Implementasi Kriptografi Visual pada Gambar Berwarna 24-bit RGB dengan *Image Blending*

dengan pustaka Numpy dan OpenCV

Andika Naufal Hilmy - 13520098
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13520098@std.stei.itb.ac.id

***Abstract*—Kriptografi visual memungkinkan enkripsi citra secara visual di mana secara visual citra yang terenkripsi tidak memiliki makna. Pada makalah ini, algoritma yang digunakan untuk kriptografi visual dengan *image blending* berdasarkan konsep pencampuran warna pada citra yang ditumpuk. Algoritma *image blending* yang dipilih adalah *multiply*, *screen*, dan *overlay*. Algoritma ini memiliki keterbatasan dalam mendekripsi share sehingga dekripsi kembali yang dilakukan tak sepenuhnya sempurna.**

Keywords—kriptografi visual; image blending.

I. Latar Belakang

Kriptografi visual merupakan salah satu cabang kriptografi yang melibatkan enkripsi dan dekripsi citra dengan memecah citra menjadi beberapa *share* berupa citra acak[1]. Dengan mengenkripsi citra, gambar rahasia tak dapat diketahui dengan mudah oleh pihak ketiga. Kriptografi visual dapat diimplementasikan pada berbagai jenis citra, baik citra hitam-putih, citra *grayscale*, ataupun citra berwarna.

Kriptografi visual telah diterapkan pada kasus nyata, terutama pada penyebaran gambar rahasia. Proses ini dilakukan pada keamanan biometrik, *watermarking*, *e-voting*, dan sebagainya[2].

Dalam makalah ini, implementasi kriptografi visual dilakukan dengan penggunaan *image blending* pada dekripsi *share*.

II. Dasar Teori

A. Kriptografi Visual

Kriptografi visual pertama kali diusulkan oleh Moni Naor dan Adi Shamir pada 1994 dengan cara membagi citra ke beberapa *share* untuk digabungkan kembali saat dekripsi[3]. Algoritma yang diterapkan pada kriptografi visual bergantung pada jenis citra karena kompleksitas algoritma yang meningkat seiring banyaknya kombinasi warna.

Pada gambar 1-bit (hitam-putih), proses enkripsi dilakukan dengan lebih sederhana dengan hanya mengekspansi satu piksel menjadi beberapa subpiksel. *Share* yang dihasilkan memiliki tiga kemungkinan, yaitu *share* horizontal, *share* vertikal, dan *share* diagonal. Pembangkitan *share* pada kriptografi visual memiliki beberapa skema sebagai berikut[1]:

- Skema (2, 2)

  Skema ini merupakan skema paling sederhana dengan pembagian citra menjadi dua buah *share* dan membutuhkan dua buah *share* untuk dekripsi. Beberapa cara pembagian *share* pada skema ini adalah *share* horizontal, *share* vertikal, dan *share* diagonal.

- Skema (2, *n*)

  Skema ini membagi citra menjadi *n* buah *share*, tetapi hanya dua *share* yang dibutuhkan untuk dekripsi.

- Skema (3, 3)

  Skema ini membagi citra menjadi tiga buah *share* dan membutuhkan ketiga *share* untuk dekripsi.

- Skema (3, *n*)

  Skema ini membagi citra menjadi *n* buah *share*, tetapi hanya tiga *share* yang dibutuhkan untuk dekripsi.

- Skema (*k*, *n*)

  Skema ini membagi citra menjadi *n* buah *share* dan hanya membutuhkan *k* buah *share* untuk dekripsi. Skema ini merupakan bentuk umum dari pembagian *share* pada proses enkripsi.

Adapun dekripsi dari *share-share* yang ada dilakukan dengan fungsi tertentu untuk mendapatkan kembali gambar aslinya. Pada skema (*k, n*) sejumlah minimal *k* citra didekripsi dengan ditumpuk dan diproses untuk mendapatkan kembali gambar semula. Gambar yang didekripsi diharapkan sesuai dengan gambar semula.

