

# OTENTIKASI IMAGE BERBASIS MAGIC SQUARE ORDE $n$

## *(Image Authentication Based on Magic Square Order $n$ )*

**Kiswara Agung Santoso**

Jurusan Matematika, Fakultas MIPA, Universitas Jember  
Jl. Kalimantan 37 Jember 68121, Jember  
e-mail: kiswara.fmipa@unej.ac.id

**Abstract.** Image is a digital media that is easy to change, so it is susceptible to being used for crime. Image changes may be affected by the unstable internet during transmission or deliberate manipulation of images for specific purposes. Hence, we need a tool to determine the authenticity of the image. One strategy that can be used is to insert code into the entire image, so that any changes to the image information can be detected. The insertion process performed by dividing the image into several blocks, where the size of each block is adjusted according to the size of the magic square order 3. For the authentication process, the image that has been manipulated will be counted for each pixel. The results show that the manipulated image can be detected. The parts of the image that have changed can be identified because the pixel value doesn't satisfy the magic square rule.

**Keywords:** Authentication, Image, Magic Square, Pixel

## 1. Pendahuluan

Penelitian sebelumnya tentang autentikasi image telah dilakukan oleh [16] dengan menerapkan Teknik kriptografi. Di dalam kriptografi, terdapat fungsi hash yang dapat digunakan untuk mengambil sidik jari dalam image. Pendeteksian dilakukan dengan membandingkan nilai hash pada image asli dan image manipulasi. Hasil yang diperoleh hanya berfokus pada mendeteksi apakah image telah mengalami perubahan atau tidak, namun secara rinci tidak menjelaskan bagian image mana yang telah berubah.

Steganografi merupakan seni menyembunyikan kode rahasia ke dalam sebuah media digital, sehingga keberadaannya tidak terdeteksi [5]. Kode rahasia dapat berupa teks biasa, teks sandi, image, atau apapun yang dapat direpresentasikan dalam bentuk bit [10]. Teknik steganografi yang paling sederhana yaitu mengganti bit LSB dari masing-masing piksel dalam image dengan kode yang akan disisipkan. Perubahan bit LSB tidak berpengaruh secara signifikan pada warna image, sehingga akan sulit diketahui oleh mata manusia.

Berdasarkan latar belakang di atas, penulis akan mengusulkan teknik baru yaitu teknik steganografi berbasis magic square orde  $n$  untuk mengetahui autentisitas image dan mengidentifikasi bagian image yang sudah mengalami perubahan. Magic square merupakan susunan bilangan acak di dalam sel, di mana jumlah tiap baris, kolom, dan

diagonalnya adalah sama [8]. Pendeteksian akan dilakukan dengan membandingkan nilai LSB piksel dalam image dengan elemen magic square yang bersesuaian. Apabila nilainya tidak sama, maka piksel berubah menjadi warna putih dan image dapat dikatakan tidak autentik.

## 2. Studi Literatur

### 2.1 Magic square

Magic square adalah susunan bilangan acak di dalam sel, di mana jumlah tiap baris, kolom, dan diagonalnya adalah sama [8]. Magic square orde  $n$  disusun oleh bilangan asli berurutan yang berbeda dari 1 sampai  $n$  dengan setiap bilangan muncul tepat 1 kali. Jumlah konstanta di setiap baris, setiap kolom, dan setiap diagonalnya disebut Konstanta Magic [17]. Konstanta magic dapat dihitung dengan Persamaan (1) berikut.

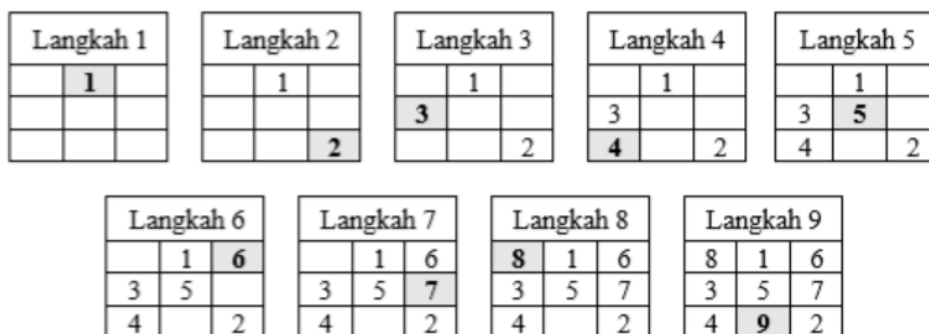
$$M_n = \frac{n(n^2+1)}{2} \quad (1)$$

Magic square dapat diklasifikasikan ke dalam tiga kategori, yaitu [8]:

