Implementasi DSA dengan SHA-512 pada Citra Digital Menggunakan Metode *Reversible Image Steganography* berbasis PVO

Arildha Rahma Savitri[1,*], Nadia Paramita Retno Adiati[2)], Ari Moesriami Barmawi[3)]

1) Rekayasa Kriptografi, Kriptografi, Politeknik Siber dan Sandi Negara, arildha.rahma@student.poltekssn.ac.id
2) Rekayasa Kriptografi, Kriptografi, Politeknik Siber dan Sandi Negara, nadia.paramita@poltekssn.ac.id
3) Cyber Security & Steganography, Informatika, Telkom University, mbarmawi@melsa.net.id

Riwayat Artikel

Dikirim 5 Agu 2025
Diterima 9 Agu 2025
Diterbitkan 31 Agu 2025

Kata kunci:

Digital Signature Algorithm (DSA)
Pixel Value Ordering (PVO)
SHA-512

Abstrak

Distribusi data digital melalui media elektronik rentan terhadap berbagai ancaman keamanan. Untuk menjamin keamanan transmisi, kriptografi digunakan sebagai mekanisme proteksi, tanda tangan digital digunakan untuk memastikan integritas dan otentikasi, sementara steganografi mencegah nilai tanda tangan dikirim secara terpisah. Penelitian ini mengimplementasikan Digital Signature Algorithm (DSA) dengan SHA-512 pada citra digital menggunakan metode reversible image steganography berbasis *Pixel* Value Ordering (PVO). Keunikan penelitian ini terletak pada pemanfaatan algoritma Scale Invariant Feature Transform (SIFT) untuk mengekstraksi fitur citra sebelum proses *hash*ing, sehingga tanda tangan digital yang dihasilkan lebih tahan terhadap variasi citra. Proses meliputi pembangkitan kunci DSA, ekstraksi fitur citra, *hash*ing dengan SHA-512, penandatanganan digital, serta penyisipan tanda tangan ke dalam citra menggunakan PVO. Hasil penelitian menunjukkan bahwa DSA dan SHA-512 dapat diimplementasikan pada citra digital baik jenis *Grayscale* ataupun RGB tanpa batasan ukuran *pixel* serta menunjukkan nilai *signature* yang disembunyikan tidak menyebabkan perubahan apapun terhadap *cover-image*. Hasil evaluasi performa dan analisis uji ketahanan metode steganografi PVO pada penelitian ini menunjukkan hasil *Detection Rate*, BPP, MSE, dan PSNR yang cukup baik.

Keyword:

Digital Signature Algorithm (DSA)
Pixel Value Ordering (PVO)
SHA-512

Abstract

The distribution of digital data via electronic media is vulnerable to various security threats. Cryptography ensures secure transmission, digital signatures provide integrity and authentication, while steganography prevents signature values from being transmitted separately. This study implements the Digital Signature Algorithm (DSA) with SHA-512 on digital images using a reversible image steganography method based on Pixel Value Ordering (PVO). The novelty lies in utilizing the Scale Invariant Feature Transform (SIFT) to extract image features prior to hashing, enabling the generated signature to be more robust against image variations. The process includes DSA key generation, image feature extraction, SHA-512 hashing, digital signing, and embedding the signature into the image using PVO. Experimental results demonstrate that the method works effectively for both grayscale and RGB images without pixel size limitations, while preserving the visual quality of the cover image. Performance evaluation and robustness testing show favorable Detection Rate, BPP, MSE, and PSNR values, highlighting the method's effectiveness in ensuring the confidentiality, integrity, and non-repudiation of digital data.

1. PENDAHULUAN

Seiring dengan berkembangnya zaman dan diikuti kemajuan dalam bidang teknologi, *file* atau dokumen saat ini diproduksi tidak hanya dalam bentuk cetak, tetapi juga dalam bentuk digital. Proses pendistribusian data secara digital melalui media elektronik secara online sangat dipermudah dalam hal proses pertukaran data, namun dibalik kemudahan tersebut terdapat beberapa kerentanan dan ancaman keamanan yang mungkin terjadi. Metode kriptografi dapat dimanfaatkan untuk memberikan layanan *confidentiality*, *data integrity*, *authentication* dan *non-repudiation* dalam menjaga kerahasiaan dokumen digital terdistribusi. Kemudian, juga diperlukan layanan lain yang berguna untuk melindungi hak cipta dari dokumen atau data digital, yaitu tanda tangan digital.

Tanda tangan digital adalah skema matematika yang mengandalkan isi pesan dan pengirim pesan untuk membuktikan keaslian pesan atau dokumen [1]. Tanda tangan digital terdiri dari serangkaian fungsi yang dihasilkan dari algoritma fungsi *hash* tertentu yang kemudian dienkripsi dengan algoritma kriptografi kunci asimetris. *Digital Signature Algorithm* (DSA) adalah standar untuk tanda tangan digital yang terdiri dari dua komponen utama, yaitu algoritma tanda tangan digital dan *Secure Hash Algorithm* (SHA). DSA digunakan untuk penandatanganan pesan dan fungsi SHA sebagai penghasil message digest. Saat proses distribusi data, dibutuhkan metode kriptografi sebagai layanan keamanan dan tanda tangan digital untuk melindungi hak cipta dari dokumen atau data digital [2]. Selain itu, diperlukan adanya metode tambahan yaitu steganografi agar nilai dari tanda tangan digital sebuah dokumen tersebut tidak dikirimkan secara terpisah, melainkan langsung menjadi satu di dalam dokumennya.

Steganografi adalah seni dan ilmu menyembunyikan informasi sedemikian rupa sehingga keberadaannya tidak dapat dideteksi dan terjadi komunikasi. Tujuan utama steganografi adalah untuk berkomunikasi secara aman dengan cara yang sama sekali tidak terdeteksi dan untuk menghindari kecurigaan terhadap transmisi data tersembunyi. Pada tahun 2020, Sabyasachi et.al [3] memadukan kriptografi dan steganografi dalam mentransmisikan data. Penelitian tersebut menunjukkan bagaimana steganografi dapat digabungkan dengan konsep tanda tangan digital untuk memberikan dukungan yang lebih baik dalam meningkatkan kerahasiaan dan keamanan transmisi data melalui internet. Perpaduan kriptografi dan steganografi digunakan untuk meningkatkan keamanan data. Data yang terenkripsi disembunyikan ke dalam gambar sampul, sehingga menghasilkan stego-image. Penelitian tersebut menunjukkan efektivitas kombinasi keduanya, namun tidak memanfaatkan teknik reversible steganography yang memungkinkan pemulihan penuh media asli setelah ekstraksi data.