B. Image Blending

Image blending merupakan metode pencampuran warna pada dua citra yang ditumpuk dengan fungsi tertentu dengan bentuk umum

$$f(a,b) = c$$

yang berlaku pada tiap piksel dengan nilai *a, b,* dan *c* merupakan bilangan real berada di antara 0 dan 1. Berikut ini beberapa fungsi *image blending*[4][5]:

- *Multiply*
  Fungsi ini memiliki bentuk berikut:

  $$f(a,b) = ab$$

  Jika fungsi ini diinvers dengan nilai *a* diketahui, fungsi inversnya adalah sebagai berikut:

  $$f(a,b) = ab = c$$
  $$<=> f^{1}(a,c) = b = c/a$$

- *Screen*
  Fungsi ini memiliki bentuk berikut:

  $$f(a,b) = 1 - (1-a)(1-b)$$

  Jika fungsi ini diinvers dengan nilai *a* diketahui, fungsi inversnya adalah sebagai berikut:

  $$f(a,b) = 1 - (1-a)(1-b) = c$$
  $$<=> f^{1}(a,c) = b = 1-(1-c)/(1/a)$$

- *Overlay*
  Fungsi ini memiliki bentuk berikut:

  $$f(a,b) = 2ab,\ a < 0.5;$$
  $$1 – 2(1-a)(1-b),\ a \geq 0.5;$$

  Jika fungsi ini diinvers dengan nilai *a* diketahui, fungsi inversnya adalah sebagai berikut:

  $$f(a,b) = 2ab,\ a < 0.5;$$
  $$1 – 2(1-a)(1-b),\ a \geq 0.5;$$
  $$<=> f^{1}(a,c) = b = c/2a,\ a < 0.5;$$
  $$1 – (1-c)/2(1-a),\ a \geq 0.5;$$

III. IMPLEMENTASI

Implementasi kriptografi visual dengan image blending dilakukan dengan skema (n,n). Implementasi dilakukan di Python 3.12 dengan bantuan pustaka Numpy untuk kalkulasi dan OpenCV untuk membaca dan menulis berkas citra. Algoritma yang diimplementasikan antara lain *multiply*, *screen*, dan *overlay*. Secara sederhana, algoritma enkripsi-dekripsi adalah sebagai berikut dalam *pseudocode*:

Enkripsi

```
function genefunction encrypt(image, number_of_shares) {
    image_pixel_array = get_pixels(image)
    shares = array(number_of_shares)
    for i in (0 <= i <= number_of_shares - 2) {
        shares[i] = random_numbers(
            size: image_pixel_array.size,
            low: 1, # Mencegah pembagian dengan nol
            high: 255
        )
        remaining_share = inverse_blend(shares[i], image_pixel_array) # Invers fungsi multiply, overlay, atau screen
    }
    shares.append(remaining_share)
    return shares
}
```

Dekripsi

```
function decrypt(shares) {
    n = length(shares)
    decrypted_image = shares[0]
    for i in (1 <= i <= n - 1) {
        decrypted_image = blend(decrypted_image, shares[i])
    }
    return decrypted_image
}
```

Kunci dari algoritma enkripsi-dekripsi ada pada bagaimana fungsi *image blending* diimplementasikan. Pada makalah ini, terdapat tiga fungsi beserta inversnya yang diimplementasikan sebagai berikut (berupa *pseudocode*):

Multiply

```
function multiply(bottom, top) {
    result = array(bottom.size)
    for each pixel {
        result.pixel = bottom.pixel * top.pixel / 255
    }
    return result
}
```

Screen

```
function screen(bottom, top) {
    result = array(bottom.size)
    for each pixel {
        result.pixel = 255 - ((255 - bottom.pixel) * (255 - top.pixel) / 255)
    }
    return result
}
```

Overlay

```
function overlay(bottom, top) {
    result = array(bottom.size)
    for each pixel {
        if bottom.pixel < 128 {
            result.pixel = 2 * bottom.pixel * top.pixel / 255
        }
        else {
            result.pixel = 255 - (2 * (255 - bottom.pixel) * (255 - top.pixel) / 255)
        }
    }
    return result
}
```

