Kriptografi Visual dengan Metode Color Split

Aden Rohmana[#1]

[#]*Departemen Teknik Informatika ITB,*
Jl. Ganesha No. 1 Bandung - Indonesia
[1]mysticswordsman21@gmail.com

Abstraksi— **Penerapan kriptografi visual untuk *natural image* cenderung kurang mempertahankan kualitas image setelah proses enkripsi – dekripsi. Gambar yang dihasilkan dengan penggabungan *shares* cenderung mengandung banyak *noise*. Atas dasar itu dirumuskan metode kriptografi visual *color split* yang diharapkan dapat mempertahankan kualitas *image* hasil dekripsi. Metode *color split* menggunakan tiga prinsip utama yakni basis operasi XOR, *color channel split*, dan *noise randomizer*.**

Kata Kunci— **kriptografi visual, *color split, pixel, plain image, share image, combined image*.**

I. PENDAHULUAN

Kriptografi visual telah dikembangkan dengan berbagai metode, mulai dari metode dasar pemisahan keutuhan pixel sampai ke kriptografi visual yang digabungkan dengan steganografi. Dalam metode kriptografi visual pada citra biner atau teks, adanya *noise* tidak begitu mempengaruhi keutuhan pesan. Sedangkan pada kriptografi visual pada citra berwarna / *natural image*, adanya *noise* sangat mempengaruhi keutuhan dan keterbacaan pesan.

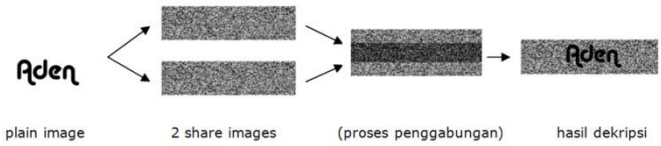
Metode kriptografi visual *color split* adalah metode kriptografi visual baru, yang diharapkan dapat mempertahankan kualitas image hasil dekripsi. Dasar metode *color split* yakni dengan menggunakan prinsip pemisahan *color value*, berbasis representasi *channel digital image* yakni RGB. Makalah ini membahas rancangan, implementasi, dan pengujian metode kriptografi visual *color split*.

II. STUDI LITERATUR

A. *Kriptografi Visual*

Kriptografi visual adalah teknik kriptografi yang membuat informasi visual (gambar, grafik, teks, dll) dienkripsi sedemikian hingga proses dekripsi dapat dilakukan dengan daya visual manusia, tanpa bantuan komputer.

Visual kriptografi awalnya ditemukan oleh Moni Naor dan Adi Shamir di tahun 1994. Mereka memperkenalkan skema *visual secret sharing*, dimana sebuah gambar dibagi menjadi sejumlah n bagian / *share*, biasanya berbentuk transparansi. Hanya orang yang mempunyai seluruh n *share* yang dapat mendekripsi gambar itu. Contoh kriptografi visual sederhana ditunjukkan pada gambar 1.



Gambar 1. Contoh kriptografi visual

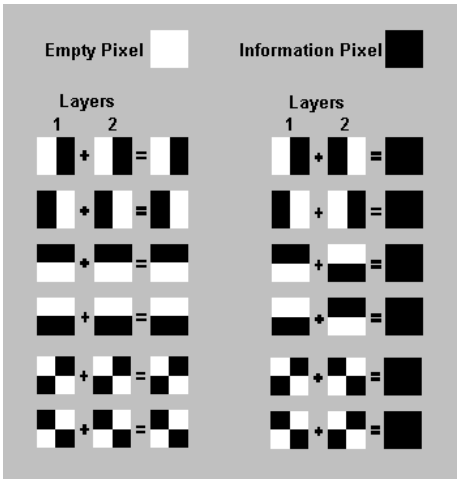
B. *Metode Kriptografi Visual*

Pada dasarnya, metode enkripsi kriptografi visual berbasis pada *pixel splitting*, yakni mempertimbangkan keutuhan *pixel*. Kriptografi visual yang dieksekusi oleh enkriptor khusus dapat melakukan *pixel-splitting* ini dengan rapi dan teratur.