#### a. Magic square orde ganjil

Prosedur pembangunan untuk magic square orde ganjil sangat sederhana. Prosedur ini dikenal dengan beberapa nama, seperti algoritma de la Loubere, metode tangga, dan metode siamese. Langkah awal yang harus dilakukan yaitu menempatkan angka 1 tepat di tengah baris atas. Kemudian, tulis bilangan bulat berturut-turut dengan jalur kanan atas (dalam arah timur laut), selangkah demi selangkah dengan kasus khusus berikut.

- Jika gerakan ke kanan atas menghasilkan lokasi di luar batas kotak, maka tempatkan angka baru di ujung berlawanan dari baris atau kolom yang akan berisi angka baru (jika baris dan kolom tidak dibatasi).
- Jika angka saat ini berada di pojok paling atas atau paling kanan, maka tempatkan angka berikutnya di pojok kiri paling bawah.
- Jika kotak sudah ditempati oleh sebuah angka, tempatkan angka baru tepat di bawah angka sebelumnya dan lanjutkan. Misalkan, prosedur di atas diilustrasikan pada pembangunan magic square orde 3 seperti Gambar 1 berikut.



Gambar 1. Pembuatan *Magic Square* Orde 3

Sehingga, terbentuklah magic square  $3 \times 3$  yang baris, kolom, dan diagonalnya jika

dijumlahkan sama dengan 15.

b. Magic square orde genap (habis dibagi 4)

Magic square di mana  $n$  berupa kelipatan genap yang habis dibagi 4 (yaitu 4, 8, 12, ...).

Untuk membangun magic square orde ini dapat mengikuti prosedur berikut.

- Bagi kotak menjadi 4 subkotak
- Gambar dua garis diagonal utama melalui setiap subkotak
- Selanjutnya, isikan seluruh kotak secara berurutan mulai dari kiri atas ke kanan bawah dengan angka 1 sampai  $n$  di setiap tempat yang kosong.
- Kemudian, hapus garis dan isikan angka yang belum dimasukkan dari kiri atas ke kanan bawah mulai dari  $n$  sampai 1.

Misalkan, prosedur di atas diilustrasikan pada pembangunan magic square orde 4 seperti Gambar 2 berikut.



Gambar 2. Pembuatan *Magic Square* Orde 4

Sehingga, terbentuklah magic square  $4 \times 4$  yang baris, kolom, dan diagonalnya jika dijumlahkan sama dengan 34.

## 2.2 Steganografi

Steganografi berasal dari bahasa Yunani yaitu “steganos” yang berarti “tersembunyi atau terselubung” dan “graphein” yang berarti “tulisan”. Dengan kata lain, steganografi adalah ilmu menyembunyikan informasi dengan suatu cara sehingga selain pengirim dan penerima tidak ada yang mengetahui dan menyadari adanya informasi rahasia yang disisipkan pada sebuah media [11]. Menurut [5], steganografi didefinisikan sebagai Teknik menyembunyikan pesan rahasia dalam media digital, sedemikian sehingga keberadaan pesan tersebut tidak terdeteksi oleh indera manusia.

Teknik steganografi digunakan untuk menyembunyikan informasi dengan cara mencegah terdeteksinya informasi yang disembunyikan melalui berbagai media, seperti image, audio, video dan teks. Adapun proses penyembunyian informasi rahasia dalam teknik steganografi diawali dengan mengidentifikasi bit bit redundant dari citra medianya (yang dapat dimodifikasi tanpa merusak integritas citra yang bersangkutan). Proses penyisipan menghasilkan suatu stego citra melalui penggantian bit redundant dengan data informasi rahasia yang disembunyikan tersebut [15].