Inverse Multiply

```
function inverse_multiply(bottom, original_image) {
    result = array(bottom.size)
    for each pixel {
        result.pixel = 255 * original_image.pixel / bottom.pixel
    }
    return result
}
```

Inverse Screen

```
function inverse_screen(bottom, original_image) {
    result = array(bottom.size)
    for each pixel {
        result.pixel = 255 - (255 * (255 -
original_image.pixel) / (255 - bottom.pixel))
    }
    return result
}
```

Inverse Overlay

```
function inverse_overlay(bottom, original_image) {
    result = array(bottom.size)
    for each pixel {
        if bottom.pixel < 128 {
            result.pixel = (255 * original_image.pixel) / (2 *
bottom.pixel)
        }
        else {
            result.pixel = 255 - (255 * (255 -
original_image.pixel) / (2 * (255 - bottom.pixel)))
        }
    }
    return result
}
```

IV. Pengujian dan Hasil

A. Pengujian

Terdapat dua skenario pengujian yang dilakukan pada ketiga algoritma yang diimplementasikan:

- Enkripsi-dekripsi langsung

  Enkripsi-dekripsi dilakukan dengan skema (*n,n*) dengan nilai *n* merupakan subset dari {2, 3, 4}. Skenario ini akan mengukur keacakan dari *share* dan kesesuaian gambar dekripsi dengan gambar enkripsi. Adapun skenario kontrolnya adalah dengan enkripsi fungsi XOR.

- Lama eksekusi, baik dekripsi maupun enkripsi dengan berbagai ukuran gambar

  Lama masing-masing proses enkripsi-dekripsi pada jumlah *share* yang berbeda dapat berpengaruh pada waktu eksekusi. Skenario ini akan mencari hubungan jumlah *share* dengan lama eksekusi.

Hasil pengujian akan diukur berdasarkan kriteria berikut:

- Kejelasan gambar setelah proses enkripsi dan dekripsi
- Lama eksekusi pada masing-masing proses dekripsi dan enkripsi
- Kerahasiaan *share*

B. Hasil

Pengujian dilakukan pada gambar sebagaimana terlampir pada Gambar 1 yang memiliki resolusi 637x632. Gambar yang diujikan merupakan gambar dalam berkas PNG dengan skema warna 24-bit RGB. Skenario pengujian yang dilakukan adalah dengan mengenkripsi gambar dan mendekripsinya secara langsung serta menulis berkas gambar hasil enkripsi-dekripsi ke berkas PNG.

Berikut ini hasil pengujian sesuai dengan skenario-skenario yang diajukan sesuai dengan Tabel 1 dengan gambar terlampir di lampiran:

TABLE I. Pengujian Enkripsi Gambar

No	Daftar Gambar				
	Gambar Semula	***Algoritma Enkripsi***	***Jumlah Share***	***Hasil***	***Lama Eksekusi (detik)***
1	Gambar 1	*Inverse Multiply*	2	Gambar 2, Gambar 3	0.0599
2	Gambar 1	*Inverse Multiply*	3	Gambar 5, Gambar 6, Gambar 7	0.0493
3	Gambar 1	*Inverse Multiply*	4	Gambar 9, Gambar 10, Gambar 11, Gambar 12	0.0569
4	Gambar 1	*Inverse Screen*	2	Gambar 14, Gambar 15	0.0391
5	Gambar 1	*Inverse Screen*	3	Gambar 17, Gambar 18, Gambar 19	0.0605
6	Gambar 1	*Inverse Screen*	4	Gambar 21, Gambar 22, Gambar 23, Gambar 24	0.0775
7	Gambar 1	*Inverse Overlay*	2	Gambar 26, Gambar 27	0.0466
8	Gambar 1	*Inverse Overlay*	3	Gambar 29, Gambar 30, Gambar 31	0.0858
9	Gambar 1	*Inverse Overlay*	4	Gamber 33, Gambar 34, Gambar 35, Gambar 36	0.1123
10	Gambar 1	*XOR*	2	Gambar 38, Gambar 39	0.0126
11	Gambar 1	*XOR*	3	Gambar 41, Gambar 42, Gambar 43	0.0198
12	Gambar 1	*XOR*	4	Gambar 45, Gambar 46, Gambar 47, Gambar 48	0.0220

