File To Image Encryption (FTIE) Menggunakan Algoritma Randomized Text Dan Arnold Cat Map (ACM) Untuk Keamanan Transmisi Data Digital

Ady Suprianto[1], Yudi Prayudi[2], Bambang Sugiantoro[3]
Magister Teknik Informatika
Universitas Islam Indonesia (UII)
Yogyakarta, Indonesia
[1]adysuprianto@yahoo.co.id, [2]prayudi@uii.ac.id, [3]bambang.sugiantoro@uin-suka.ac.id

***Abstrak*— Pengiriman melalui jaringan publik sangat rentan terhadap pengaksesan atau pencurian informasi ditengah-tengah proses transmisi. Hal ini berdampak kerugian bagi pihak yang memiliki informasi tersebut. Salah satu cara untuk mengamankan informasi ketika proses transmisi dilakukan yaitu dengan melakukan enkripsi terhadap informasi tersebut. *File To Image Encryption* (FTIE) adalah teknik enkripsi yang melakukan enkripsi dari *file* menjadi sebuah gambar. Teknik FTIE merupakan teknik enkripsi yang dikembangkan dari teknik *Text To Image Encription* (TTIE). Penelitian ini dilakukan dengan menggabungkan dua algoritma yaitu algoritma *Randomized Text* dan algoritma *Arnold Cat Map* (ACM). Hal ini bertujuan untuk menghasilkan *chipertext* yang memiliki tingkat keamanan lebih kuat karena nilai dan posisi *byte* yang acak. Hasil penelitian menunjukkan bahwa teknik FTIE menggunakan algoritma *Randomized Text* dan ACM memiliki ketahanan yang kuat dari berbagai macam analisis dan memiliki integritas yang dapat dipertanggungjawabkan.**

Kata kunci— FTIE; Randomized Text; ACM

I. PENDAHULUAN

Peningkatan jumlah pengguna internet setiap tahun menyebabkan tingginya angka kejahatan yang terjadi di dunia *cyber*. Salah satu kejahatan yang terjadi di dunia *cyber* adalah pencurian data. Menurut data yang di *update* oleh *breachlevelindex.com* jumlah pencurian data yang terjadi pada tahun 2016 mencapai angka 1.3 milyar. Hal ini terjadi karena adanya celah tidak aman ketika data diproses dan ditansmisikan.

Menurut Ariyus [1], ada 4 ancaman keamanan terhadap sebuah informasi ketika proses transmisi dilakukan yaitu *interruption, interception, modification,* dan *fabrication.* Untuk mengantisipasi ancaman-ancaman tersebut, banyak sekali teknik yang bisa digunakan untuk menjamin keamanan data ketika proses transmisi dilakukan. Salah satu teknik yang paling sering digunakan adalah kriptografi, yaitu dengan mengenkripsi data kedalam bentuk yang tidak bisa dibaca dan dimengerti oleh pihak yang tidak berhak. Kriptografi dapat diartikan sebagai proses untuk melindungi data dalam arti yang luas [2] .

Menurut Kromodimoeljo [3], ada beberapa kondisi dimana kriptografi dibutuhkan yaitu, informasi yang bersifat sensitif, mencegah penyadapan, dan mencegah penyamaran. Kriptografi memberikan keamanan informasi berupa kerahasiaan (*confidentiality*), integritas data (*data integrity*), otentikasi (*authentication*), dan nirpenyangkalan (*nonrepudiation*) [4].

*Text To Image Encription (*TTIE) adalah teknik enkripsi yang mentranformasi sebuah teks menjadi sebuah gambar. Namun, teknik ini masih perlu pengembangan karena masih adanya beberapa kekurangan. Kekurangan tersebut dapat terjadi saat proses pertukaran kunci ketika transmisi dilakukan, dan terjadi akibat terlalu besarnya ukuran kunci karena setiap karakter dari *plaintext* digantikan oleh tiga angka acak sehingga memerlukan setidaknya 3 kali lipat data hanya untuk mendekripsi sebuah paket [5].

Adanya beberapa kelemahan pada teknik TTIE mendorong peneliti untuk melakukan pengembangan dengan membangun sebuah teknik *File To Image Encription* (FTIE). FTIE adalah teknik enkripsi yang melakukan enkripsi dari *file* menjadi sebuah sebuah gambar. Teknik ini lebih efektif terhadap banyak jenis dan informasi dibandingkan TTIE yang hanya melakukan ekripsi terhadap teks.

Untuk memperoleh *chipertext* dengan tingkat keamanan yang lebih tinggi, perlu dilakukan kombinasi antara dua algoritma. Berdasarkan hal tersebut, penelitian ini bertujuan untuk membangun teknik FTIE dengan mengkombinasikan dua algoritma, yaitu algoritma *Randomized Text* dan ACM. Selanjutnya akan dilakukan pengujian pada teknik FTIE terhadap analisis entropi, analisis *differensial*, analisis *bruteforce*, analisis waktu enkripsi dan dekripsi, dan uji integritas. Setelah dilakukan pengujian, teknik FTIE selanjutnya akan diimplementasikan pada sebuah aplikasi.

Pemilihan algoritma *Randomized Text* dikarenakan algoritma *Randomized Text* merupakan salah satu dari *randomized encryption*, yang artinya setiap kali proses enkripsi dilakukan, akan selalu menghasilkan *chipertext* yang berbeda-beda [6] dan juga dikarenakan *Randomized encryption* bisa memberikan tingkat keamanan yang lebih tinggi dengan peningkatan nilai *entropy* [7].