Metode LSB (Least Significant Bit) merupakan metode paling sederhana dalam steganografi. Metode ini bekerja dengan cara mengganti bit LSB dari masing-masing piksel dengan informasi yang akan disisipkan [10]. Metode LSB memanfaatkan

kelemahan indra visual manusia dalam mengamati perubahan sedikit pada image. Misalkan, ambil 3 piksel dari image warna sebagai berikut [11].

$$\begin{pmatrix} 39 & 233 & 200 \\ 39 & 200 & 233 \\ 200 & 39 & 233 \end{pmatrix} = \begin{bmatrix} 00100111 & 11101001 & 11001000 \\ 00100111 & 11001000 & 11101001 \\ 11001000 & 00100111 & 11101001 \end{bmatrix}$$

Gambar 3. Konversi bilangan desimal menjadi bit

Kemudian huruf 'A' dengan nilai biner 10000011 akan disisipkan pada setiap bit terakhir dari image. Sehingga, diperoleh

$$\begin{bmatrix} 00100111 & 11101000 & 11001000 \\ 00100110 & 11001000 & 11101000 \\ 11001001 & 00100111 & 11101001 \end{bmatrix}$$

Gambar 4. Hasil penyisipan pada bit terakhir (LSB)

Dari hasil di atas dapat diketahui bahwa bit yang benar-benar berubah hanya ada 4 bit. Secara keseluruhan, penyisipan ini hanya mengubah sedikit nilai piksel dan perubahan intensitas warna sangat kecil. Pada image 24-bit, komponen bit warna merah, biru, dan hijau dapat digunakan karena masing-masing diwakili oleh byte. Sehingga, kapasitas muatan informasi yang disisipkan bisa mencapai 3 kali jumlah pada image [13]. Karena ada 256 kemungkinan intensitas dari setiap warna, mengubah LSB piksel akan menghasilkan perubahan kecil pada intensitas warna. Perubahan ini tidak dapat terdeteksi oleh mata manusia dan informasi dapat disembunyikan secara efektif.

### 3. Metode Penelitian

#### 3.1 Data Penelitian

Data yang digunakan dalam penelitian adalah enam buah image yang terdiri dari tiga image asli dan tiga image manipulasi. Image asli merupakan image yang akan disisipi magic square orde  $n$  untuk keamanannya. Sedangkan image manipulasi merupakan image yang sudah diedit dengan aplikasi Adobe Photoshop CS4 untuk dideteksi autentisitasnya.

#### 3.2 Langkah-langkah Penelitian

Langkah-langkah yang akan dilakukan pada penelitian ini adalah sebagai berikut.

##### a. Studi Literatur

Langkah awal penelitian ini yaitu mengumpulkan berbagai literatur tentang citra digital, autentikasi, steganografi, metode LSB, dan magic square.

##### b. Perhitungan Manual

Pada langkah ini diberikan contoh perhitungan manual untuk proses penyisipan dan proses autentikasi image berbasis magic square orde  $n$ . Hasil yang diperoleh dapat digunakan untuk pembuatan program.

c. Pembuatan Program

Pembuatan program dilakukan pada software MATLAB R2015a. Pada langkah ini, akan dibuat program penyisipan dan autentikasi image berbasis magic square orde  $n$ .

1) Proses Penyisipan

- Masukkan image asli.
- Masukkan nilai  $n$  yang dipilih dan bangkitkan magic square orde  $n$ .
- Bagi image menjadi beberapa blok, di mana setiap blok berukuran  $n \times n$ .
- Konversi nilai piksel pada image menjadi bilangan biner (bit) dan ambil bit LSB-nya. Untuk setiap blok, ganti bit LSB pada image dengan elemen magic square yang bersesuaian.