Untuk meningkatkan kapasitas steganografi, pada tahun 2013 Li et al. mengusulkan metode *reversible image steganography* berbasis *Pixel Value Ordering* (PVO) [4]. PVO termasuk penyembunyian data lossless, yaitu setelah proses ekstraksi, data dan gambar sampul dapat dipulihkan [4]. PVO secara fleksibel membagi gambar sampul menjadi blok non-overlapping untuk mendapatkan histogram hasil prediksi eror. Kemudian, nilai maksimum dan minimum di setiap blok digunakan untuk menyematkan data. Pada setiap blok, nilai maksimum dinaikkan satu dan nilai minimum dikurangi satu, atau tidak diubah keduanya sehingga urutan *pixel* tidak berubah setelah disematkan, hal ini menjamin reversibility [5]. Meskipun PVO memiliki kapasitas dan kualitas yang baik, namun pada penelitian sebelumnya, umumnya hanya menggunakan data teks atau citra kecil sebagai *payload*, dan jarang mengeksplorasi penyisipan tanda tangan digital yang dihasilkan dari fitur citra.Oleh karena itu, pada penelitian ini akan dijabarkan implementasi dari DSA menggunakan SHA-512 yang digabungkan dengan salah satu metode *reversible image steganography* berbasis PVO dengan tambahan tahap ekstraksi fitur menggunakan *Scale Invariant Feature Transform* (SIFT) sebelum *hash*ing untuk menunjukkan salah satu layanan dari tanda tangan digital yaitu mencegah terjadinya non-repudiation terhadap pengiriman suatu informasi oleh seseorang. Penggunaan SIFT diharapkan meningkatkan kekokohan tanda tangan terhadap variasi citra, sekaligus menjaga kualitas visual citra setelah penyisipan. Evaluasi kinerja pada penelitian ini mencakup *Detection Rate, Bits Per Pixel* (BPP), *Mean Squared Error* (MSE), dan *Peak Signal-to-Noise Ratio* (PSNR). Selain itu, dilakukan uji ketahanan terhadap gangguan seperti *salt and pepper noise* dan serangan geometrik. Namun, mengingat perkembangan metode steganografi yang pesat, penelitian ini juga

mengidentifikasi adanya ruang untuk perbandingan yang lebih komprehensif dengan teknik-teknik terbaru, serta perlunya analisis performa pada berbagai kondisi, seperti penggunaan jenis gambar berbeda, format *file* beragam, dan ukuran data yang bervariasi.

2. LANDASAN TEORI

2.1. Digital Signature Algorithm (DSA)

Pada bulan Agustus 1991, U.S National Institute of Standards and Technology (NIST) mengusulkan DSA sebagai standar publik untuk tanda tangan digital. DSA telah menjadi U.S. *Federal Information Processing Standard* (FIPS 186) yang disebut *Digital Signature Standard* (DSS). DSA menggunakan SHA sebagai algoritma *hash*-nya [2]. Fungsi *hash* digunakan dalam proses pembuatan tanda tangan untuk mendapatkan *message digest*. Kemudian, message digest ditandatangani. Setelah itu, tanda tangan digital dikirim ke penerima yang dituju bersama dengan data yang ditandatangani atau sering disebut pesan. Penerima pesan dan tanda tangan memverifikasi tanda tangan dengan menggunakan kunci publik pengirim. Fungsi *hash* yang sama juga harus digunakan dalam proses verifikasi [2].

Parameter *Digital Signature Algorithm* (DSA)

- p = bilangan prima; $2^{511} < p < 2^{512}$. Parameter p bersifat publik
- q = faktor dari $p - 1$; $2^{159} < p < 2^{160}$. Parameter q berisfat publik.
- $g = h^{(p-1)/q} \, (mod\ p) > 1$; h adalah *random integer* dengan $1 < h < p$.

Pembangkitan Kunci

1) Pilih bilangan prima p dan q ; $(p - 1)\ mod\ q = 0$
2) Hitung $g = h^{(p-1)/q} \, (mod\ p)$
3) Tentukan kunci privat x ; $x < q$
4) $y = g^x \, (mod\ p)$; $1 < y < p$

Proses *Signing*

1) Ubah pesan m menjadi *message digest* dengan fungsi *hash* SHA
2) Tentukan bilangan acak $k < q$
3) Hitung r dan s sebagai berikut:
   $r = (g^k \ mod\ p)\ mod\ q$
   $s = (k^{-1}(H(m) + x * r))\ mod\ q$

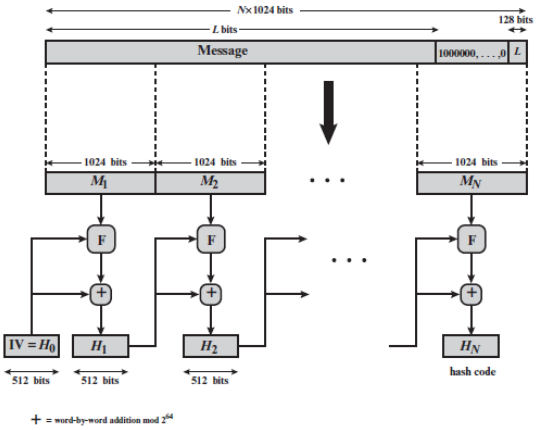
Digital signature dari pesan m adalah bilangan r dan s.

Proses Verifikasi

1) $w = s^{-1}\ mod\ q$
2) $u1 = (H(m) * w)\ mod\ q$
   $u2 = (r * w)\ mod\ q$
3) $v = \big((g^{u1} y^{u2}) mod\ p\big) mod\ q$
4) Jika $v = r$, tanda tangan sah, berarti pesan asli dan dikirim oleh pengirim yang benar.

2.2. *Secure Hash Algorithm* (SHA)

SHA adalah *one way hash function* yang dibuat oleh NIST dan digunakan bersama dengan DSA yang merupakan standar untuk pembuatan tanda tangan digital [1]. Algoritma SHA dijelaskan dalam dua tahap, yaitu *preprocessing* dan komputasi *hash*. *Preprocessing* melibatkan *padding* pesan, penguraian pesan yang telah di-*padding* menjadi blok m-bit, dan pengaturan nilai inisialisasi yang akan digunakan dalam perhitungan *hash*. Komputasi *hash* menghasilkan serangkaian nilai *hash*. Nilai *hash* akhir yang dihasilkan oleh perhitungan *hash* digunakan untuk menentukan *message digest.* Proses *generate message digest* digambarkan pada Gambar 1.