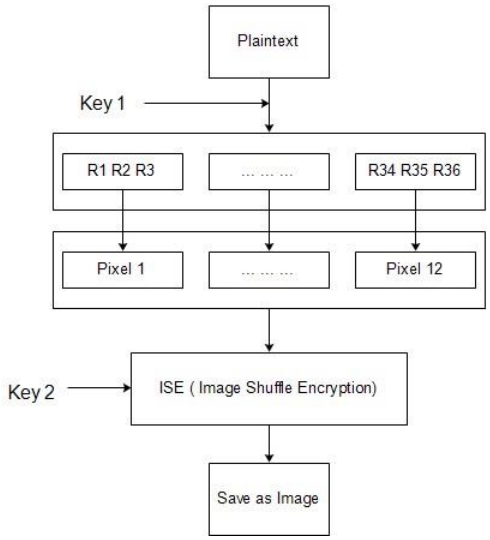
Sedangkan pemilihan algoritma ACM karena ACM adalah algoritma yang memiliki sifat transformasi yang sederhana namun kuat [8]. Irfan dan Prayudi [9] melakukan penelitian dengan menggabungkan antara algoritma *Chaos* yaitu *Logistic Map* dan ACM, dengan algoritma populer *RSA* untuk memberikan keamanan ganda pada sebuah gambar. Hasil penelitian menunjukkan bahwa penggabungan antara algoritma *Chaos* dengan *RSA* terbukti kuat terhadap berbagai analisis serangan baik serangan pada *chipper image* maupun terhadap *chipertext*.

Kombinasi antara algoritma *Randomized Text* dan ACM pada sebuah teknik FTIE diharapkan mampu menghasilkan *chipertext* yang memiliki tingkat keamanan lebih kuat karena memiliki nilai dan posisi *byte* yang acak.

II. STUDI PUSTAKA

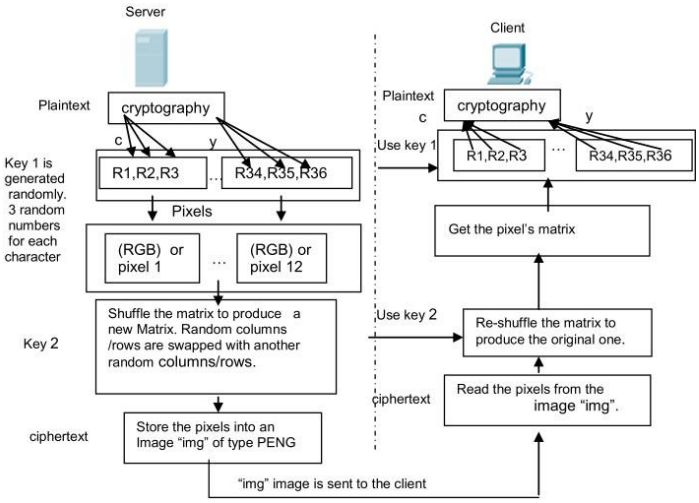
A. *Text To Image Encryption (TTIE)*

Ahmad Abusukhon [10] pertama kali menemukan teknik enkripsi yang disebut dengan *Text To Image Encryption* (TTIE). Teknik ini merupakan teknik yang mengubah sebuah *plaintext* menjadi sebuah gambar, yang terdiri dari 2 tahap yaitu tahap TTIE itu sendiri dan tahap *Image Shuffle Encryption* (ISE). Pada tahap TTIE, teks biasa akan ditranformasikan (dienkripsi) ke dalam sebuah gambar. Pada tahap ini masing-masing karakter dari *plaintext* disimpan kedalam sebuah *array*. Satu karakter dari *array* ini akan mewakili 1 pixel dari gambar, dari satu pixel gambar ada 3 bilangan bulat dengan rentang nilai 0-255, dimana dalam salah satu bilangan tersebut terdapat 1 bilangan yang merupakan karakter dari *plaintext.* Hal tersebut akan mempersulit seorang kriptanalis untuk menentukan mana yang merupakan karakter asli dari *plaintext.* Pada tahap ISE akan dilakukan pengacakan posisi pada *array* yang dihasilkan di tahap TTIE yang meliputi pengacakan baris dan pengacakan pada kolom. Hal tersebut akan mempersulit seorang kriptanalis untuk menentukan posisi asli dari sebuah *pixel*. Konsep dari TTIE ditunjukkan pada gambar dibawah ini.



Gambar 1. Konsep Text to image encryption (TTIE)

Skema transmisi data dari server ke klien menggunakan TTIE ditunjukkan oleh gambar berikut ini.



Gambar 2. Skema transmisi Text to image encryption (TTIE)

B. *Randomized Text*

Algoritma Randomized Text merupakan salah satu algoritma yang *chipertext*-nya bersifat dinamis yang artinya *chipertext* yang dihasilkan selalu berubah-ubah walaupun dari plaintext yang sama dengan kunci yang sama pula. Randomized Text memiliki persamaan enkripsi yang cukup sederhana yaitu:

$$C1 = K + 2P + R$$
$$C2 = 2K + P + R \quad (1)$$

Randomized Text melakukan enkripsi per-karakter dengan persamaan diatas, dimana setiap satu karakter dari *plaintext* akan menghasilkan 2 karakter *chipertext*. Sehingga tiap kali melakukan enkripsi, *chipertext* yang dihasilkan pasti akan menghasilkan dua kali lipat dari ukuran *plaintext*. Sedangkan persamaan dekripsinya adalah:

$$P = (C1-K) – (C2-2K) \quad (2)$$

Pada saat proses dekripsi ukuran *plaintext* akan menjadi setengah dari ukuran *chipertext*.