TABLE II. PENGUJIAN DEKRIPSI GAMBAR

No	Daftar Gambar					
	Gambar Semula	***Algoritma Dekripsi***	***Jumlah Share***	***Hasil***	***Lama Eksekusi (detik)***	***Kejelasan Gambar***
1	Gambar 2, Gambar 3	*Multiply*	2	Gambar 4	0.0136	Ada banyak *noise* dan lebih gelap
2	Gambar 5, Gambar 6, Gambar 7	*Multiply*	3	Gambar 8	0.0145	Ada banyak *noise* dan lebih gelap
3	Gambar 9, Gambar 10, Gambar 11, Gambar 12	*Multiply*	4	Gambar 13	0.0166	Ada banyak *noise* dan lebih gelap
4	Gambar 14, Gambar 15	*Screen*	2	Gambar 16	0.0519	Ada banyak *noise* dan lebih terang
5	Gambar 17, Gambar 18, Gambar 19	*Screen*	3	Gambar 20	0.0886	Ada banyak *noise* dan lebih terang
6	Gambar 21, Gambar 22, Gambar 23, Gambar 24	*Screen*	4	Gambar 25	0.1301	Ada banyak *noise* dan lebih terang
7	Gambar 26, Gambar 27	*Overlay*	2	Gambar 28	0.0355	Ada banyak *noise*
8	Gambar 29, Gambar 30, Gambar 31	*Overlay*	3	Gambar 32	0.0735	Ada banyak *noise*
9	Gamber 33, Gambar 34, Gambar 35, Gambar 36	*Overlay*	4	Gambar 37	0.0871	Ada banyak *noise*
10	Gambar 38, Gambar 39	*XOR*	2	Gambar 40	0.0064	Tidak ada *noise*
11	Gambar 41, Gambar 42, Gambar 43	*XOR*	3	Gambar 44	0.0166	Tidak ada *noise*
12	Gambar 45, Gambar 46, Gambar 47, Gambar 48	*XOR*	4	Gambar 49	0.0116	Tidak ada *noise*

C. *Analisis Hasil*

Berdasarkan pengujian yang dilakukan, algoritma *algoritma* memakan waktu paling lama untuk melakukan enkripsi-dekripsi pada gambar, sementara algoritma *multiply* menghabiskan waktu yang relatif lebih cepat dengan waktu enkripsi rata-rata sekitar 0.05 detik dan waktu dekripsi rata-rata sekitar 0.01 detik. Sementara itu, algoritma *overlay* memiliki rata-rata eksekusi waktu yang berada pada pertengahan algoritma *screen* dan *multiply*.

Namun, dari tiga algoritma yang diimplementasikan, semuanya memiliki waktu eksekusi yang lebih panjang daripada algoritma XOR dengan durasi paling tinggi 0.022 detik pada enkripsi dan paling lama 0.0166 detik untuk dekripsi.

Berdasarkan percobaan yang dilakukan, algoritma *screen* dan *overlay* memiliki tren yang meningkat dengan bertambahnya jumlah *share* yang dihasilkan saat enkripsi dan digunakan saat dekripsi. Sementara itu, algoritma *multiply* tidak terpengaruh oleh peningkatan jumlah *share* dengan waktu eksekusi relatif stagnan, sekitar 0.05 detik pada enkripsi dan 0.01 detik pada dekripsi.