Gambar 1. Proses *generate message digest* menggunakan SHA-512

Step 1. *Padding* Pesan

Tujuan *padding* adalah untuk memastikan bahwa pesan adalah kelipatan 512 atau 1024-bit, tergantung pada algoritmanya. *Padding* dapat disisipkan sebelum perhitungan *hash* dimulai. Pesan di-*padding* sehingga panjangnya kongruen dengan 896 (mod 1024). *Padding* terdiri dari 1-bit diikuti dengan jumlah 0-bit yang diperlukan.

Step 2. Penguraian/Perluasan Pesan

Pesan dan hasil *padding*-nya harus diuraikan menjadi N m-bit blok. Untuk SHA-512, pesan dan nilai *padding*-nya harus diuraikan menjadi $N \times 1024$ -bit $M_1, M_2, \dots, M_N$. Karena 1024-bit dari blok input dapat dinyatakan sebagai 16 kata 64-bit, maka 64-bit pertama dari blok pesan i dinotasikan sebagai M_0^i, dan 64-bit selanjutnya adalah M_1^i, dan seterusnya hingga M_{15}^i.

Step 3. Inisialisasi Nilai *Hash* (H_0)

Sebelum perhitungan *hash* dimulai, nilai H_0 harus ditetapkan. Ukuran dan jumlah kata dalam H_0 bergantung pada ukuran *message digest*. Untuk SHA-512, nilai *hash* awal harus terdiri dari delapan kata 64-bit berikut, dalam *hexadecimal*:

$$a = 6A09E667F3BCC908 \qquad e = 510E527FADE682D1$$
$$b = BB67AE8584CAA73B \qquad f = 9B05688C2B3E6C1F$$
$$c = 3C6EF372FE94F82B \qquad g = 1F83D9ABFB41BD6B$$
$$d = A54FF53A5F1D36F1 \qquad h = 5BE0CD19137E2179$$

Kemudian, setelah *preprocessing* akan dilanjutkan dengan proses komputasi SHA-512, yang terdiri dari pemrosesan pesan dalam blok 1024-bit sebanyak 80-*round*.

Step 4. Memproses Pesan dalam Blok 1024-bit (128-*byte*)

SHA-512 terdiri dari 80-*round*. Setiap *round* mengambil input nilai $abcdefgh$ 512-bit. Setiap *round* t menggunakan nilai 64-bit, yang diturunkan dari blok 1024-bit (M_i). *Output* dari *round* ke-80 ditambahkan ke input *round* pertama H_{i-1} untuk menghasilkan H_i. Penambahan menggunakan *addition modulo* 2^{64}.

Step 5. *Output*

Setelah semua blok N 1024-bit diproses, *output* tahap ke-N adalah *message digest* 512-bit.

$H_0 = IV$

$H_i = SUM_{64}(H_{i-1}, abcdefgh_i)$

Keterangan:

- IV = *Initial value hash* $abcdefgh$, pada step 3
- $SUM64$ = operasi penambahan modulo 2^{64}
- $abcdefgh_i$ = *output* dari putaran terakhir pemrosesan blok pesan ke-i
  N = jumlah blok dalam pesan (termasuk *padding* dan panjang)

2.3. *Scale Invarian Feature Transform* (SIFT)

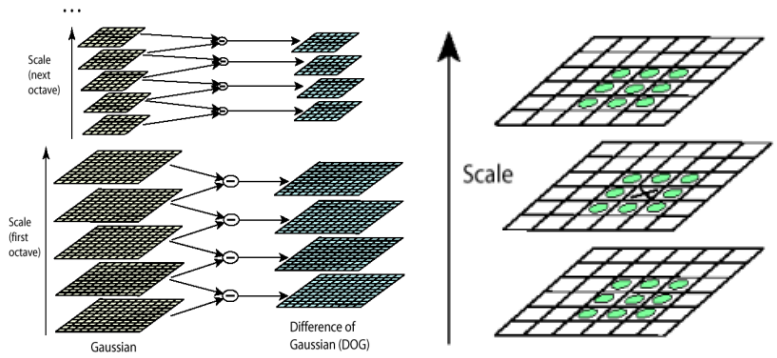
Algoritma *Scale Invarian Features Transform* (SIFT) adalah algoritma untuk metode ekstraksi fitur pada citra dengan mengubah citra menjadi fitur lokal yang kemudian akan digunakan sebagai fitur untuk mendeteksi objek yang diinginkan [6]. Alasan penggunaan SIFT pada penelitian ini adalah karena sifatnya yang invarian terhadap skala, rotasi, dan sebagian besar perubahan. Berbeda dengan *hash*ing langsung pada citra asli yang sensitif terhadap transformasi geometrik atau gangguan kecil, SIFT mengekstrak fitur lokal yang stabil, sehingga hasil *hash*ing menjadi lebih konsisten meskipun citra mengalami perubahan ukuran, orientasi, atau sedikit noise. Dengan demikian, SIFT dapat menghasilkan *signature* yang tahan terhadap modifikasi citra. SIFT bekerja melalui beberapa tahap utama, yaitu:

a) *Scale-Space Extrema Detection*

Scale Space Extrema Detection dilakukan untuk menemukan titik-titik potensial atau *keypoint* pada berbagai skala $L(x, y, \sigma)$ yang dihasilkan menggunakan Gaussian blur. Kemudian, mencari perbedaan antar skala *(Difference of Gaussian*) guna mendeteksi titik maksimum atau minimum lokal. Skala ruang suatu citra didefinisikan sebagai fungsi:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

- $L\ (x, y, \sigma)$
  Output dari persamaan, yang merupakan ciri khas invarian yang diekstraksi dari citra
- $G\ (x, y, \sigma)$
  Filter Gaussian, digunakan untuk memburamkan gambar.
- $I\ (x, y)$
  Gambar masukan



Gambar 2. *Difference of Gaussian* (DoG) dan Perbandingan *Pixel*

Pada Gambar 2 memperlihatkan contoh *Difference of Gaussian* (DoG) dan perbandingan *pixel*. Untuk setiap oktaf ruang skala, gambar awal dengan Gaussian digabungkan berulang kali untuk menghasilkan satu set gambar ruang skala yang ditunjukkan di sebelah kiri. Gambar-gambar Gaussian yang berdekatan dikurangi untuk menghasilkan gambar-gambar Gaussian yang berbeda di sebelah kanan. Setelah setiap oktaf, gambar Gaussian diambil sampelnya dan prosesnya berulang. Untuk mendeteksi maksimum dan minimum lokal dari $D(x, y, \sigma)$, 1-*pixel* dalam gambar dibandingkan dengan delapan tetangganya pada citra saat ini dan sembilan tetangga pada skala di atas dan di bawah. Jika nilai *pixel* saat ini lebih besar dari seluruh

tetangga, maka *pixel* tersebut adalah maksimum lokal. Jika nilai *pixel* saat ini lebih kecil dari seluruh tetangga, maka *pixel* tersebut adalah minimum lokal. Dengan cara ini, total 26 pemeriksaan dilakukan. Jika itu adalah titik ekstrim lokal, itu adalah titik kunci potensial. Ini pada dasarnya berarti bahwa *keypoint* paling baik diwakili dalam skala itu.