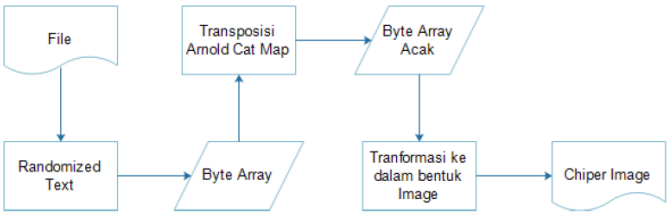
C. *Arnold Cat Map (ACM)*

Arnold Cat Map (ACM) ditemukan oleh Vladimir Arnold pada tahun 1960. Ketika melakukan penelitan, Arnold menggunakan sebuah gambar kucing dalam melakukan percobaan, sehingga algoritma hasil penelitian yang dilakukan dinamakan dengan *Arnold Cat Map (ACM*) [11]. ACM adalah fungsi chaos dwimatra yang mentransformasikan koordinat (x,y) di dalam citra ke koordinat baru di dalam citra yang berukuran N x N ke koordinat baru (x', y') menggunakan persamaan sebagai berikut:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} mod(N) \quad (3)$$

III. METODOLOGI

Skema *File To image encryption* (FTIE) dengan menggunakan algoritma *Randomized Text* dan ACM dapat dilihat pada gambar berikut:

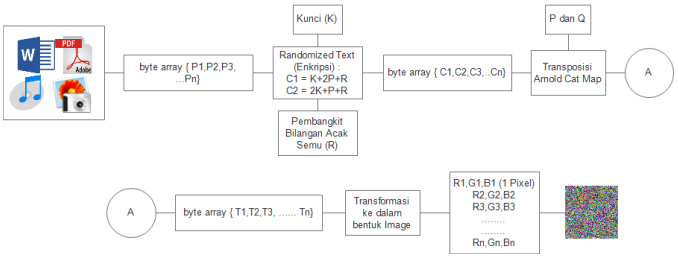


Gambar 3. Skema tahapan FTIE dengan menggunakan algoritma Randomized Text dan ACM

Gambar 3 di atas memperlihatkan tahapan-tahapan enkripsi dari sebuah file menjadi sebuah gambar atau image. Pertama, sebuah file akan dienkripsi dengan menggunakan algoritma Randomized Text. Kedua, byte array yang dihasilkan dari tahap pertama akan ditransformasi menggunakan algoritma ACM. Ketiga, proses transformasi dengan menggunakan algoritma ACM akan menghasilkan byte array yang sudah teracak posisinya. Selanjutnya, byte array yang sudah teracak tersebut akan ditransformasi ke dalam bentuk gambar, dan pada akhirnya output yang dihasilkan berupa sebuah gambar. Selanjutnya, penjelasan untuk masing-masing tahapan dapat dilihat pada Gambar 4 dan Gambar 5.

A. *Rancangan Model Enkripsi*

Rancangan model enkripsi dapat dilihat pada gambar berikut:



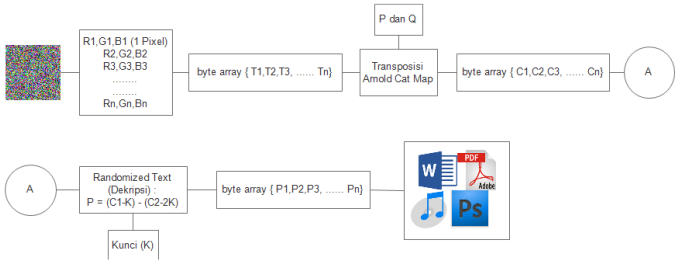
Gambar 4. Skema Enkripsi FTIE

Prosedur enkripsi teknik FTIE pada dokumen digital dijelaskan sebagai berikut:

1) Menentukan dokumen digital yang akan di enkripsi.
2) Mengambil nilai byte dari dokumen digital tersebut kemudian dimasukkan kedalam byte array (P1…..Pn).
3) Menentukan nilai kunci (K)
4) Mengambil nilai dari pembangkit bilangan acak
5) Enkripsi masing-masing nilai byte yang ada pada byte array menggunakan persamaan enkripsi Randomized Text kemudian hasilnya menjadi byte array baru (C1…..Cn)
6) Menentukan nilai P dan Q
7) Melakukan pengacakan posisi menggunakan algoritma ACM 1-D sesuai dengan nilai P dan Q kemudian akan menghasilkan byte array baru (T1…..Tn).
8) Mengelompokkan byte array (T1…..Tn) menjadi nilai R,G,B yang masing-masing R,G,B mewalikili 1 pixel warna.
9) Gabung pixel-pixel menjadi satu kesatuan gambar.

