapat diberikan untuk setiap *pixel* disebut sebagai *color depth* atau *bits resolution*.

Warna *image* dalam dunia komputasi dapat direpresentasikan ke dalam tiga buah warna yaitu warna merah, hijau dan biru. Warna–warna tersebut secara umum dapat membentuk warna lain dengan memberikan nilai yang berbeda pada ketiga warna tersebut. Warna-warna itu dinamakan warna-warna primer.

Citra *bitmap true color* (24 bit) mempunyai karakteristik tersebut di atas. Citra ini menggunakan kombinasi nilai *pixel* yang menunjukkan warna gambar tersebut.

Dapat dikatakan citra *bitmap true color (24 bit)* tersusun dari kombinasi ketiga warna primernya. Untuk setiap warna primer mempunyai beberapa nilai *pixel* dari 0 sampai dengan 255 (1 byte) yang menyatakan warnanya. Dengan demikian jumlah variasi warna *image* tersebut adalah 16.777.216 buah.

TABLE 1 TABEL COLOR DEPTH DAN WARNA

nama warna	jumlah bit	jumlah warna
hitam dan putih	1	2
grey scale	2	256
256 color	8	256
high color	16	66
true color	24	16.777.216
true color	32	4.294.967.296
true color	36	68.719.476.736

B. *Teori Chaos*

Chaos Theory adalah studi di bidang matematika, fisika, ekonomi dan filsafat yang mempelajari perilaku sistem dinamik yang sangat sensitif terhadap kondisi awal. Kepekaan atau sensitifitas ini secara populer disebut sebagai efek kupu-kupu atau *the butterfly effect*. Perbedaan kecil dalam kondisi awal (seperti yang disebabkan oleh kesalahan pembulatan dalam perhitungan numerik) banyak menghasilkan hasil divergen untuk sistem chaos.

Input Chaos pada setiap proses *generate* berbeda-beda. Input merupakan hasil dari fungsi sebelumnya, sehingga ketika ada perbedaan input awal yang diberikan maka hasilnya akan berbeda (adanya keterkaitan nilai x secara kontinu). Perbedaan ini akan sangat signifikan karena jumlah iterasi untuk tiap *keystream* berbeda-beda tergantung pada nilai x sebelumnya.

Rumus Chaos yang digunakan :

*PTD = R *PTI * (1 - PTI)*

Dimana :

PTD = Populasi Tahun Depan

PTI = Populasi tahun ini

C. *One Time Pad*

One-time pad adalah suatu sistem di mana suatu kunci rahasia yang digunakan hanya sekali untuk mengenkripsi pesan yang kemudian didekripsi lagi dengan kunci yang sama. Panjang kunci yang digunakan pada metode ini harus sama dengan panjang plainteks.

Enkripsi pada pesan berupa karakter dapat dinyatakan sebagai penjumlahan modulo 26 dari satu karakter plainteks dengan satu karakter kunci OTP [KOB94] :

$c_i = (p_i + k_i) \bmod 26$

Yang dalam hal ini, p_i adalah plainteks ke-i, dan c_i adalah huruf cipherteks ke-i.

Angka 26 muncul karena sistemnya menggunakan abjad. Artinya hanya abjad A – Z saja yang dapat dikodekan dengan sistem seperti ini. Berdasarkan rumus tersebut terlihat tidak ada kebutuhan mengulang penggunaan kunci selama proses enkripsi. Setelah pengirim mengenkripsikan pesan dengan kunci, ia menghancurkan kunci tersebut.

OTP ini dapat diperluas dengan penggunaan sistem bilangan biner. Semua tipe data dapat dianggap sebagai data biner. Dan karena bilangan biner hanya mengenal 0 dan 1, maka basis 26 diubah menjadi basis 2 [KUR04]. Penjumlahan modulo 2 ini dinyatakan dengan XOR. Dan inilah yang sering digunakan dalam sistem digital sekarang ini. Cipherteks diperoleh dengan melakukan penjumlahan modulo 2 satu bit plainteks dengan satu bit kunci:

$c_i = (p_i + k_i) \bmod 2$

yang dalam hal ini, p_i : bit plainteks, k_i : bit kunci, c_i : bit cipherteks. Plainteks diperoleh dengan melakukan penjumlahan modulo 2 satu bit cipherteks dengan satu bit kunci:

$p_i = (c_i + k_i) \bmod 2$

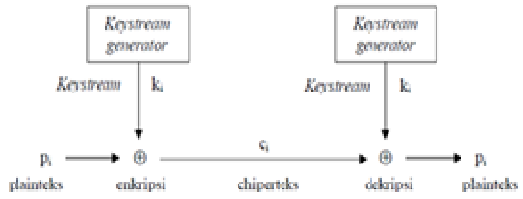
Mengingat operasi penjumlahan modulo 2 identik dengan operasi bit dengan operator XOR, maka persamaan enkripsi dapat ditulis sebagai:

$c_i = p_i \otimes k_i$

dan proses dekripsi menggunakan persamaan:

$p_i = c_i \otimes k_i$

Skema proses enkripsi dan dekripsi dapat dilihat pada gambar 1.



Gambar 1 One-Time Pad

III. ANALISA DAN PERANCANGAN

A. *Analisa Pembuatan Kunci*

Pertimbangan mengapa prinsip Chaos digunakan dalam generate kunci, yaitu:

1. *Sensitive Dependence* dalam *Chaotic Algorithm.*
2. Tingkat random kunci yang dihasilkan dari teknik *Chaos.*

Barisan nilai *chaos* yang digunakan sebagai aliran kunci adalah bilangan real antara 0 dan 1. Dari nilai kunci awal , dapat dihitung nilai dari kunci-kunci selanjutnya dengan berdasar pada rumus chaos (persamaan logistik).

Contoh perhitungan dengan nilai key yang berbeda :

x1 = 0.00230872

x2 = 0.002308716

Selisih nilai = 0.000000004

Nilai r = 4

Dengan menggunakan persamaan logistic dan pengulangan sebanyak 20 kali.

f(x1) = |4 x 0.00230872 x (1 - 0.00230872) |

= 0.817174220213535 ~ 81

f(x2) = |4 x 0.002308716 x (1 – 0.002308716)|

= 0.84970023329208 ~ 84

B. *Analisa Fungsi Pemotongan*

Agar barisan nilai *chaos* dapat dipakai enkripsi dan dekripsi, maka nilai *chaos* dikonversi ke nilai integer.

Secara matematis, nilai *chaos x* dikonversi ke *integer* dengan menggunakan persamaan berikut:

$T(x, size) = ||x * 10^{count}|| , x \neq 0$

Dalam hal ini *count* mulai dari 1 dan bertambah 1 sampai $x * 10^{count} > 10^{size-1}$. Hasilnya kemudian diambil bagian *integer* saja (dilambangkan dengan pasangan garis ganda pada persamaan diatas).

Berikut contoh pengubahan key dalam real ke integer :

x = 0.328,

maka *size* = 3,

$T(0.328,4) = ||0.328*10^{count}||, x \neq 0.$

Setelah nilai $T > 10^2$ pengulangan akan berhenti. Dan nilai yang dihasilkan 328.

Hasil dari perhitungan kunci yang telah menjadi integer diubah kedalam bentuk nilai biner. Bit-bit tersebut dapat dikenai fungsi pemotongan bit. Karena disini yang digunakan citra *bitmap true color* 24 bit, maka jumlah bit yang dipotong adalah 24 bit.

Ada 2 fungsi pemotongan bit yang digunakan, yaitu pemotongan didepan *(KeyCutterFront)* dan pemotongan dibelakang *(KeyCutterFront)*.

1) KeyCutterFront

```
Function KeyCutterBack (input_key) -> integer[]
 temp ← key.length
    if (temp <= 24) then
      for (i = 0 to 24) do
       output[i] ← input_key.substring(i,1)
      end for
    end if
    if (temp > 24) then
      for (i = 0 to 24) do
       output[i] ← input_key.substring(temp-24,1)
       temp ← temp + 1
      end for
    end if
return (output)
```

Contoh 1 penerapan keyCutterFront :

Nilai kunci = 328

Biner =

000000000000000101001000

Output = 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,1,0,0,0

Contoh 2 penerapan keyCutterFront :

Nilai kunci = 123456789

Biner =

111010110111100110100010101

Karena panjang kunci biner adalah 27 atau lebih dari 24 maka nilai output yang diambil adalah

111010110111100110100010101

Output = 1,1,1,0,1,0,1,1,,0,1,1,1,1,0,0,1,1,0,1,0,0,0,1,0

2) KeyCutterBack

```
Function KeyCutterBack (input_key) -> integer[]
 temp ← key.length
    if (temp <= 24) then
      for (i = 0 to 24) do
       output[i] ← input_key.substring(i,1)
      end for
    end if
    if (temp > 24) then
      for (i = 0 to 24) do
       output[i] ← input_key.substring(temp-24,1)
       temp ← temp + 1
      end for
    end if
return (output)
```