2) Proses Autentikasi

- Masukkan image manipulasi.
- Masukkan nilai  $n$  yang sudah diketahui dan bangkitkan magic square orde  $n$ .
- Bagi image menjadi beberapa blok, di mana setiap blok berukuran  $n \times n$ .
- Konversi nilai piksel pada image menjadi bit dan ambil bit LSB-nya
- Untuk setiap blok, bandingkan bit LSB yang diperoleh dengan elemen magic square yang bersesuaian. Jika nilainya sama, maka piksel tetap dan jika nilainya tidak sama, maka piksel berubah menjadi warna putih

d. Simulasi Program

Pada tahap ini, program yang telah dibuat diterapkan pada image asli untuk proses penyisipan. Sedangkan untuk proses autentikasi dilakukan pada image manipulasi.

e. Analisis Hasil

Setelah program dijalankan, hasil yang diperoleh akan dianalisis. Tahap ini dilakukan untuk mengetahui keberhasilan dalam melakukan autentikasi image berbasis magic square orde  $n$ .

f. Kesimpulan

Penarikan kesimpulan dilakukan dengan menjawab tujuan penelitian yang telah dibuat dan saran-saran untuk perbaikan bagi penelitian selanjutnya.

## 4. Hasil dan Pembahasan

Untuk mengetahui proses penyisipan dan autentikasi image menggunakan teknik steganografi berbasis magic square orde  $n$ , maka dilakukan perhitungan secara manual.

### a. Proses Penyisipan

Proses penyisipan dilakukan dengan metode LSB, di mana 4 bit terakhir setiap piksel diganti dengan elemen magic square orde  $n$  yang bersesuaian. Adapun nilai  $n$  yang

akan digunakan adalah  $n = 3, 4, \dots, 31$ . Setiap satu digit elemen magic square akan disisipkan ke dalam satu layer warna. Sehingga, untuk  $n > 31$  tidak dapat digunakan karena elemen terbesarnya mengandung 4 digit angka, sedangkan setiap piksel dalam image hanya memiliki 3 layer warna. Berikut adalah aturan penyisipan dalam image.

1. Untuk  $n = 3$

Magic square orde 3 tersusun atas bilangan asli dengan elemen terbesarnya adalah angka 9. Karena setiap elemennya hanya mengandung satu angka saja, maka penyisipan dilakukan pada layer merah setiap piksel dalam image. Misalkan ambil satu nilai piksel pada image di layer merah.  $f(200,300) = (192)$

Kemudian, bangkitkan magic square orde 3 yaitu:  $\begin{pmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{pmatrix}$

Hitung elemen magic square yang akan disisipkan dengan modulo  $n$  yang dipilih. Karena titik yang dipilih (200,300), maka  $200 \bmod 3 = 2$  dan  $300 \bmod 3 = 0$ . Jadi, nilai LSB piksel ke-(200,300) diganti dengan elemen magic square ke(2,3) yaitu 7. Proses penyisipan dapat dilihat seperti pada Gambar 5 berikut.

$$\begin{array}{c} (192) = [1100 \mathbf{0000}] \\ \downarrow \text{ganti} \\ (7) = [0111] \xrightarrow{\text{menjadi}} [1100 \mathbf{0111}] = (199) \end{array}$$

Gambar 5. Proses penyisipan untuk  $n=3$

2. Untuk  $4 \leq n \leq 9$

Magic square orde 4, 5, 6, 7, 8, dan 9 tersusun atas bilangan asli dengan masing-masing elemen terbesarnya adalah 16, 25, 36, 49, 64, dan 81. Karena elemen-elemen tersebut mengandung dua angka, maka penyisipan dilakukan pada dua layer piksel dalam image. Nilai puluhan disisipkan pada layer merah dan nilai satuan disisipkan pada layer hijau.

Misalkan ambil satu nilai piksel pada koordinat (402,118) di layer merah adalah 69 dan layer hijau adalah 73.