Ketiga algoritma yang diiimplementasikan memiliki hasil dekripsi kembali yang tidak sepenuhnya sesuai dengan hasil semula. Baik algoritma *multiply*, *screen*, dan *overlay* menghasilkan gambar dekripsi yang memiliki banyak *noise* dengan *tone* warna yang berbeda. Algoritma *multiply* menghasilkan gambar dekripsi kembali yang cenderung lebih gelap. Algoritma *screen* menghasilkan gambar dekripsi kembali yang cenderung lebih terang. Sementara itu, algoritma *overlay* menghasilkan gambar dekripsi kembali yang lebih keabu-abuan. Selain itu, *share* terakhir dari masing-masing algoritma masih memiliki makna yang mewakili gambar yang dienkripsi sehingga tidak cukup aman untuk mengamankan gambar rahasia. Dengan demikian, ketiga algoritma tersebut tidak seefektif dan tidak seefisien algoritma XOR yang dijadikan variabel kontrol pada percobaan.

V. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan percobaan yang dilakukan, terdapat beberapa kesimpulan yang dapat ditarik dari percobaan ini:

- Algoritma *image blending* menghasilkan gambar dekripsi kembali yang tidak sesuai dan tidak sejelas algoritma XOR.
- Algoritma *image blending* menghasilkan satu *share* yang tidak cukup rahasia dari sekian *share* yang dibuat dengan menggambarkan gambar aslinya.
- Algoritma *image blending* tidak seefisien algoritma XOR dengan waktu eksekusi yang relatif lebih lama.

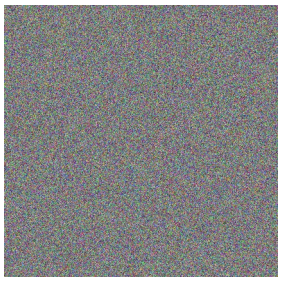
B. *Saran*

Berikut ini saran yang dapat penulis ajukan:

- Sebaiknya percobaan kriptografi visual dilakukan dengan algoritma XOR atau algoritma lainnya yang lebih efektif dalam mengenkripsi-dekripsi gambar rahasia
- Baik proses enkripsi maupun dekripsi harus memiliki waktu eksekusi yang sekecil mungkin karena berkaitan dengan operasi pada matriks berdimensi tiga yang cukup besar.

LAMPIRAN

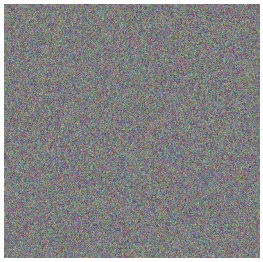
Gambar 1: Gambar yang digunakan untuk pengujian



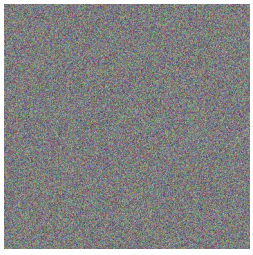
Gambar 2: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (2,2)

Gambar 3: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (2,2)

Gambar 4: Gambar hasil dekripsi Gambar 2 dan Gambar 3 dengan algoritma *multiply* pada skema (2,2)

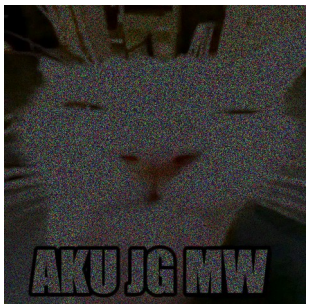


Gambar 5: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (3,3)

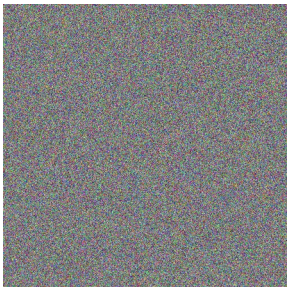


Gambar 6: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (3,3)

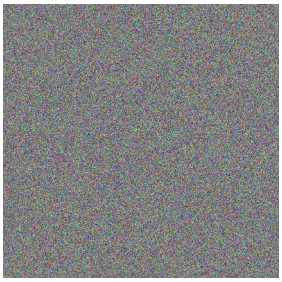
Gambar 7: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (3,3)