b) *Keypoint Localization*

Setelah kandidat *keypoint* ditemukan, dilakukan analisis menggunakan matriks Hessian yang membantu menilai bentuk lokal di sekitar titik tersebut. Dengan cara ini, sistem dapat membedakan apakah suatu titik berada di tepi atau di area dengan detail yang cukup. Berikut adalah langkah-langkah untuk menghilangkan tepi menggunakan matriks Hessian:

- Hitung matriks Hessian pada titik (i, j) yaitu baris ke-i kolom ke-j dari matriks DoG, dengan menghitung turunan parsial kedua dari DoG pada titik tersebut:

$$H = \begin{bmatrix} \frac{d^2I}{dx^2} & \frac{d^2I}{dxdy} \\ \frac{d^2I}{dydx} & \frac{d^2I}{dy^2} \end{bmatrix}.$$

- Selanjutnya, menghitung eigenvalue dari matriks Hessian. Misalkan α adalah nilai eigen dengan *magnitude* terbesar dan β adalah yang lebih kecil. Maka jumlah nilai eigen matriks H adalah *trace* dari $H(Tr(H))$ dan perkalian nilai eigen matriks H adalah determinan matriks $H\big(Det(H)\big)$ yang dapat ditulis sebagai berikut:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$Det(H) = D_{xx}D_{yy} - \left(D_{xy}\right)^2 = \alpha\beta.$$

- Setiap titik yang terletak pada gambar, hasil dari nilai eigen yang diperoleh nantinya akan dibandingkan dengan nilai ambang batas. Jika nilai eigen lebih kecil dari ambang batas, maka titik tersebut dapat dianggap sebagai keypoint, dan sebaliknya jika nilai eigen lebih besar dari ambang batas, maka titik tersebut dianggap sebagai tepi dan dihapus. Nilai pada tepi gambar akan dihapus jika nilai kelengkungan utama di atas ambang batas $|D\ (\hat{x})| < 0{,}03$.

c) *Orientation Assignment*

Setelah mendapatkan nilai eigen yang menunjukkan kelengkungan utama dari matriks Hessian, dilakukan analisis lebih lanjut dengan mempertimbangkan nilai eigen yang diperoleh [7]. Untuk melakukan proses *orientation assignment* dengan metode histogram gradien, langkah-langkah yang perlu dilakukan adalah sebagai berikut:

- Menghitung nilai gradien G_x dan G_y pada setiap *pixel* (x, y) di sekitar titik *keypoint* menggunakan operator Prewitt,

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

- Menghitung *magnitude gradien* (m) dan *orientation gradien* (θ) pada setiap *pixel* menggunakan rumus:

$$\boldsymbol{m(x,y)} = \sqrt{\big(\boldsymbol{L(x+1,y) - L(x-1,y)}\big)^2 + \big(\boldsymbol{L(x,y+1) - L(x,y-1)}\big)^2}$$
$$\theta(x,y) = tan^{-1}\left(\frac{\left(L(x,y+1) - L(x,y-1)\right)}{L(x+1,y) - L(x-1,y)}\right).$$

- Selanjutnya, hitung histogram orientasi dengan membagi rentang 360 derajat menjadi delapan Bin, dengan interval setiap 45 derajat [8]. Kemudian, setiap *pixel* diberi bobot (*weight*) berdasarkan *magnitude gradien* dari *pixel* tersebut. menggunakan *Gaussian weighting function* [9] dengan σ yaitu 1,6 kali skala *keypoint*. Bobot *Gaussian* dihitung dengan memperhatikan jarak (d) antara *pixel* dan *keypoint* dalam daerah sekitarnya. Jarak tersebut dapat dihitung dengan menggunakan jarak *Euclidean*,

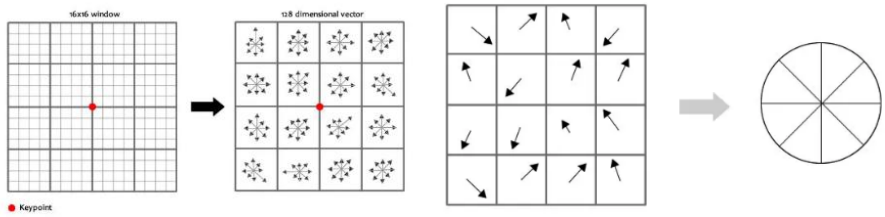
$$w(\theta) = exp\left(-\frac{(d^2)}{(2 \times (\sigma^2))}\right).$$

- Langkah terakhir adalah melakukan normalisasi histogram pada vektor fitur atau kontribusi yang telah dihasilkan untuk memperbaiki invarian skala dan rotasi dengan menggunakan Teknik L2 *Normalization*.

$$c_{i_{normalized}} = \frac{c_i}{\sqrt{\sum c_i^2}} \times 0.2.$$

d) *Generation of Keypoint Descriptors*

Pada titik ini, setiap *keypoint* memiliki lokasi, skala, *orientation*. Berikutnya menghitung deskriptor untuk wilayah gambar lokal tentang setiap *keypoint* yang khas dan invarian untuk variasi seperti perubahan sudut pandang dan iluminasi [7]. Untuk melakukan ini, jendela 16×16 di sekitar titik kunci diambil dan dibagi menjadi 16 sub-blok berukuran 4×4 [10], seperti terlihat pada Gambar 3.



Gambar 3. *Keypoint* dan Vektor Fitur

Histogram masing-masing berisi 8 bin, dan setiap deskriptor berisi larik 4 histogram di sekitar *keypoint*. Ini mengarah ke vektor fitur SIFT dengan 4 × 4 × 8 = 128 elemen. Arah 4×4×8 memberikan nilai 128 bin. Ini direpresentasikan sebagai vektor fitur untuk membentuk *keypoint descriptor* [10].