Kemudian, bangkitkan magic square orde 4 yaitu:  $\begin{pmatrix} 16 & 02 & 03 & 13 \\ 05 & 11 & 10 & 08 \\ 09 & 07 & 06 & 12 \\ 04 & 14 & 15 & 01 \end{pmatrix}$

Hitung elemen magic square yang akan disisipkan dengan modulo  $n$  yang dipilih. Karena titik yang dipilih (402,118), maka  $402 \bmod 4 = 2$  dan  $118 \bmod 4 = 2$ . Jadi, nilai LSB piksel ke-(402,118) diganti dengan elemen magic square ke (2,2) yaitu 11. Proses penyisipan dapat dilihat seperti pada Gambar 6 berikut.

$$\begin{pmatrix} 69 \\ 73 \end{pmatrix} = \begin{bmatrix} 0100 & 0101 \\ 0100 & 1001 \end{bmatrix}$$

↓ ganti

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{bmatrix} 0001 \\ 0001 \end{bmatrix} \xrightarrow{\text{menjadi}} \begin{bmatrix} 0100 & 0001 \\ 0100 & 0001 \end{bmatrix} = \begin{pmatrix} 65 \\ 65 \end{pmatrix}$$

Gambar 6. Proses penyisipan untuk  $n=4$

3. Untuk  $10 \leq n \leq 31$

Magic square orde 10, 11, ... , 31 tersusun atas bilangan asli dengan elemen terbesarnya adalah 100, 121, ... . , 961. Karena elemen-elemen tersebut mengandung tiga angka, maka penyisipan dilakukan pada tiga layer piksel dalam image. Nilai ratusan disisipkan pada layer merah, nilai ribuan disisipkan pada layer hijau, dan nilai satuan disisipkan pada layer biru.

Misalkan ambil satu nilai piksel pada posisi (264,118), dimana nilai piksel pada layer merah bernilai 170, pada layer hijau bernilai 174, dan piksel pada layer biru bernilai 186. Kemudian bangkitkan magic square orde 10 seperti gambar 7 berikut:

092	099	001	008	015	067	074	051	058	040
098	080	007	014	016	073	055	057	064	041
004	081	088	020	022	054	056	063	070	047
085	087	019	021	003	060	062	069	071	028
086	093	025	002	009	061	068	075	052	034
017	024	076	083	090	042	049	026	033	065
023	005	082	089	091	048	030	032	039	066
079	006	013	095	097	029	031	038	045	072
010	012	094	096	078	035	037	044	046	053
011	018	100	077	084	036	043	050	027	059

Gambar 7. Magic square orde 10

Hitung elemen magic square yang akan disisipkan dengan modulo  $n$  yang dipilih. Jika titik yang dipilih (264,118), maka  $264 \bmod 10 = 4$  dan  $118 \bmod 10 = 8$ . Jadi, nilai LSB piksel ke-(264,118) diganti dengan elemen magic square ke(4,8) yaitu 069. Proses penyisipan dapat dilihat seperti pada Gambar 8 berikut.

$$\begin{pmatrix} 170 \\ 174 \\ 186 \end{pmatrix} = \begin{bmatrix} 1010 & 1010 \\ 1010 & 1110 \\ 1011 & 1010 \end{bmatrix}$$

↓ ganti

$$\begin{pmatrix} 0 \\ 6 \\ 9 \end{pmatrix} = \begin{bmatrix} 0000 \\ 0110 \\ 1001 \end{bmatrix} \xrightarrow{\text{menjadi}} \begin{bmatrix} 1010 & 0000 \\ 1010 & 0110 \\ 1011 & 1001 \end{bmatrix} = \begin{pmatrix} 160 \\ 166 \\ 185 \end{pmatrix}$$

Gambar 8. Proses penyisipan untuk  $n=10$

## b. Proses Autentikasi

Perubahan suatu image dapat dilihat dari perubahan nilai piksel yang terjadi dalam image. Apabila nilai pikselnya berbeda dengan image asli, maka image tersebut telah dimanipulasi. Oleh karena itu, proses autentikasi dapat dilakukan dengan menghitung nilai piksel dalam image yang memenuhi aturan magic square orde  $n$ . Setiap piksel diambil bit LSB-nya dan dibandingkan dengan elemen magic square yang bersesuaian. Apabila nilainya tidak sama, maka piksel diubah menjadi warna putih.