Model sederhana yang dikemukakan oleh Moni Naor dan Adi Shamir yakni kriptografi visual pada citra biner. Pada masing-masing *pixel*, dilakukan hal sebagai berikut:

1. Membangkitkan permutasi acak p pada himpunan {1, ..., m!}.
2. Jika pixel P berwarna hitam, maka ambil matriks boolean S dari C_0 pada indeks ke-p. Sedangkan jika berwarna putih, diambil dari C_1.
3. Untuk $1 \leqslant i \leqslant n$, baris ke-i pada S menyatakan seluruh *subpixel* dari P pada share ke-i.

Setiap *pixel* gambar dibagi menjadi bagian yang lebih kecil lagi. Jumlah blok hitam dan blok putih akan selalu sama. Jika sebuah pixel dibagi menjadi dua bagian, akan ada satu blok hitam dan satu blok putih. Ini dijelaskan pada gambar 2.



Gambar 2. Metode pixel splitting dengan 4 blok (Rijmenants, 2004)

C. *Aplikasi Kriptografi Visual*

Kriptografi visual dapat digunakan untuk mengamankan suatu pesan / informasi yang dibagi ke berbagai pihak. Tingkat keamanan pesan / informasi tersebut menjadi lebih kuat, karena orang perlu mengumpulkan semua share dari pihak – pihak tersebut.

Kriptografi visual juga dapat digunakan untuk implementasi enkripsi *one-time pad*, dimana sebuah transparansi berfungsi sebagai *shared random pad*, dan sisanya berfungsi sebagai *ciphertext*.

D. *Model Warna RGB*

Model warna RGB adalah format warna dengan prinsip *additive color model* dimana cahaya berwarna merah, hijau, dan biru ditambahkan dengan metode tertentu untuk menghasilkan warna – warna yang ada. Warna merah, hijau dan biru ini yang digunakan karena merupakan tiga warna dominan dalam spektrum cahaya. Gambaran pencampuran warna RGB dapat dilihat pada gambar 3.

Gambar 3. Basis model warna RGB

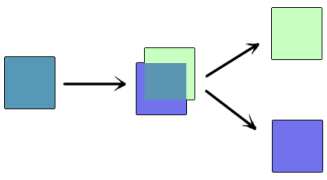
Dalam perumusan metode kriptografi color split, model warna RGB digunakan sebagai dasar pemisahan warna per *channel* (R, G, dan B).

III. Analisis dan Perancangan Solusi

A. *Konsep Metode Color Split*

Metode kriptografi visual yang diterapkan adalah berbasis pemisahan *color value* tiap *pixel* dari plain image. *Color value* diambil dari representasi warna RGB; channel *Red*, *Green* dan *Blue*.

Metode ini tidak mengubah jumlah total *pixel* dan dimensi dari *plain image* ke *cipher image*, sehingga *cipher images* dan hasil citra penggabungan berdimensi sama dengan plain image. Gambar 4 menunjukkan gambaran umum *color split* per *pixel*.

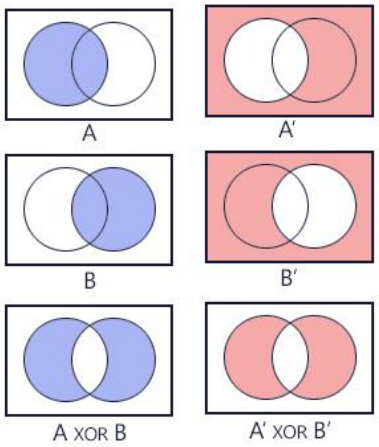


Gambar 4. Gambaran umum color split per pixel

B. *Penggunaan Operasi XOR Sebagai Basis Pemisahan dan Penggabungan*

Pada umumnya *noise* yang di*generate* akan mereduksi citra hasil penggabungan. Solusi metode yang mempertahankan keduanya adalah dengan kriptografi visual berbasis XOR.

Inversi pada basis XOR memungkinkan *noise generation* yang mengembalikan nilai yang sama dengan operasi semula. Karakteristik XOR yang dimaksud ditunjukkan pada gambar 5.