2.4. Steganografi

Kata steganografi berasal dari bahasa Yunani Steganos, yang berarti tertutup atau rahasia dan graphy berarti menulis atau menggambar [11]. Sistem steganografi menyematkan konten tersembunyi di media sampul agar tidak menimbulkan kecurigaan penyadap. Hampir semua format *file* digital dapat digunakan untuk steganografi, namun format yang lebih cocok adalah yang memiliki tingkat redundansi yang tinggi. Redundansi dapat didefinisikan sebagai bit-bit dari suatu objek yang memberikan akurasi lebih besar dari yang diperlukan untuk penggunaan dan tampilan objek. Bit redundan dari sebuah objek adalah bit yang dapat diubah tanpa terdeteksi perubahannya dengan mudah.

Model dasar steganografi terdiri dari *Carrier, Message* dan *Password*. *Carrier* juga dikenal sebagai *cover-object*, yaitu tempat untuk menyematkan pesan dan berfungsi untuk menyembunyikan keberadaan pesan. *Message* adalah data yang ingin dirahasiakan oleh pengirim, dapat berupa teks biasa, *ciphertext*, gambar lain, atau apa pun yang dapat disematkan dalam aliran bit seperti tanda hak cipta, komunikasi rahasia, atau nomor seri. *Password* dikenal sebagai stego-key, yang memastikan bahwa hanya penerima yang mengetahui kunci decoding yang sesuai yang dapat mengekstrak pesan dari cover-object. Cover-object dengan pesan yang disematkan secara rahasia kemudian disebut stego-object. Memulihkan pesan dari stego-object memerlukan cover-object itu sendiri dan kunci decoding yang sesuai jika stego-key digunakan selama proses penyandian. Teknik steganografi pada gambar digital dapat dievaluasi dengan tiga parameter utama [12], yaitu:

a) *Hiding Capacity*

Hiding capacity adalah kapasitas yang merujuk pada jumlah informasi yang disembunyikan dalam media gambar *cover*. *Hiding capacity* dibagi dalam dua cara, yaitu *maximum hiding capacity* dan *bitrate*. M*aximum hiding capacity* adalah jumlah maksimum data yang dapat disembunyikan dalam gambar dan direpresentasikan dalam bit atau *byte* atau *kilobyte*. Sedangkan, *bitrate* atau

laju bit adalah jumlah maksimum bit yang dapat disembunyikan per *pixel* atau sering disebut sebagai *Bit Per Pixel* (BPP), dengan rumus:

$$\text{BPP} = \frac{Max\ Hiding\ Capacity}{\text{Ukuran } Pixel}.$$

BPP dikatakan rendah apabila kapasitas penyembunyian data per *pixel* relatif terbatas, dan perubahan visual mungkin hampir tidak terlihat. Sebaliknya, BPP yang tinggi mencerminkan kapasitas penyembunyian yang lebih besar dan mungkin menghasilkan perubahan visual yang lebih signifikan. Kemudian, terdapat detection rate untuk mengukur tingkat deteksi yang dilakukan oleh pihak yang tidak berwenang. Detection rate adalah persentase *pixel* yang berhasil dideteksi sebagai pesan tersembunyi dari total *pixel* yang diperiksa pada gambar terenkripsi. Semakin rendah detection rate, semakin sulit bagi pihak yang tidak berwenang untuk mendeteksi informasi tersembunyi dan semakin tinggi tingkat keamanan [13]. Berikut merupakan rumus *detection rate*:

$$\text{DR} = \left(\frac{Pixel \text{ yang terdeteksi}}{\text{Jumlah } Pixel}\right) \text{x } 100\%.$$

b) Ukuran Distorsi

Ukuran distorsi digunakan untuk mengukur sejauh mana kualitas gambar hasil steganografi dibandingkan dengan gambar asli, atau untuk membandingkan kualitas hasil steganografi dari berbagai teknik steganografi yang berbeda distorsi dapat diukur dengan menggunakan banyak metrik, beberapa contohnya adalah:

- *Mean Square Error* (MSE)

  MSE adalah nilai eror kuadrat rata-rata antara gambar asli dengan gambar hasil penyisipan (gambar *stego*). MSE antara gambar asli dan gambar *stego* dihitung dengan menggunakan:

  $$\text{MSE} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - q_{ij})^2.$$

  Nilai *pixel* gambar asli dan nilai *pixel* gambar *stego* pada baris ke-i dan kolom ke-j dilambangkan dengan p_{ij} dan q_{ij}. Sedangkan, jumlah baris dan kolom pada gambar digital dilambangkan dengan m dan n. MSE memberikan gambaran tentang sejauh mana perbedaan antara gambar asli dan gambar *stego*. Semakin tinggi nilai MSE, semakin besar perbedaan antara dua set data tersebut.

- *Peak Signal to Noise Ratio* (PSNR)

  PSNR adalah ukuran distorsi dalam gambar *stego*. PSNR biasanya digunakan untuk mengukur tingkat *imperceptibility*. Berikut merupakan rumus dari PSNR:

  $$\text{PSNR} = 10 \times log_{10} \frac{255 \times 255}{\text{MSE}}.$$

  Nilai PSNR yang tinggi, berarti distorsinya rendah. Nilai PSNR yang lebih dari 40 desibel (dB) dianggap sangat baik. Jika nilai PSNR berada di antara 30 dB dan 40 dB, nilai tersebut masih dapat diterima, tetapi jika nilai PSNR kurang dari 30 dB maka nilai tersebut tidak dapat diterima karena distorsinya sangat tinggi.

c) Keamanan

Suatu teknik steganografi dikatakan aman jika tahan terhadap berbagai *steganalysis*. Ada berbagai skema steganalysis untuk menguji keamanan teknik steganografi. Teknik *steganalysis* yang secara khusus terkait dengan metode PVO adalah *steganalysis* berbasis *Distortion Measure* (DM). Teknik DM pada dasarnya mengukur distorsi atau perbedaan antara *pixel* yang satu dengan *pixel* lainnya yang seharusnya terurut dalam domain spasial pada gambar stego dengan yang ada pada gambar asli. Sebuah metode steganografi juga memiliki enam aspek yang dapat

menentukan baik atau tidaknya sebuah steganografi dalam melakukan pekerjaannya [14], diantaranya yaitu:

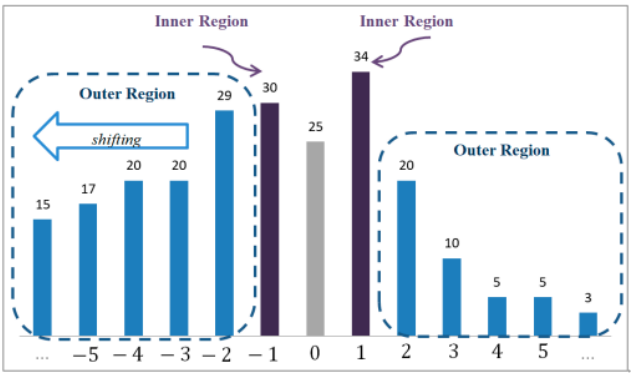
- Kapasitas (*Capacity*), diukur dengan BPP.
- Tidak Terlihat (*Imperceptibility*), diukur dengan PSNR dan MSE.
- Keamanan (*Security*), diukur dengan BPP & DR rendah.
- Ketahanan (*Robustness*), diukur dengan PSNR dan diuji dengan *geometrical attack* serta *salt & pepper noise.*
- Pemulihan (*Recovery*), diukur dengan nilai DR setelah serangan.
- Kompleksitas (*Complexity*), diukur dengan kemudahan implementasi & efisiensi algoritma.

2.5. *Pixel* Value Ordering (PVO)

PVO adalah metode penyisipan data yang menyembunyikan pesan rahasia dalam perbedaan *pixel* terbesar dan *pixel* terbesar kedua dari sebuah blok. Sebuah gambar digital dibagi menjadi beberapa blok, dan pada masing-masing blok nilai *pixel* diurutkan dari yang terkecil hingga terbesar. Dari setiap blok, diperoleh nilai penting:

- *Maximum predicted error* (PE_{max}) dihasilkan melalui penggunaan perbedaan antara nilai *pixel* terbesar $x_{\Delta(n)}$ dengan nilai *pixel* terbesar kedua $x_{\Delta(n-1)}$.
- *Minimum predicted error* (PE_{min}) dihasilkan melalui perbedaan antara nilai *pixel* terkecil $x_{\Delta(1)}$ dengan nilai *pixel* terkecil kedua $x_{\Delta(2)}$.

Pada tahap *embedding*, Li *et al*. [15] membagi histogram menjadi dua bagian (Proses *Location Map*), yaitu bagian sisi kiri dan bagian sisi kanan, seperti terlihat pada Gambar 4. Kedua bagian ini dipisahkan menjadi dua wilayah, yaitu wilayah dalam dan wilayah luar. Wilayah dalam didefinisikan sebagai area nilai eror, yaitu nilai yang mewakili perbedaan antara *pixel* terbesar dan terbesar kedua dari sebuah blok, terletak antara 1 dan -1. Semua nilai lain dianggap berada di wilayah luar.



Gambar 4. Histogram Perbedaan PVO

Ketika nilai perbedaan prediksi PE_{max} dan PE_{min} yang dihasilkan oleh PVO digunakan, titik nilai puncak sebagian besar gambar akan berada dalam kisaran 1 dan -1. Nilai 1 dan −1 dianggap sebagai wilayah dalam tempat informasi rahasia dapat disematkan. Jika nilai PE_{max} sama dengan 1, maka satu bit rahasia b dapat disembunyikan ke dalam nilai $x_{\Delta(n)}$ untuk menghasilkan nilai *stego* $x'_{\Delta(n)}$. Jika PE_{max} sama dengan 0, maka nilai tersebut tidak dapat digunakan untuk menyembunyikan informasi dan nilai *stego* sama dengan nilai awalnya $x_{\Delta(n)}$. Jika PE_{max} tidak terletak di wilayah dalam dan tidak sama dengan 0, maka nilai $x_{\Delta(n)}$ perlu ditambahkan 1. Di sisi lain, jika nilai PE_{min} sama dengan −1, maka satu bit rahasia b dapat disembunyikan ke dalam nilai $x_{\Delta(1)}$ untuk menghasilkan nilai *stego* $x'_{\Delta(1)}$. Jika PE_{min} sama dengan 0, maka nilai tersebut tidak dapat digunakan untuk menyembunyikan informasi, artinya nilai *stego* sama dengan nilai awalnya yaitu

$x_{\Delta(1)}$. Jika PE_{min} tidak terletak di daerah dalam dan tidak sama dengan 0, maka nilai $x_{\Delta(1)}$ perlu dikurangi 1, yang dapat dinyatakan dengan notasi berikut:

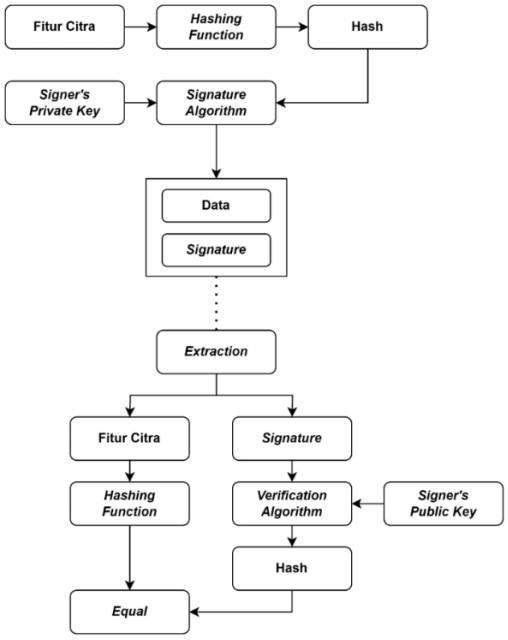
$$x'_{\Delta(n)} = \begin{cases} x_{\Delta(n)}, & \boldsymbol{if\ PE_{max} = 0} \\ x_{\Delta(n)} + b, & \boldsymbol{if\ PE_{max} = 1} \\ x_{\Delta(n)} + 1, & otherwise \end{cases}$$

$$x'_{\Delta(1)} = \begin{cases} x_{\Delta(1)}, & \boldsymbol{if\ PE_{min} = 0} \\ x_{\Delta(1)} - b, & \boldsymbol{if\ PE_{min} = -1} \\ x_{\Delta(1)} - 1, & otherwise \end{cases}$$

Metode PVO memiliki keunggulan dalam menjaga kualitas gambar sekaligus memberikan kapasitas penyisipan yang cukup baik pada gambar dengan tekstur yang bervariasi. Namun, kapasitasnya dapat menurun pada citra homogen. Dibandingkan metode terbaru seperti *Prediction-Error Expansion* (PEE), *Histogram Shifting* (HS), atau steganografi berbasis *deep learning* yang menawarkan kapasitas lebih tinggi dan ketahanan lebih baik terhadap *steganalysis*, PVO memiliki keunggulan pada implementasi sederhana dan kualitas citra yang baik untuk citra dengan tekstur bervariasi. Oleh karena itu, penggunaan PVO dalam penelitian ini dipadukan dengan proses pra-pemrosesan menggunakan DSA, SHA 512 dan SIFT untuk meningkatkan keamanan tanpa mengorbankan kualitas visual citra.