Misalkan sebuah image manipulasi akan dideteksi perubahannya dan diketahui image asli telah diamankan dengan magic square orde 3. Pertama, ambil nilai piksel  $3 \times 3$  pada image manipulasi dan konversi nilainya menjadi bit seperti Gambar 9. Karena  $n = 3$ , maka nilai piksel yang dihitung hanya pada layer merah saja.

$$\begin{pmatrix} 184 & 71 & 17 \\ 164 & 14 & 71 \\ 110 & 4 & 132 \end{pmatrix} = \begin{bmatrix} 1011 & 1000 & 0100 & 0111 & 0001 & 0001 \\ 1010 & 0100 & 0000 & 1110 & 0100 & 0111 \\ 0110 & 1110 & 0000 & 0100 & 1000 & 0100 \end{bmatrix}$$

Gambar 9. Konversi nilai piksel kedalam bit

Kemudian, ambil bit LSB masing-masing piksel dan konversi nilainya ke dalam desimal (lihat Gambar 10).

$$\begin{bmatrix} 1000 & 0111 & 0001 \\ 0100 & 1110 & 0111 \\ 1110 & 0100 & 0100 \end{bmatrix} = \begin{pmatrix} 8 & 7 & 1 \\ 4 & 14 & 7 \\ 14 & 4 & 4 \end{pmatrix}$$

Gambar 10. Konversi LSB menjadi desimal

Selanjutnya, bandingkan elemen yang bersesuaian dengan nilai bit LSB yang didapat dengan elemen dari magic square orde 3. Apabila nilainya tidak sama, maka dicetak tebal, seperti gambar 11 berikut.

$$\begin{pmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{pmatrix} \approx \begin{pmatrix} 8 & 7 & 1 \\ \mathbf{4} & \mathbf{14} & 7 \\ \mathbf{14} & 4 & 4 \end{pmatrix}$$

Gambar 11. Perbandingan elemen magic square dengan bit LSB piksel

Terakhir, ubah nilai piksel yang tidak memenuhi aturan magic square menjadi 255. Jadi, selain  $f(1,1)$  dan  $f(2,3)$  nilai pikselnya berubah. Perubahan dapat dilihat pada Gambar 12 berikut.

$$\begin{pmatrix} 184 & 71 & 17 \\ 164 & 14 & 71 \\ 110 & 4 & 132 \end{pmatrix} \xrightarrow{\text{menjadi}} \begin{pmatrix} 184 & 255 & 255 \\ 255 & 255 & 71 \\ 255 & 255 & 255 \end{pmatrix}$$

Gambar 12. Perubahan nilai piksel yang tidak memenuhi aturan magic square

## c. Simulasi Program

Pada dasarnya, magic square dapat dibangun dengan beberapa cara sehingga menghasilkan bentuk yang berbeda-beda untuk setiap  $n$ . Misalkan, untuk magic square



orde 3 yang memiliki delapan bentuk magic square (lihat Gambar 13). Masing-masing bentuk memenuhi syarat magic square yaitu apabila setiap baris, setiap kolom, dan dua diagonal utamanya dijumlahkan menghasilkan nilai yang sama dengan 15. Namun, di dalam matlab terdapat fungsi “magic(n)” sebagai pembangkit magic square berukuran  $n \times n$ . Fungsi ini hanya dapat menghasilkan satu bentuk tertentu dari setiap magic square orde  $n$ .

$$\begin{pmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{pmatrix} \quad
 \begin{pmatrix} 8 & 3 & 4 \\ 1 & 5 & 9 \\ 6 & 7 & 2 \end{pmatrix} \quad
 \begin{pmatrix} 6 & 7 & 2 \\ 1 & 5 & 9 \\ 8 & 3 & 4 \end{pmatrix} \quad
 \begin{pmatrix} 4 & 9 & 2 \\ 3 & 5 & 7 \\ 8 & 1 & 6 \end{pmatrix} \quad
 \begin{pmatrix} 2 & 9 & 4 \\ 7 & 5 & 3 \\ 6 & 1 & 8 \end{pmatrix} \\
 \begin{pmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{pmatrix} \quad
 \begin{pmatrix} 4 & 3 & 8 \\ 9 & 5 & 1 \\ 2 & 7 & 6 \end{pmatrix} \quad
 \begin{pmatrix} 6 & 1 & 8 \\ 7 & 5 & 3 \\ 2 & 9 & 4 \end{pmatrix}$$

Gambar 12. Delapan bentuk magic square orde 3

Penelitian ini dibagi menjadi dua buah program, yaitu program penyisipan dan program autentikasi. Program penyisipan digunakan untuk mengamankan image dengan cara menyisipkan magic square orde  $n$  ke dalam image. Sedangkan 19 program autentikasi digunakan untuk mendeteksi autentisitas image yang sudah dimanipulasi.