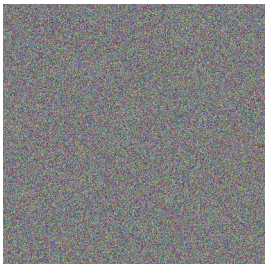
Gambar 8: Gambar hasil dekripsi Gambar 5, Gambar 6, dan Gambar 7 dengan algoritma *multiply* pada skema (3,3)



Gambar 9: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (4,4)

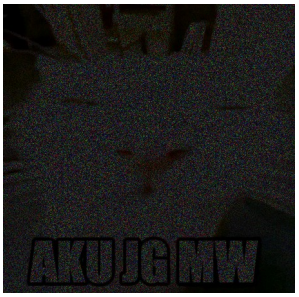


Gambar 10: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (4,4)

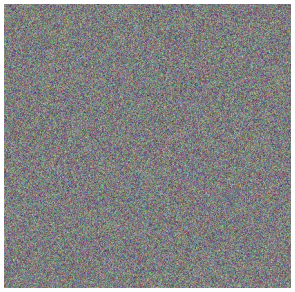


Gambar 11: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (4,4)

Gambar 12: Share keempat hasil enkripsi Gambar 1 dengan algoritma *inverse multiply* pada skema (4,4)



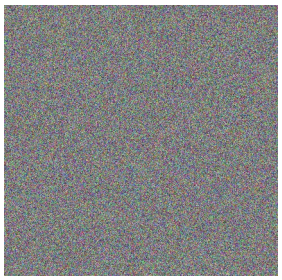
Gambar 13: Gambar hasil dekripsi Gambar 9, Gambar 10, Gambar 11, dan Gambar 12 dengan algoritma *multiply* pada skema (4,4)



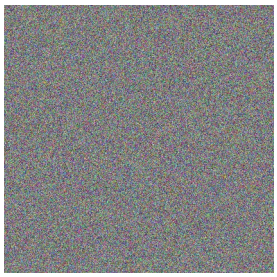
Gambar 14: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (2,2)

Gambar 15: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (2,2)

Gambar 16: Gambar hasil dekripsi Gambar 14 dan Gambar 15 dengan algoritma *screen* pada skema (2,2)



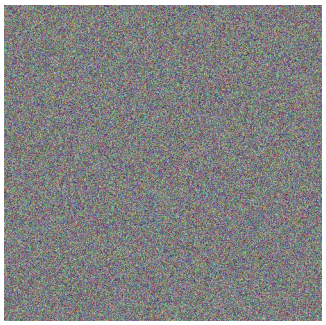
Gambar 17: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (3,3)



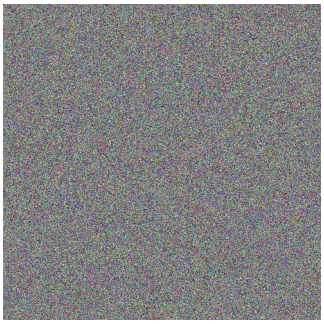
Gambar 18: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (3,3)

Gambar 19: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (3,3)

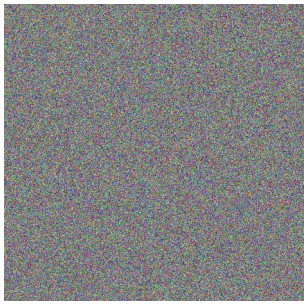
Gambar 20: Gambar hasil dekripsi Gambar 17, Gambar 18, dan Gambar 19 dengan algoritma *screen* pada skema (3,3)



Gambar 21: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (4,4)



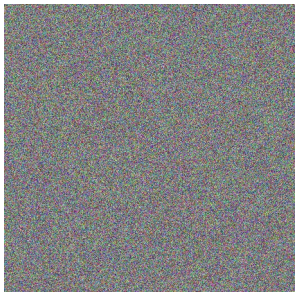
Gambar 22: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (4,4)



Gambar 23: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (4,4)

Gambar 24: Share keempat hasil enkripsi Gambar 1 dengan algoritma *inverse screen* pada skema (4,4)