3. METODE PENELITIAN

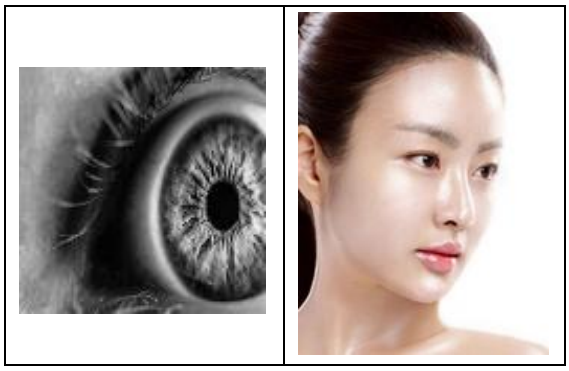
Penelitian ini dilakukan dengan dua metode penelitian, yaitu metode telaah kepustakaan dan metode eksperimen. Metode telaah kepustakaan dilakukan dengan cara studi literatur sumber-sumber yang mendukung penelitian ini seperti buku, artikel ilmiah, dan sumber-sumber lainnnya. Studi literatur bertujuan untuk memahami konsep dan teori terkait digital signature, DSA, dan Steganografi terutama metode *reversible image steganography* berbasis PVO. Metode eksperimen terdiri dari proses pembangkitan kunci, *hashing* dan *signing*, *embedding*, ekstraksi, verifikasi, evaluasi performa, dan uji *robustness*. Diagram alir penelitian ditampilkan pada Gambar 5.



Gambar 5. Diagram Penelitian

4. HASIL DAN PEMBAHASAN

Dalam implementasi model DSA ini akan dilakukan dengan proses pembangkitan kunci, ekstraksi fitur, *hash*ing dan *signing*, *embedding*, ekstraksi, verifikasi dan evaluasi performa. Salah satu percobaan menggunakan gambar *grayscale* dan RGB berukuran 400×400 *pixel* dengan format JPEG seperti terlihat pada Gambar 6.



Gambar 6. Contoh Gambar *Grayscale* (kiri) dan RGB (kanan)

Berikut merupakan percobaan yang telah dilakukan:

A. Pembangkitan Kunci

Proses pembangkitan kunci akan menghasilkan *output* berupa kunci publik (p, q, g, y) dan kunci privat x yang dapat dilihat pada Tabel 1.

Tabel 1. *Output* Pembangkitan Kunci

Gambar *Grayscale*	
p	13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084095
q	1157920892373161954235709850086879078532699846656405640394575840079131296397 47
g	3153692574574233073931147462144077611977337381744196010383363923228409391206242100763827775820475316794350115190517524181182588684305211008073510234715076
y	3153692574574233073931147462144077611977337381744196010383363923228409391206242100763827775820475316794350115190517524181182588684305211008073510234715076
x	3153692574574233073931147462144077611977337381744196010383363923228409391206242100763827775820475316794350115190517524181182588684305211008073510234715076
Gambar RGB	
p	13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084095
q	1157920892373161954235709850086879078532699846656405640394575840079131296397 47
g	2257954669008145632734427537072389397148208407250793828195081869139182467694189048976934865392697906132832618874108796131231655625600094290687076787066 11
y	5553516418472262520587950659933581590433230573970526381381947352710701134397174076706298994193695670459634114205462930860244750171646225365245011362766351
x	8409840448342710314315824221759234589123646352000402734080538617434354183606

B. Pembangkitan Tanda Tangan

Dalam proses tanda tangan digital ini dilakukan *hashing* pada *file* gambar dan pembentukan tanda tangan digital. Proses ini memerlukan *input* berupa *file* gambar dan kunci privat. Sebelum dilakukan proses *hashing*, dilakukan ekstraksi fitur menggunakan algoritma SIFT pada gambar yang akan ditandatangani. Sehingga, didapatkan nilai *keypoint descriptor* sejumlah 128, pada Tabel 2, yang digunakan untuk proses *hashing*.

Tabel 2. Nilai *Keypoint Descriptor*

Jenis Gambar	Nilai *Keypoint Descriptor*
Grayscale	[0; 0; -0.0003; -0.0001; -0.0001; -0.0003; 0; 0; 0; 0; -0.0006; -0.0004; -0.0004; -0.0006; 0; 0; 0; 0; -0.0054; -0.0003; -0.0003; -0.0054; 0; 0; 0; 0; 0.0003; 0.0002; 0.0002; 0.0003; 0; 0; 0; 0; -0.0002; 0.0000; 0.0000; -

	0.0002; 0; 0; 0; 0; -0.0000; -0.0002; -0.0002; -0.0000; 0; 0; 0; 0; -0.0001; -0.0000; -0.0000; -0.0001; 0; 0; 0; 0; -0.0001; -0.0002; -0.0002; -0.0001; 0; 0; 0; 0; 0.0000; 0.0000; 0.0000; 0.0000; 0; 0; 0; 0; 0.0003; 0.0000; 0.0000; 0.0003; 0; 0; 0; 0; -0.0005; 0.0001; 0.0001; -0.0005; 0; 0; 0; 0; 0.0002; 0.0001; 0.0001; 0.0002; 0; 0; 0; 0; 0.0002; 0.0034; 0.0034; 0.0002; 0; 0; 0; 0; -0.0001; -0.0000; -0.0000; -0.0001; 0; 0; 0; 0; -0.0019; 0.2000; 0.2000; -0.0019; 0; 0; 0; 0; 0.0001; 0.0000; 0.0000; 0.0001; 0; 0]
RGB	0; 0

Kemudian, dilakukan *hash*ing menggunakan SHA-512 dan menghasilkan nilai *digest* pada Tabel 3.

Tabel 3. Nilai *Digest*

Jenis Gambar	Nilai Digest
Grayscale	$H(m)$ = 735b82b642afa435d7f31d018f8f6840613741173d1412810e865c4417e68490a49bcfd973be674d9e7b04ad1a7b356b659cdfec7b630caea7c7bf8d53c235aa
RGB	$H(m)$ = 5523f23a5a829439f115138164fe8d80b340f5ff684bcd5f936652e22dc774e41cd735714cc95d482bc753889f24f5169c7937e8b30066dab14cd13256a2b3e9

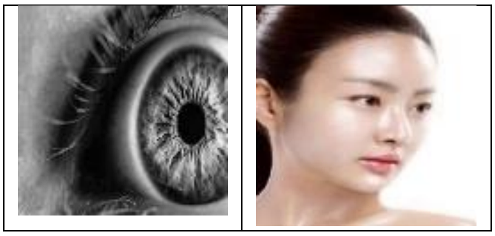
Setelah itu dibangkitkan tanda tangan pada gambar, dan diperoleh nilai *signature* pada Tabel 4.