Program penyisipan dimulai dengan memasukkan image asli yang sudah disiapkan. Kemudian, penentuan ukuran magic square ( $n$ ) yang akan digunakan sebagai kode penyisipan. Dalam pengujian ini, image asli yang akan digunakan adalah Dokumen\_1, Dokumen\_2, dan Dokumen\_3 dengan ukuran yang sama yaitu  $678 \times 480$ . Penyisipan dilakukan dengan  $n$  yang berbeda-beda. Untuk Dokumen\_1 disisipi dengan magic square orde 3, Dokumen\_2 disisipi dengan magic square orde 4, dan Dokumen\_3 disisipi dengan magic square orde 10.

a. Percobaan pertama

Percobaan ini dilakukan pada Dokumen1\_edit di mana dokumen asli sudah disisipi kode magic square orde 3. Hasil yang diperoleh dapat dilihat pada Gambar 13 di bawah ini.



(a) Dokumen1\_edit



(b) Dokumen1\_hasil

Gambar 13. Perbandingan image asli dan image hasil penyisipan dengan  $n = 3$   
 Dokumen1\_edit merupakan Dokumen1\_sisip yang ditempel dengan tanda tangan palsu

b. Percobaan kedua

Percobaan ini dilakukan pada Dokumen2\_edit di mana dokumen asli sudah disisipi kode magic square orde 4. Hasil yang diperoleh dapat dilihat pada Gambar 14 di bawah ini.



(a) Dokumen2\_edit



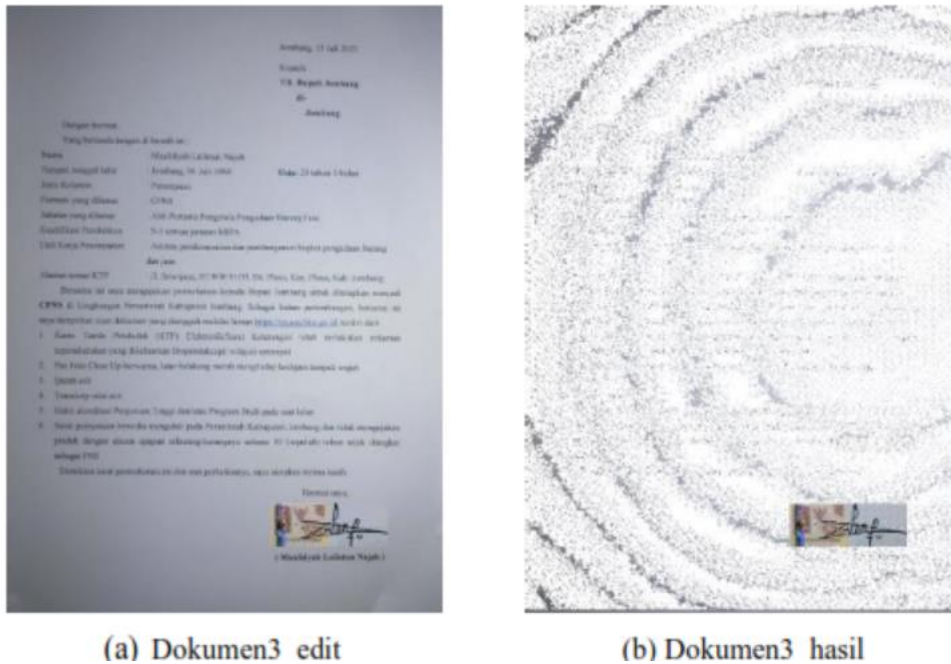
(b) Dokumen2\_hasil

Gambar 14. Hasil autentikasi percobaan kedua dengan  $n = 4$

Dokumen2\_edit merupakan Dokumen2\_sisip yang ditambahi dengan satu kalimat/catatan di bagian bawah

c. Percobaan ketiga

Percobaan ini dilakukan pada Dokumen3\_edit di mana dokumen asli sudah disisipi kode magic square orde 10. Hasil yang diperoleh dapat dilihat pada Gambar 15 di bawah ini.