B. *Rancangan Model Dekripsi*

Rancangan model dekripsi dapat dilihat digambarkan dalam gambar berikut ini:



Gambar 5. Skema Dekripsi FTIE

Prosedur dekripsi teknik FTIE pada dokumen digital dijelaskan sebagai berikut:

1) Menentukan gambar yang akan di dekripsi
2) Mengambil nilai *pixel*, dimana satu *pixel* akan memiliki nilai R,G,B. kemudian digabung menjadi *byte array* (T1…..Tn)
3) Menetukan nilai P dan Q
4) Melakukan pengembalian posisi *byte* ke kondisi awal menggunakan persamaan ACM 1-D sesuai dengan nilai P dan Q kemudian akan menghasilkan *byte array* baru (C1…..Cn)
5) Menentukan nilai kunci (K)
6) Dekripsi masing-masing nilai *byte* yang ada pada *byte array* menggunakan persamaan dekripsi *Randomized Text* kemudian hasilnya menjadi *byte array* baru (P1…..Pn)
7) Menggabung *byte array* menjadi dokumen digital.

C. *Analisis dan Pengujian*

Analisis dan pengujian dilakukan untuk mengetahui tingkat kekuatan dan efektifitas algoritma. Adapun beberapa factor yang dinilai adalah sebagai berikut:

1) *Analisis Diferensial*

Analisis diferensial dalam penelitian ini bertujuan untuk menunjukkan perbedaan yang terdapat antara *chiperimage* yang satu dengan yang lain. Ada 2 jenis analisis yang akan diterapkan yaitu analisis Histogram dana analisis *Number of Pixel Change Rate* (NPCR). Analisis histogram bertujuan untuk membandingkan pola antara gambar dari hasil yang satu dengan hasil yang lain, sehingga didapatkan karakter dari histogram yang dienkripsi menggunakan teknik FTIE.

Pengujian NPCR dilakukan untuk menjamin bahwa pada setiap titik matriks terdapat perubahan elemen warna [12].

2) Analisis Entropy

Analisis *entropy* bertujuan untuk menganalisis tingkat keacakan sebuah informasi yang sudah terenkripsi, Nilai entropi ideal jika sebuah informasi dienkripsi dan dalam kondisi teracak adalah 7,99902 (~8). Jika nilai entropi lebih kecil dari 8, dapat dikatakan sistem enkripsi masih bisa ditebak [13].

3) Analisis lama waktu eksekusi

Analisis ini dilakukan untuk mengetahui performa dari algoritma yang digunakan. Pada penelitian ini akan dilakukan analisis lama waktu eksekusi yang dibutuhkan dalam melakukan enkripsi dan dekripsi berdasarkan ukuran dari sebuah *file* dan akan ditampilkan dalam sebuah tabel dan grafik.

4) Analisis exhaustive attack / brute force attack

Serangan Bruteforce adalah serangan yang dilakukan dengan cara mencoba setiap nilai kunci yang mungkin digunakan untuk melakukan dekripsi pada *chipertext*, serta melihat hasil yang didapatkan jika melakukan dekripsi dengan menggunkan kunci yang salah. Pada tahapan ini akan dilakukan analisis perhitungan estimasi waktu yang dibutuhkan untuk memecahkan *chipertext* yang dihasilkan dari proses enkripsi menggunakan teknik FTIE.

5) Uji integritas data

Pengujian integritas bertujuan untuk membandingkan kecocokan antara *file* asli dengan *file* hasil dekripsi. Jika terdapat ketidakcocokan antara *file* asli dengan *file* hasil dekripsi, maka integritas data tidak bisa diterima, artinya ada kemungkinan bahwa data telah dimanipulasi atau terjadi kerusakan pada saat di *file* tersebut ditransmisikan.

IV. HASIL DAN PEMBAHASAN

Percobaan pada penelitian ini menggunakan perangkat Visual Studio 2012, processor Intel(R) Core(TM) i5-4200M CPU @2.50 GHz, RAM 6 GB, Harddisk 500 GB, Sistem Operasi Windows 10 Pro, Visual Studio 2012 dan Bahasa pemrograman C#. Adapun *file* yang dipakai untuk melakukan ujiacoba terdapat pada tabel dibawah ini:

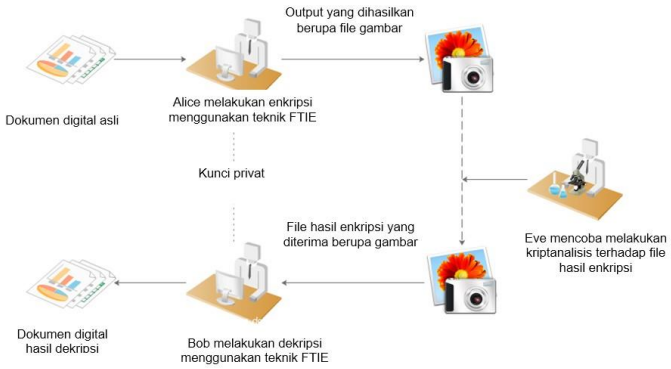
TABEL I. FILE UJI COBA DAN ANALISIS

Nama *File*	Ukuran (*bytes*)	Tipe *File*
CV.docx	13907	Word Document
Data Analisis.xlsx	15074	Excel Worksheet
Intro Template.mp4	328281	MP4 *File*
Lampiran_2016725.zip	23168	ZIP archive
MITK_Kelompok.pptx	66165	PowerPoint Presentation
MTIF Bab 2.pdf	71705	PDF Document
rainbows3.jpg	966193	JPG *File*
Surat 001.mp3	966355	MP3 Format Sound

Dalam penelitian ini, tahapan dimulai dengan pembuatan aplikasi untuk proses enkripsi dengan menggabungkan antara algoritma *Randomized Text* dan ACM. Langkah selanjutnya yaitu percobaan melakukan enkripsi dan dekripsi pada *file*, kemudian langkah yang terakhir yaitu dengan melakukan analisis keamanan dari *ciphertext* hasil enkripsi algoritma yang sudah diterapkan.