Penerapan keyCutterBack dari contoh nilai kunci sebelumnya :

Nilai Kunci = 328

Output =

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,1,0,0,0

Nilai Kunci =123456789

Biner =

111 010110111100110100010101

Output =

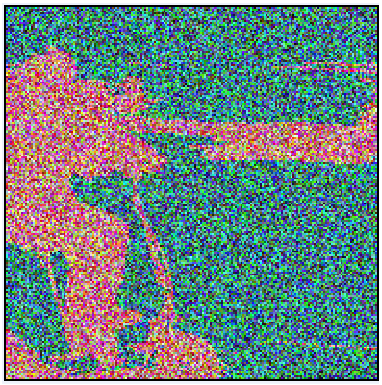
0,1,0,1,1,0,1,1,1,1,0,0,1,1,0,1,0,0,0,1,0,1,0,1

IV. PENGUJIAN DAN HASIL UJI

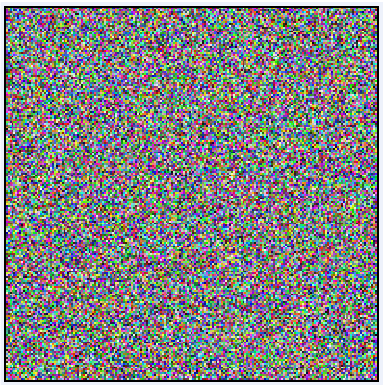
Berikut hasil dari sebuah pengujian terhadap prototype :

Pengujian 1 :

Gambar 2 Gambar Awal Pengujian 1

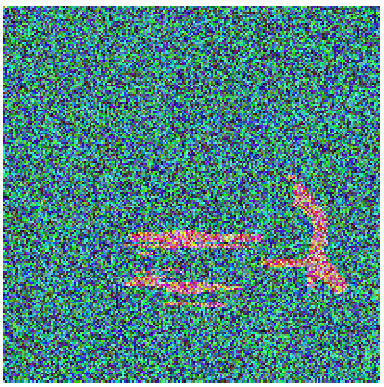


Gambar 3 Hasil Enkripsi Pengujian 1 dengan KeyCutterFront

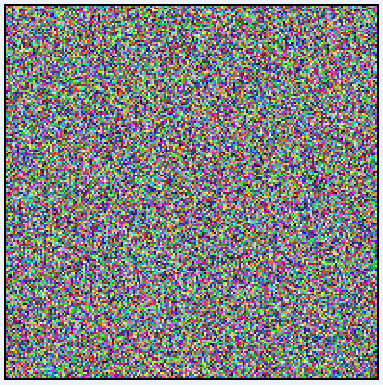


Gambar 4 Hasil Enkripsi Pengujian 1 dengan KeyCutterBack

Gambar 5 Gambar Awal Pengujian 2



Gambar 6 Gambar Hasil Enkripsi Pengujian 2 KeyCutterFront



Gambar 7 Hasil Enkripsi Pengujian 2 KeyCutterBack

V. KESIMPULAN

1. Teori chaos terbukti pada proses enkripsi, karena dengan nilai kunci yang hanya berbeda sedikit dari kunci awal, hasil gambar yang didapat menjadi tidak sama dengan gambar awal apabila menggunakan pemotongan 24 bit key dari belakang.
2. Sensitifitas dari masing-masing kunci terhadap hasil enkripsi terdapat pada 24 bit dari belakang yang key-key yang dihasilkan.
3. Kombinasi algoritma one time pad dan fungsi chaos mampu menyelesaikan masalah enkripsi dan dekripsi dengan baik dengan melakukan pemotongan 24 bit key dari belakang .

REFERENSI

[1] Adisty Putri, Iren, Enkripsi dan Dekripsi dengan One Time Pad dan Efek Chaos pada Citra Bitmap, 2010, Departemen Teknik Informatika Institut Teknologi Harapan Bangsa, Bandung.

[2] Munir, Rinaldi, Kriptografi, 2006, Penerbit Informatika, Bandung

[3] Mao Hewlett, Wenbo, Modern Cryptography : Teori and Practice, Prentice Hall PTR, 2003