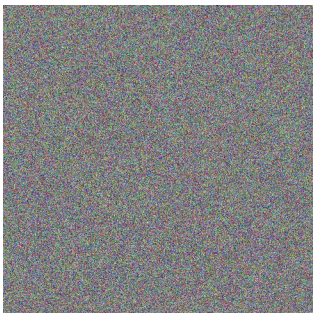
Gambar 25: Gambar hasil dekripsi Gambar 21, Gambar 22, Gambar 23, dan Gambar 24 dengan algoritma *screen* pada skema (4,4)



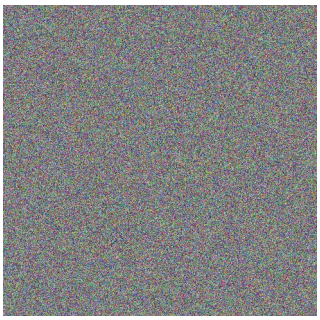
Gambar 26: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (2,2)

Gambar 27: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (2,2)

Gambar 28: Gambar hasil dekripsi Gambar 26 dan Gambar 27 dengan algoritma *overlay* pada skema (2,2)



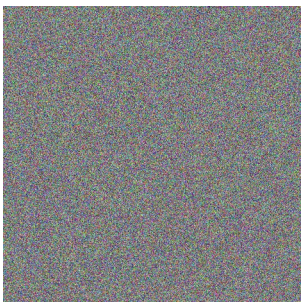
Gambar 29: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (3,3)



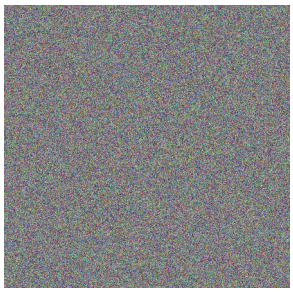
Gambar 30: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (3,3)

Gambar 31: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (3,3)

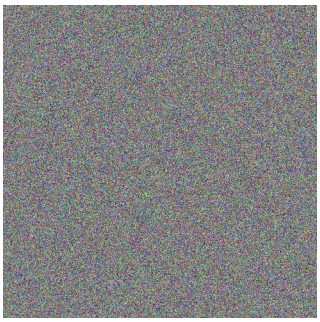
Gambar 32: Gambar hasil dekripsi Gambar 29, Gambar 30, dan Gambar 31 dengan algoritma *overlay* pada skema (3,3)



Gambar 33: Share pertama hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (4,4)



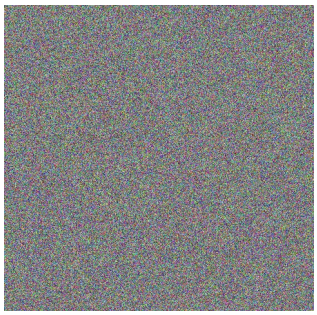
Gambar 34: Share kedua hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (4,4)



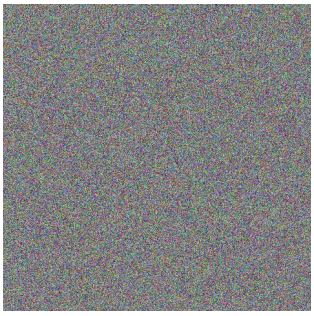
Gambar 35: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (4,4)

Gambar 36: Share keempat hasil enkripsi Gambar 1 dengan algoritma *inverse overlay* pada skema (4,4)

Gambar 37: Gambar hasil dekripsi Gambar 33, Gambar 34, Gambar 35, dan Gambar 36 dengan algoritma *overlay* pada skema (4,4)

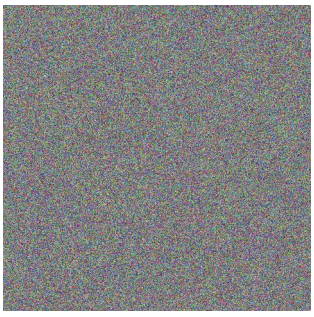


Gambar 38: Share pertama hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (2,2)