A XOR B = A' XOR B'

Gambar 5. Karakteristik XOR yang dimanfaatkan

C. *Alur Proses Metode Color Split*

Dilakukan perumusan metode untuk proses enkripsi saja, sedangkan proses dekripsi cukup melalui operasi XOR seluruh *share images*. Proses enkripsi metode kriptografi visual color split meliputi beberapa tahap, antara lain:

a) *Image Iterator*
b) *Channel splitter*
c) *Color split – Middle Split* dan *Full Split*
d) *Noise Randomizer*

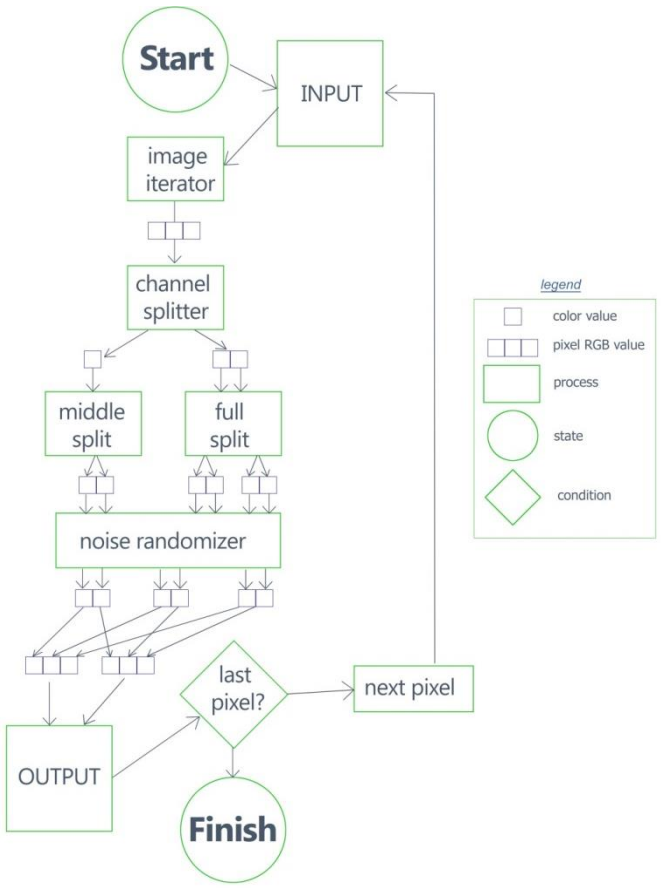
Proses dimulai dengan masukan gambar (*plain image*), disimpan di dalam variabel *image_plain*. Kemudian *image iterator* melakukan iterasi per pixel, mengambil nilai RGB tiap pixel. Nilai RGB ini disimpan dalam variabel pixel_plain.

Channel splitter berfungsi untuk memisahkan nilai RGB ke masing-masing channel Red, Green, dan Blue. plain_pixel diproses oleh channel splitter, mengeluarkan tiga *value* sesuai nilai RGB plain_pixel, masing-masing Red, Green, dan Blue.

Kemudian *channel-channel* tersebut diproses oleh *color splitter – middle split* dan *full split*. Satu diantara tiga *channel* tersebut diproses dengan *middle split*, dan dua sisanya diproses dengan *full split*. Hasil *split* yakni tiap *channel* terbagi menjadi sejumlah *n_cipher* (jumlah *cipher* yang diinginkan). Output ini dimasukkan ke variabel *array of pixels* pixel_cipher[ ].

Setelah terbagi menjadi dua, pixel_cipher[ ] diproses ke *noise randomizer*. Di sini ditentukan apakah *pixel* ini menjadi *noise pixel* atau tidak. Kemudian hasilnya berupa *final cipher pixel* yang dimasukkan ke *output buffer* ber-type *array of images* image_cipher.

Lalu jika *pixel* yang diproses tadi adalah *pixel* terakhir dari plain image, maka proses enkripsi berakhir. Jika tidak, sistem meneruskan proses enkripsi ke pixel berikutnya. Alur prosesnya secara lengkap dijelaskan pada gambar 6.