A. Skenario Transmisi FTIE

Skenario pengiriman dokumen digital yang sudah terenkripsi dengan teknik FTIE menggunakan algoritma *Randomized Text* dan ACM ditunjukkan oleh gambar berikut ini:



Gambar 6. Skenario Transmisi FTIE

Pada Gambar 6 diberikan sebuah gambaran alice ingin mengirimkan citra yang memiliki informasi penting kepada rekan kerjanya bob melalui jaringan publik yang tentunya rawan terhadap penyadapan atau pengaksesan oleh pihak lain. Untuk menjaga kerahasiaan dan keamanan pada dokumen digital yang dikirim maka alice sebelumnya melakukan enkripsi menggunakan teknik FTIE sehingga dokumen digital yang ingin dikirimkan menjadi sebuah gambar dengan format PNG. Diasumsikan eve mendapatkan gambar yang alice kirimkan kepada bob, yang eve lakukan adalah mencoba melakukan analisis-analisis kriptanalis untuk memecahkan informasi yang terkandung pada gambar, eve tidak mengetahui apakah gambar tersebut merupakan hasil dari enkripsi sebuah gambar ataukah cuma sekedar gambar biasa yang tidak memiiliki arti ataukah ada gambar tersebut hasil dari enkripsi *file* yang lain. Apabila eve mengira bahwa gambar tersebut merupakan hasil enkripsi dari sebuah gambar, maka eve tidak akan mungkin memecahkan informasi dari gambar tersebut.

B. Analisis Diferensial

1) Analisis Histogram

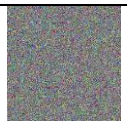
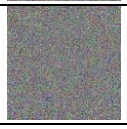
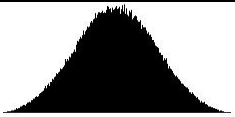
Teknik FTIE menggunakan Randomized Text dan ACM merupakan salah satu teknik yang disebut sebagai dynamic encryption yang dimana artinya setiap kali melakukan enkripsi walaupun menggunakan kunci yang sama maka akan menghasilkan chipertext yang berbeda-beda. Dikarenakan chipertext yang dihasilkan selalu berubah-ubah maka analisis yang dilakukan akan meliputi beberapa kondisi enkripsi diantaranya:

a. File yang sama menggunakan kunci yang sama

Analisis dengan menggunakan *file* yang sama dengan menggunakan kunci yang sama bertujuan untuk mengetahui seberapa besar keterkaitan pola antara *chipertext* yang satu dengan chipertext yang lain yang dienkripsi menggunakan kunci yang sama.

Berikut adalah tabel hasil enkripsi *file* "Intro Template.mp4" dengan menggunakan kunci enkripsi K = 8C4A73D5, P = 8491, dan Q = 1289. Hasil enkripsi yang diambil adalah 3 hasil enkripsi secara berurutan. Hasil enkripsi dapat dilihat pada tabel dibawah ini:

TABEL II. HISTOGRAM *FILE* YANG SAMA DENGAN KUNCI YANG SAMA

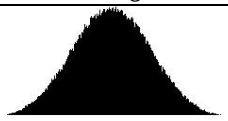
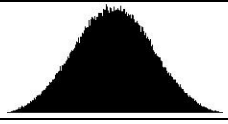
#	Gambar	Histogram
1		
2		
3		

Tabel II di atas menunjukkan bahwa histogram yang dihasilkan memiliki kemiripan pola histogram antara *chipertext* yang satu dengan *chipertext* yang lain walaupun *chipertext* yang dihasilkan berbeda-beda. Dari hal tersebut dapat dikatakan bahwa hasil enkripsi dari *file* yang sama dengan menggunakan kunci yang sama akan menghasilkan *chipertext* yang berbeda-beda akan tetapi dengan pola histogram yang sama.

b. File yang sama menggunakan kunci yang berbeda

Analisis dengan menggunakan *file* yang sama dengan menggunakan kunci yang berbeda dilakukan untuk membandingkan seberapa besar perubahan yang terjadi apabila sebuah *file* dienkripsi dengan menggunakan kunci yang berbeda-beda. Berikut tabel hasil enkripsi terhadap *file* "Intro Template.mp4" dengan menggunakan kunci yang berbeda-beda.