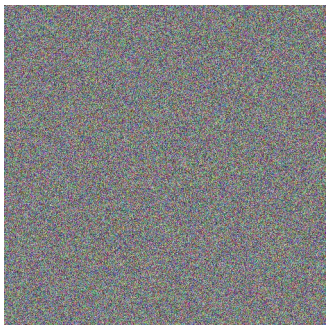


Gambar 39: Share kedua hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (2,2)

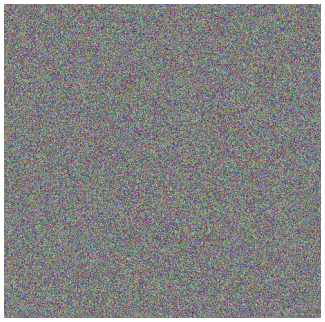
Gambar 40: Gambar hasil dekripsi Gambar 38 dan Gambar 39 dengan algoritma *XOR* pada skema (2,2)



Gambar 41: Share pertama hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (3,3)

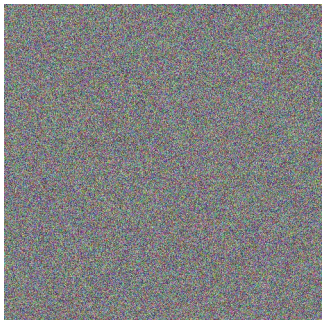


Gambar 42: Share kedua hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (3,3)

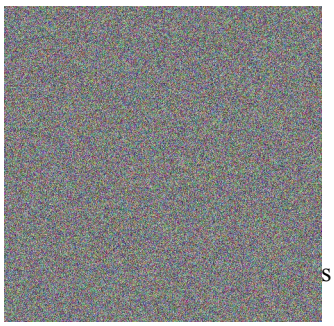


Gambar 43: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (3,3)

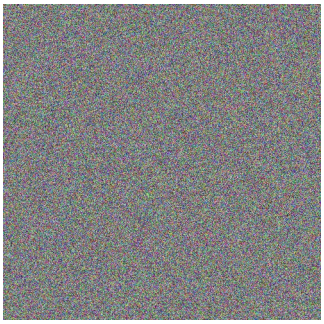
Gambar 44: Gambar hasil dekripsi Gambar 41, Gambar 42, dan Gambar 43 dengan algoritma *XOR* pada skema (3,3)



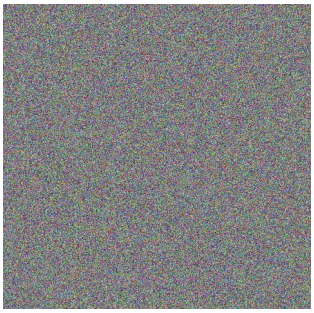
Gambar 45: Share pertama hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (4,4)



Gambar 46: Share kedua hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (4,4)



Gambar 47: Share ketiga hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (4,4)



Gambar 48: Share keempat hasil enkripsi Gambar 1 dengan algoritma *XOR* pada skema (4,4)

Gambar 49: Gambar hasil dekripsi Gambar 45, Gambar 46, Gambar 47, dan Gambar 48 dengan algoritma *XOR* pada skema (4,4)

REFERENCES

[1] R. Munir, *Kriptografi Visual, Teori dan Aplikasinya Bagian 1*. 2024. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2023-2024/36-Kriptografi-Visual-Bagian1-2024.pdf

[2] A. Pandey and S. Som, "Applications and usage of visual cryptography: A review," *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Noida, India, 2016, pp. 375-381, doi: 10.1109/ICRITO.2016.7784984.

[3] M. Naor and A. Shamir, "Visual cryptography," in Lecture notes in computer science, 1995, pp. 1–12. doi: 10.1007/bfb0053419.

[4] Krita. "Mix." https://docs.krita.org/en/reference_manual/blending_modes/mix.html

[5] J. Gruschel, "Overlay mode," Mar. 01, 2006. https://www.pegtop.net/delphi/articles/blendmodes/overlay.htm

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024

Ttd

Andika Naufal Hilmy, 13520098