Gambar 15. Hasil autentikasi percobaan ketiga dengan  $n = 10$

Dokumen3\_edit merupakan Dokumen3\_sisip yang dipotong bagian tanda tangan dan materainya saja, kemudian ditempel ke dokumen lain.

### 5. Kesimpulan

Berdasarkan hasil yang diperoleh dapat ditarik kesimpulan bahwa untuk mengamankan image dengan teknik steganografi dilakukan dengan menyisipkan magic square orde  $n$  ke dalam image. Penyisipan dimulai dengan membagi image menjadi blok-blok berukuran  $n \times n$ , kemudian mengganti empat bit LSB setiap pikselnya dengan elemen magic square orde  $n$  yang bersesuaian. Penyisipan ini digunakan sebagai identitas image, sehingga apabila image dimanipulasi oleh orang yang tidak berwenang dapat terdeteksi dengan mudah. Pendeteksian dilakukan dengan cara menghitung nilai piksel dalam image. Apabila ada nilai piksel yang tidak memenuhi aturan magic square orde  $n$ , maka image dapat dikatakan tidak autentik. Selain itu, hasil yang diperoleh juga dapat menunjukkan perbedaan image asli dengan image manipulasi.

## Daftar Pustaka

- [1] Ahmad, U. 2005. *Pengolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.
- [2] Al-Hasan, dkk. 2018. A new steganography technique using magic square and affine cipher. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*. **10(2-8)**:119-124.
- [3] Anderson, R., dan F.A.P. Petitcolas, 1998. On the limits of steganography. *Journal of IEEE*. 16(4): 474-481.
- [4] Gangwar, A. 2013. Improved RGB-LSB seganography using secret key. *International Journal of Computer Trend and Technology*. **4(2)**: 85-89.
- [5] Gonzalez, R. C., R. E. Woods, dan S. L. Eddins. 2009. *Digital Image Processing using Matlab*. USA: Gatesmark.
- [6] Gorain, G. C. 2010. *Mathematics of Magic squares*. Research Gate: J.K College.
- [7] Hajar, S. 2021. Analisa metode message digest 5 (MD5) untuk mendeteksi orisinalitas citra digital. *Jurnal Pelita Informatika*. **9(3)**: 142-148.
- [8] Johnson, N. F., dan S. Jajodia. 1998. Exploring steganography: seeing the unseen. *Journal of IEEE*. **31(2)**.
- [9] Johnson, N. F., Z. Duric, dan S. Jajodia. 2003. *Information Hiding: Steganography and Watermarking-Attacks and Countermeasures*. New York: Springer.
- [10] Kurup, S., A. Rodrigues, dan A. Bhise. 2016. A data hiding technique based on magic squares. *International Conference on Data Mining and Advanced Computing (SAPIENCE)*.
- [11] Moerland, T. 2003. *Steganography and Steganalysis*. Leiden Institute of Advanced Computing Science. <http://www.liacs.nl/home/tmoerl/privtech.pdf>. [Diakses pada 20 Oktober 2021].
- [12] Munir. 2004. *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika.
- [13] Pfitzmann, A. 1999. *Information Hiding*. Jerman: Springer.

- [14] Saputra, I., dan S. D. Nasution. 2019. Analisa algoritma SHA-256 untuk mendeteksi orisinilitas citra digital. *Proceedings of Seminar Nasional Riset Information Science*: 164-178.
- [15] Sesiano, J. 2019. *Magic square*. Geneva: Springer.
- [16] Simanullang, P.M., S. Sinurat, dan I. Saputra. 2019. Analisa metode SHA-384 untuk mendeteksi orisinalitas citra digital. *Konferensi Nasional Teknologi Informasi dan Komputer*. **3(1)**: 187- 198.
- [17] Sutoyo, T., dkk. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Andi.