TABEL III. HISTOGRAM *FILE* YANG SAMA DENGAN KUNCI BERBEDA

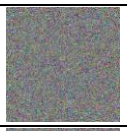
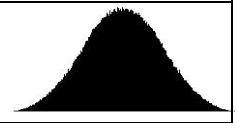
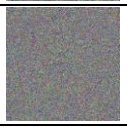
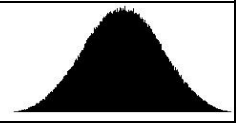
Kunci	Gambar	Histogram
K = TAC32AD1 P = 5422 Q = 9193		
K = D7JKEW3A P = 2038 Q = 1889		
K = AIU892JH P = 2631 Q = 9717		

Tabel III di atas menunjukkan bahwa histogram yang dihasilkan memiliki kemiripan pola histogram antara *chipertext* yang satu dengan *chipertext* yang lain. Dari hal tersebut dapat dikatakan bahwa hasil enkripsi menggunakan *file* yang sama dengan kunci yang berbeda-beda akan menghasilkan chipertext yang berbeda-beda akan tetapi dengan pola histogram yang sama. Sifat tersebut ternyata sama dengan sifat yang dimiliki oleh hasil enkripsi *file* yang sama menggunakan kunci yang sama.

c. File yang berbeda dengan kunci yang sama dan ukuran yang sama

Analisis dengan menggunakan 2 buah *file* yang berbeda namun memiliki ukuran yang sama dengan menggunakan kunci yang sama dilakukan untuk mengetahui apabila 2 jenis *file*nya berbeda apakah akan menghasilkan pola histogram yang berbeda ataukah pola histogram yang sama. Apabila pola histogram yang dihasilkan sama, maka hal tersebut menunjukkan bahwa *chipertext* aman terhadap analisa diferensial secara visual yang mencari korelasi pola antar *chiper*. Berikut adalah tabel hasil enkripsi dari *file* "rainbows3.jpg" dan "Surat 001.mp3" yang memiliki ukuran yang sama yaitu 944 *kilobytes*. Kunci enkripsi yang digunakan adalah K = 8C4A73D5, P = 8491, dan Q = 1289.

TABEL IV. HISTOGRAM *FILE* BERBEDA DENGAN KUNCI YANG SAMA

Nama *file*	Gambar	Histogram
rainbows3.jpg		
Surat 001.mp3		

Tabel IV di atas menunjukkan bahwa histogram yang dihasilkan memiliki kemiripan pola histogram antara *chipertext* yang satu dengan *chipertext* yang lain. Dari hal tersebut dapat dikatakan bahwa hasil enkripsi menggunakan *file* yang berbeda yang ukurannya sama dan dengan kunci enkripsi yang sama akan menghasilkan chipertext yang berbeda-beda namun dengan pola histogram yang sama.

2) Analisis NPCR

Analisis NPCR bertujuan untuk mengetahui jumlah *pixel* yang berubah antara chiper yang satu dengan chiper yang lain yang dienkripsi menggunakan kunci yang sama. Besarnya nilai NPCR menunjukkan bahwa tiap *pixel* pada gambar hasil enkripsi mengalami perubahan yang besar pula. Makin banyak nilai *pixel* yang berubah maka makin bagus kualitas acak yang

dihasilkan pada tiap kali enkripsi dilakukan. Analisis akan dilakukan dengan membandingkan antara hasil enkripsi dari *file* yang sama menggunakan kunci yang sama. Hasil analisis NPCR ditunjukkan pada tabel dibawah ini.

TABEL V. HASIL NILAI NPCR

No	Nama *File*	NPCR
1	CV.docx	99,27
2	Data Analisis.xlsx	99,49
3	Intro Template.mp4	99,34
4	Lampiran_2016725.zip	99,37
5	MITK_Kelompok.pptx	99,35
6	MTIF Bab 2.pdf	99,42
7	rainbows3.jpg	99,35
8	Surat 001.mp3	99,35

Dari Tabel V di atas terlihat rata – rata yang didapatkan untuk nilai pengujian NPCR hampir mencapai 100% atau mendekati sempurna. Hal ini mengindikasikan terjadi perubahan pada matriks *pixel* gambar secara merata. Nilai NPCR yang tinggi dapat diartikan bahwa sebuah algoritma tersebut memiliki resistensi yang tinggi terhadap *differensial attack* [13]. Menurut Ahmed [14] dan Huang [15] nilai NPCR ideal adalah pada keadaan mendekati 100% atau berada diatas 95%. Ini berarti tiap kali melakukan enkripsi menggunakan algoritma FTIE antara hasil enkripsi sebelumnya dengan hasil enkripsi sesudahnya hampir seluruh nilai-nya berubah

C. *Analisis Entropy*

Analisis *entropy* menunjukkan tingkat keacakan *chipertext* yang dihasilkan dari proses enkripsi. Makin besar nilai *entropy* maka makin bagus kualitas keacakan chipertext tersebut. Hasil analisis nilai *entropy* ditunjukkan pada table dibawah ini:

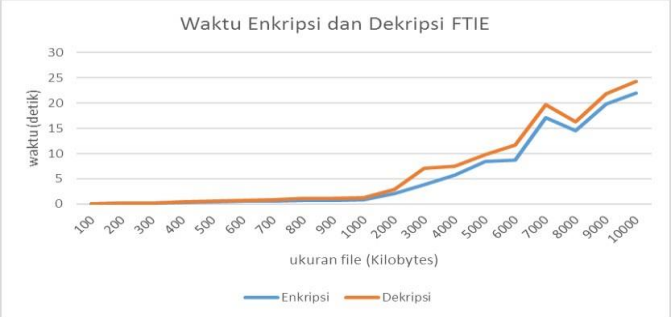
TABEL VI. HASIL NILAI ENTROPY

No	Nama *File*	*Entropy*
1	CV.docx	7.985
2	Data Analisis.xlsx	7.984
3	Intro Template.mp4	7.987
4	Lampiran_2016725.zip	7.983
5	MITK_Kelompok.pptx	7.985
6	MTIF Bab 2.pdf	7.985
7	rainbows3.jpg	7.987
8	Surat 001.mp3	7.987
Rata-rata		7.985

Tabel VI menunjukkan bahwa nilai rata-rata *entropy* dari seluruh *chipertext* adalah 7,985. berdasarkan hasil penelitian jolfaei dan mirghadri [16] menyatakan bahwa, jika sebuah informasi dienkripsi dan dalam kondisi teracak sempurna, maka nilai entropi yang ideal adalah ≈ 8. berdasarkan teori tersebut maka algoritma FTIE yang dirancang ini aman dari serangan *entropy* atau sulit ditebak oleh kriptanalis karena nilai informasi yang terdapat didalam gambar telah berada dalam keadaan teracak.