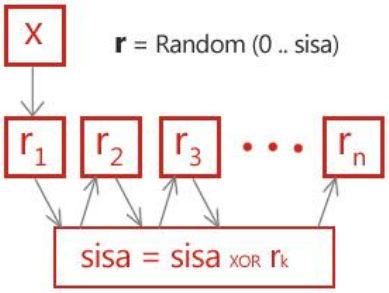


Gambar 6. Alur Metode Kriptografi Visual *Color Split*

D. *Color Split – Middle Split dan Full Split*

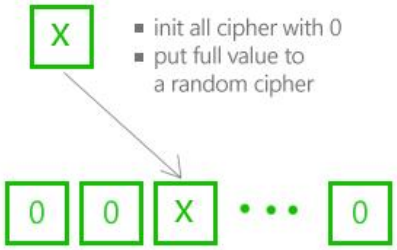
Untuk memaksimalkan variansi *pixel* pada *cipher image*, *color split* dilakukan dengan dua cara yakni *middle split* dan *full split*.

1) *Middle Split:* Membagi RGB *value* dari *plain image* ke masing-masing *cipher image* secara parsial. Pembagian dilakukan secara sekuensial, dan besar pembagian per *cipher* ditentukan secara acak. Skema *middle split* ditunjukkan pada gambar 7.



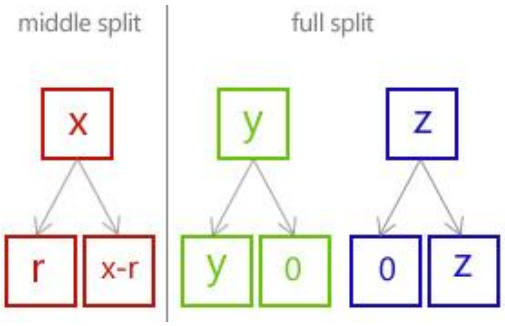
Gambar 7. Skema *Middle Split*

2) *Full Split:* memindahkan *value* dari *plain image* ke salah satu *cipher image* secara acak. *Value cipher* sisanya bernilai nol. Skema *Full Split* dijelaskan pada gambar 8.



Gambar 8. Skema *Full Split*

Metode kriptografi visual *color split* ini menggunakan dua channel *full split* dan satu *middle split*, dengan contoh ditunjukkan pada gambar 9.



Gambar 9. Contoh *Channel Split*

E. *Noise Randomizer*

Setelah proses pemisahan warna, *pixel value* pada *cipher images* telah terdistorsi sampai suatu tingkat tertentu. Namun karena warna hanya dapat dipisah terbatas pada rentang nol sampai *original value*, maka *cipher images* cenderung masih dapat dikenali oleh indera manusia. Maka dari itu diperlukan *noise* untuk kamuflase / menyamarkan gambar.

Perbandingan jumlah kontras warna yang optimal agar memaksimalkan ketidakteraturan adalah 1:1. Maka persentase noise yang digunakan untuk *optimum contrast* adalah 50% dari seluruh *pixel*.

Noise generation dibuat berdasarkan karakteristik XOR yang telah dibahas di analisis, yakni inverse. Maksimum *value* dari tiap channel RGB adalah 255, maka inverse dari *value* adalah 255 – *value*.

Nilai inversi XOR yang mengembalikan nilai yang sama hanya berlaku pada jumlah cipher genap. Untuk jumlah *cipher* ganjil, salah satu *noise* di antara *cipher images* dihilangkan.

F. *Proses Dekripsi*

Metode dekripsi yakni menggunakan operasi XOR semua *pixel cipher*. Urutan operasi XOR tidak harus urut. Berikut adalah skema dekripsi dengan XOR.

pixel_combine = pixel_cipher[0] XOR pixel_cipher[1] XOR pixel_cipher[2] .... XOR pixel_cipher[n_cipher]

IV. Implementasi Perangkat Lunak

A. *Model Use Case Perangkat Lunak*

Pemodelan *use case* dibangun sesuai kebutuhan fungsional dan kebutuhan non-fungsional perangkat lunak. User dapat melakukan *load image*, *split image*, *combine image*, dan *save image*.

B. *Rancangan Kelas*

Program terdiri dari tiga kelas utama, yakni SplitterMain, ImagePanel, dan ImageProcessor. SplitterMain adalah kelas utama keseluruhan program. ImagePanel berfungsi untuk tempat penyimpanan gambar dan untuk *preview image*. ImageProcessor berfungsi sebagai pemroses image dengan metode color split.