D. *Analisis Lama Waktu Eksekusi*

Analisis ini menunjukkan jumlah waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi berdasarkan ukuran *file*. Waktu enkripsi dan dekripsi secara grafik dapat dilihat pada gambar dibawah ini :



Gambar 7. Grafik Waktu enkripsi dan dekripsi FTIE berdasarkan ukuran *file*

Gambar 7 di atas menunjukkan berapa lama waktu yang dibutuhkan untuk melakukan enkripsi dan dekripsi dari sebuah file dengan rentan ukuran file yaitu dari 100 kilobytes sampai 10.000 kilobytes, dari grafik dapat dilihat bahwa waktu yang dibutuhkan untuk melakukan dari 100 sampai 2000 kilobytes mengalamani peningkatan yang cukup signifikan, peningkatan yang cukup besar terjadi pada ukuran file sekitar 7000 kilobytes. dari gambar tesebut juga memperlihatkan bahwa untuk melakukan enkripsi terhadap sebuah file dengan ukuran 10.000 kilobytes dibutuhkan waktu selama 21 detik. Waktu tersebut sangat lama untuk performa sebuah teknik kriptorafi sehingga dapat dikatakan bahwa teknik enkripsi FTIE tidak cocok digunakan untuk melakukan enkripsi terhadap file yang berukuran besar. Namun karena hasil dari enkripsi adalah sebuah gambar, dan ukuran normal dari sebuah gambar dengan ukuran dan resolusi yang besar kurang lebih bisa mencapai 10 MB, sehingga teknik FTIE dapat dikatakan sangat direkomendasikan hanya digunakan untuk file yang ukurannya tidak lebih dari ukuran normal sebuah gambar.

E. *Analisis Exhaustive Attack / Brute force Attack*

Pada penelitian ini, parameter kunci yang digunakan untuk melakukan enkripsi adalah *K, P, dan Q*. Kunci K, menggunakan 256 karakter, P dan Q kuncinya berupa angka antara 0 sampai 9. 53 . Diasumsikan minimal panjang karakter kunci K adalah 8 karakter. Dan Q 4 karakter. Maka jumlah kemungkinan kunci adalah:

≈ 2568 * 104 * 104

≈ 18.446.744.073.709.551.616 * 10.000 * 10.000

≈ 1.844.674.407.370.955.161.600.000.000

Jadi kemungkinan panjang kunci minimal untuk melakukan serangan bruteforce pada teknik FTIE adalah 1.844.674.407.370.955.161.600.000.000 kemungkinan. Jika diasumsikan sebuah komputer mampu melakukan perhitungan sebanyak 100 milyar per detik, maka waktu yang dibutuhkan untuk dapat membuka ciphertext dapat dihitung dengan melakukan pembagian antara jumlah kombinasi kunci dengan

jumlah kemampuan komputer dalam melakukan komputasi adalah sebagai berikut:

$$\approx \frac{1.844.674.407.370.955.161.600.000.000}{100.000.000.000}$$

$\approx$ 18.446.744.073.709.551 detik

$\approx$ 5.124.095.576.030.431 jam.

$\approx$ 213.503.982.334.601 hari

$\approx$ 7.019.301.315.875 bulan

$\approx$ 584.942.417.355 tahun

Dari data percobaan perhitungan diatas, terlihat waktu yang dibutuhkan untuk melakukan percobaan dekripsi menggunakan metode bruterforce terhadap minimal kemungkinan kunci yang ada sangat lama atau tidak mungkin untuk dilakukan. Sehingga dapat diambil kesimpulan bahwa penerapan teknik FTIE dapat menghasilkan chipertext yang aman dari serangan bruteforce.

F. Uji Integritas

Pengujian integritas terhadap dokumen hasil dekripsi, dilakukan dengan tujuan untuk mengetahui apakah terdapat perubahan terhadap nilai hash dari sebuah *file* yang telah di dekripsi dengan nilai hash dari *file* asli yang digunakan untuk melakukan ujicoba sebelumnya. Pengujian integritas akan dilakukan pada *file* "rainbows3.jpg". Skenario pengujian yaitu pertama dengan melakukan enkripsi terhadap *file* "rainbows3.jpg" kemudian hasil akan didekripsi kembali dengan menggunakan kunci yang sama seperti kunci ketika dilakukan enkripsi. Hasil dari pengujian terhadap integritas dapat dilihat pada tabel berikut ini:

TABEL VII. HASIL UJI INTEGRITAS

Nama File	Hash Original	Hash Hasil Dekripsi
CV.docx	af92b01c3d9c856f34870619c780a868	af92b01c3d9c856f34870619c780a868
Data Analisis.xlsx	3f1b83f8a5a3435fb591355bdf733c4b	3f1b83f8a5a3435fb591355bdf733c4b
Intro Template.mp4	679090bf2c0e75d2cc39b8c79d62ae0c	679090bf2c0e75d2cc39b8c79d62ae0c
Lampiran_2016725.zip	7dd231d886f28e5f4db8db2f95089bac	7dd231d886f28e5f4db8db2f95089bac
MITK_Kelompok.pptx	e825bb4565d6f233363a9b93e0a4ca66	e825bb4565d6f233363a9b93e0a4ca66
MTIF Bab 2.pdf	7a578f9792b54215b8930af7dae1ee13	7a578f9792b54215b8930af7dae1ee13
rainbows3.jpg	c8d2021ac303e7ecc9287e74cd3f026c	c8d2021ac303e7ecc9287e74cd3f026c
Surat 001.mp3	176da0f1f3f9f064407020b2e517b175	176da0f1f3f9f064407020b2e517b175

Dapat dilihat pada Tabel VII bahwa nilai hash antara *file* asli dengan nilai *hash* setelah dilakukan dekripsi adalah sama. Hal ini menunjukkan bahwa ekripsi teknik FTIE menggunakan algoritma *Randomized Tex*t dan ACM dapat dipertanggungjawabkan integritasnya.

V. KESIMPULAN

Setelah melakukan beberapa hal terkait dengan perancangan, pengujian dan analisis maka diperoleh beberapa kesimpulan sebagai berikut:

1) Teknik enkripsi FTIE menggunakan algoritma *Randomized Text* dan ACM terdiri dari 3 tahap, yaitu tahap FTIE, tahap enkripsi menggunakan algoritma *Randomized Text* dan tahap pengacakan menggunakan algoritma ACM.
2) Berdasarkan hasil analisis dan pengujian yang dilakukan, teknik FTIE menggunakan algoritma *Randomized Text* dan ACM memiliki ketahanan yang kuat dari berbagai macam analisis seperti analisis histogram, analisis NPCR, analisis *entropy*, dan tahan terhadap serangan *bruteforce*. Selain itu, uji integritas menunjukkan bahwa hasil dekripsi FTIE memiliki integritas yang dapat dipertanggungjawabkan.
3) Implementasi teknik FTIE menggunakan algoritma *Randomized Text* dan ACM pada sebuah aplikasi telah berhasil melakukan proses enrkipsi dan dekripsi dengan baik, serta menunjukkan bahwa teknik FTIE memiliki kemampuan untuk menyamarkan *file* yang dienkripsi sehingga akan menyulitkan seorang kriptanalis untuk menentukan jenis dari *file* tersebut.

DAFTAR PUSTAKA

[1] Ariyus, D. (2006). Kriptografi Keamanan Data dan Komunikasi. Yogyakarta: Graha Ilmu

[2] Oppliger, R. (2005). Contemporary Cryptography. Retrieved from http://www.esecurity.ch/serieseditor.html

[3] Kromodimoeljo, S. (2009). *Teori dan Aplikasi Kriptografi*. SPK IT Consulting.

[4] Menezes, A. J., Oorschot, P. C. Van, & Vanstone, S. a. (1997). Handbook of Applied Cryptography. *Electrical Engineering*, *106*, 780. http://doi.org/10.1.1.99.2838

[5] Singh, S., & Jain, A. (2013). An Enhanced Text to Image Encryption Technique using RGB Substitution and AES, *4*(May), 2108–2112.

[6] Memon, J., Zaidi, M., Rozan, A., Uddin, M., Abubakar, A., Chiroma, H., & Daud, D. (2014). Randomized Text Encryption : a New Dimension in Cryptography, *9*(February), 365–374.

[7] Gallager, R. (1972). Information Theory and Reliable Communication. New York: Springer-Verlag Wien.

[8] Lin, C., & Chang, S. (2001). Distinguishing JPEG Compression from. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS OF VIDEO TECHNOLOGY*, *11*(2), 153–168.

[9] Irfan, P., & Prayudi, Y. (2015). Penggabungan Algoritma Chaos dan Rivers Shamir Adleman (RSA) untuk Peningkatan Keamanan Citra, 5–10.

[10] Ahmad Abusukhon Mohammad Talib, M. A. N. (2012). Analyzing the Efficiency of Text-to-Image Encryption Algorithm. *International Journal of Advanced Computer Science and Applications(IJACSA)*, *3*(11), 35–38. Retrieved from http://ijacsa.thesai.org/

[11] Struss, K. (2009). A Chaotic Image Encryption. *Mathematics Senior Seminar*. University of Minnesota, Morris.

[12] Jolfaei, A., & Mirghadri, A. (2011). Image Encryption Using Chaos and Block Cipher, *4*(1), 172–185.

[13] Liu, H., & Wang, X. (2010). Color image encryption based on onetime keys and robust chaotic maps. *Computers and Mathematics with Applications*, *59*(10), 3320–3327. http://doi.org/10.1016/j.camwa.2010.03.017

[14] Wu, Y. (2011). NPCR and UACI Randomness Tests for Image Encryption

[15] Ahmed, H. (2007). Encryption Efficiency Analysis and Security Evaluation of RC6 Block Cipher for Digital Images, *3*(1), 1–7.

[16] Huang, C. K. (2010). Multi Chaotic Systems Based Pixel Shuffle for Image Encryption.