C. *Implementasi Antarmuka*

Program memiliki antarmuka sederhana yakni satu *window* utama untuk proses enkripsi dan dekripsi. Keseluruhan antarmuka dibagi dalam 3 panel yakni panel plain image, cipher images, dan combined image.

V. Pengujian

Pengujian yang dilakukan bertujuan untuk menguji:

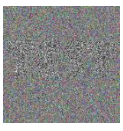
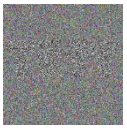
- Kesesuaian hasil implementasi dengan spesifikasi perangkat lunak
- Tes pada metode kriptografi visual color split
- Melihat tingkat performansi metode, meliputi keamanan skema dan keamanan individual share images.

Skenario pengujian yakni dengan melakukan proses enkripsi-dekripsi pada berbagai jenis *image*, dan dengan jumlah skema *n_cipher* antara (2, 2) – (6, 6).

A. *Performansi pada Bermacam Jenis Image*

Hasil pengujian pada bermacam jenis image ditunjukkan pada tabel II.

TABEL I
PERFORMANSI PADA BERMACAM JENIS IMAGE

Jenis *Image*	***Plain Image***	***Share Images***		***Combined Image***
Biner				
Grayscale				

Jenis *Image*	***Plain Image***	***Share Images***		***Combined Image***
Berwarna – kartun				
Berwarna – *natural image*				

Secara umum performansi sudah bagus di bermacam jenis *image*, kecuali pada citra biner. Proses enkripsi pada citra biner menghasilkan *share images* yang cenderung masih kelihatan bentuknya seperti pada *plain image*, disebabkan karena citra biner hanya mengandung nilai warna 0 dan 255 yang masing-masing terkumpul dalam blok area yang berdekatan.

B. *Performansi Ketidakteraturan share images*

Hasil pengujian dalam hal ketidakteraturan *share images* ditunjukkan pada tabel III.

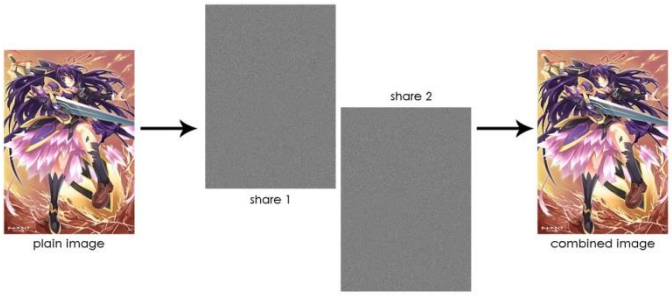
TABEL II
PERFORMANSI KETIDAKTERATURAN SHARE IMAGES

Split	***Share Images***					
1 (*Plain Image*)						
2						
3						
4						
5						
6						

Ketidakteraturan *share images* bagus pada jumlah *share* genap, namun kurang bagus di jumlah *share image* ganjil. Hal ini karena pada algoritma *noise adder*, sesuai sifat XOR, pada jumlah *share image* ganjil perlu dikurangi *noise* pada salah satu share image.

C. *Performansi Kualitas Image Hasil Dekripsi*

Pada pengujian bagian ini, didapatkan bahwa pada semua kasus kualitas *image* hasil dekripsi secara visual persis dengan *plain image*. Hasil pengujian kualitas *image* dekripsi ditunjukkan pada gambar 13.



Gambar 10. Hasil pengujian kualitas image dekripsi

D. *Performansi Keamanan Metode*

Hasil pengujian perihal keamanan metode ditunjukkan pada tabel IV.

TABEL III
PERFORMANSI KEAMANAN METODE

Split	*Combine*				
	2	3	4	5	6
1 (*plain image*)					
2					
3					
4					
5					
6					

Pada setiap skema, cukup menggunakan 2 *share images* saja sudah didapatkan gambaran bentuk *plain image* walau ada *noise*. Ini terutama disebabkan karena metode *color split* cenderung membagi *color value* merata pada tiap *share images*. *Color value* yang terbagi merata tersebut cenderung dapat dikenali secara parsial dengan hanya dua *share images* saja.

Pada awalnya metode *color split* hanya dirancang untuk skema kriptografi visual (2,2). Maka untuk keamanan, dapat dikatakan metode *color split* bersifat aman untuk skema (2,n). Sedangkan penggabungan seluruh *share image* sejumlah n adalah untuk *perfection*, untuk mendapatkan *image* dekripsi yang persis dengan *plain image*.

E. *Pengujian Fungsionalitas*

Pengujian fungsionalitas berjalan seperti harapan. Sesuai spesifikasi, aplikasi dapat melakukan *load image, preview image , split image, combine image* dan *save image*.

VI. PENUTUP

A. *Kesimpulan*

Kesimpulan yang diperoleh dengan pengerjaan tugas akhir ini antara lain:

1. Metode Kriptografi Visual *color split* telah dirumuskan dengan dasar pemisahan *pixel color value* dan menggunakan basis operasi XOR.
2. Perangkat lunak implementasi metode kriptografi visual color split telah dibuat dan berjalan sesuai harapan.
3. Kualitas gambar hasil penggabungan berhasil dipertahankan, karena dekripsi *share images* dari metode *color split* menghasilkan *image* yang serupa dengan *plain image.*
4. Metode *color split* bagus digunakan untuk jumlah *share* genap, namun kurang bagus digunakan untuk jumlah *share* ganjil. Ini disebabkan karena enkripsi dengan metode *color split* menghasilkan *share images* yang tidak teratur untuk jumlah *share* genap, namun masih cenderung teratur untuk jumlah *share* ganjil.
5. Tingkat keamanan lebih baik untuk jumlah *share* kecil saja, yakni 2. Untuk skema selebihnya relatif kurang aman karena dapat didekripsi dengan menggabungkan 2 *share image* saja.

B. *Saran*

Saran pengembangan tugas akhir ini adalah:

1. Metode akan lebih baik jika disempurnakan terutama untuk jumlah share images ganjil.
2. Skema yang ada (2,n) akan lebih baik jika bisa dikembangkan menjadi skema (k,n) yang fleksibel sehingga diperlukan sejumlah k share images untuk mendapatkan hasil yang berbentuk seperti plain image.

DAFTAR REFERENSI

[1] Poynton, C. A. (2002) : *Digital Video and HD: Algorithms and Interfaces (The Morgan Kaufmann Series in Computer Graphics)*, San Fransisco, U.S.A., Morgan Kaufmann Publishers, 234 – 237.

[2] Galer, M, dan Horvat, L. (2002): *Digital Imaging: Essential Skills, Second Edition (Photography Essential Skills)*, Burlington MA, Focal Press, 72-75.

[3] Nakajima, M. dan Yamaguchi, Y. (2005) : *Extended Visual Cryptography for Natural Images*, Departement of Graphics and

Computer Sciences Graduate School of Arts and Science – The University of Tokyo, Japan.

[4] Romdhoni, M. A. (2006) : *Kriptografi Visual pada Citra Biner dan Citra Berwarna serta Pengembangannya dengan Steganografi dan Fungsi XOR*, Makalah Tugas Akhir Program Studi Informatika, Institut Teknologi Bandung.

[5] Blundo, C., Santis, A. D., dan Stinson, D. R. (1998) : *On the Contrast in Visual Cryptography Schemes*, Departement of Computer Science and Engineering University of Nebrasca-Lincoln, Lincoln NE 68588, USA.

Pustaka dari situs internet:

[6] Shamir, A. Dan Naor, M. (1994) : *Visual Cryptography*, http://www.cs.nccu.edu.tw/~raylin/UndergraduateCourse/ComtenporaryCryptography/Spring2009/VisualCrypto.pdf, diunduh pada 29 Desember 2012.

[7] Rijmenants, D. (2004) : *Cipher Machines and Cryptology - Visual Cryptography*, http://users.telenet.be/d.rijmenants/en/visualcrypto.htm, diunduh pada 29 Desember 2012.

[8] King, J, C. (2008) : *Why Color Management?*, Adobe Systems Incorporated, http://www.color.org/whycolormanagement.pdf, diunduh pada 29 Desember 2012.