Tabel 4. Nilai *Signature*

Jenis Gambar	Nilai *Signature*
Grayscale	(r, s) = (4572174691723554426287294215222075311834500980264252275212319828699033609485, 109853145852336414673467589329163649607426454923726419127324238342961340302439)
RGB	(r, s) = (70958018311398068205020626776725931944057718986909062979132069828948504344625, 68601662015825715757070969908282800360354069336961588681929564000499268198939)

C. *Embedding* dan Ekstraksi

Nilai *signature* yang telah didapatkan pada Tabel 4 disematkan atau *embedding* ke dalam gambar yang dikirimkan menggunakan *reversible image* steganografi berbasis PVO. Gambar 7 merupakan hasil gambar *stego* setelah dilakukan *embedding* dan ekstraksi terhadap Gambar 6.



Gambar 7. Hasil Gambar *Stego*

D. Verifikasi Tanda Tangan

Proses verifikasi tanda tangan digital memerlukan input berupa kunci publik, pesan gambar, dan tanda tangan digital. Karena dalam tahap sebelumnya telah mendapatkan kunci publik (p, q, g, y), pesan gambar, dan tanda tangan digital (r, s) maka dapat dilakukan proses verifikasi tanda tangan digital. Perhitungan nilai parameter $(w, u1, u2, v)$ dapat dilihat pada Tabel 6.

Tabel 6. Perhitungan Nilai Parameter

Gambar *Grayscale*	
w	114744645680325795425134431202901840518229230334559206241972592445810495932022
$u1$	34636282011841315995183290098886942650553212451494231007705570107242702863401
$u2$	54080440546662106197055313362701740639885925088811107157418685212910085196915
v	4572174691723554426287294215222075311834500980264252275212319828699033609485
Gambar RGB	
w	56001450620252375370870319034830695748619430143793288518744521471697604863228

$u1$	37851839307314222170123094798826510964058611305244427900888820984017664960723
$u2$	95388748033476963636012344629063056104510377143036790520817582344809589972921
v	70958018311398068205020626776725931944057718986909062979132069828948504344625

Karena nilai $r = v$, maka tanda tangan yang dikirimkan adalah tanda tangan yang sah.

E. Evaluasi Performa

Hasil analisis evaluasi performa metode *reversible image steganography* berbasis PVO terhadap *capacity, imperceptibility*, dan *robustness* pada gambar *Grayscale* dan RGB dengan ukuran 400×400 *pixel* dan format JPEG pada Gambar 6 dapat dilihat pada Tabel 7.

Tabel 7. Hasil Analisis Evaluasi Performa

Jenis Gambar	*Detection Rate*	BPP	MSE	PSNR
Grayscale	0.00%	0.0032	0.0039	72.1579
RGB	0.03%	0.0032	0.3021	53.3283

1. *Capacity*

   Semakin rendah nilai BPP maka semakin sedikit bit yang disisipkan pada setiap *pixel* gambar. Hal tersebut menunjukkan efisiensi tinggi dalam penggunaan ruang untuk menyembunyikan pesan rahasia. Dengan kata lain, semakin rendah nilai BPP, semakin sedikit perubahan yang terjadi pada setiap *pixel* gambar, sehingga menyebabkan perubahan yang lebih kecil pada gambar hasil steganografi. Nilai *Detection Rate* yang sebesar 0.00% menunjukkan bahwa data yang tersembunyi sangat sulit untuk dideteksi.

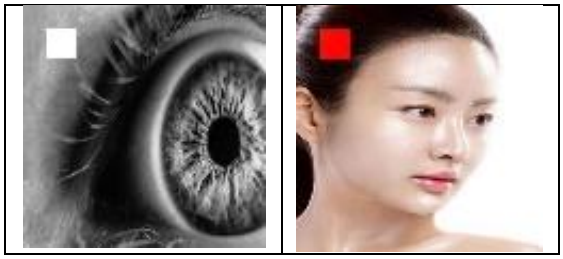
2. *Imperceptibility*

   Gambar dengan format JPEG memiliki performa yang baik dalam mempertahankan kualitas visual karena memiliki PSNR yang tinggi, yaitu 72.1579.

3. *Robustness*

   Gambar *grayscale* dengan nilai PSNR yang tinggi dan MSE yang rendah menunjukkan bahwa gambar tersebut memiliki kualitas rekonstruksi yang sangat baik ketika mengalami proses kompresi dan dekompresi ulang. Sehingga, gambar *grayscale* memiliki tingkat ketahanan yang lebih baik dalam menjaga kualitas gambar.

   Hasil uji *robustness* menggunakan jenis gangguan *Salt and Pepper* dengan intensitas *noise* adalah 0.02 atau 2% serta presentase hasil perbandingan *file* hasil ekstraksi antara sebelum dan sesudah diberi *noise Salt and Pepper* menunjukkan ketepatan signature 100%. Hasil pengujian ini menunjukkan bahwa sistem PVO pada gambar *grayscale* memiliki ketahanan dan performa yang baik dalam mengatasi *Salt and Pepper* khususnya intensitas 2% dengan tidak terdapat perbedaan pada *file signature* setelah diekstraksi. Gambar 8 merupakan hasil uji *robustness* menggunakan *Geometrical Attack* dengan ukuran *cropping* adalah 50×50 *pixel* pada posisi koordinat (40,40) untuk gambar berukuran 400×400 *pixel* dengan format JPEG serta hasil perbandingan *file* hasil ekstraksi antara sebelum dan sesudah dilakukan *cropping*.



Gambar 8. Hasil Uji *Robustness*

Data yang dihasilkan menunjukkan bahwa tidak terjadi perubahan nilai signature hasil ekstraksi pada gambar *grayscale* dan RGB format JPEG yang telah mengalami *cropping*, dikarenakan bagian gambar yang dipotong adalah bagian yang kosong atau background yang polos.

5. KESIMPULAN

Implementasi DSA menggunakan SHA-512 yang diterapkan pada media gambar dapat menjadi cara untuk menjaga keaslian pada dokumen dalam bentuk digital sebagaimana telah ditunjukkan pada penelitian ini. Secara keseluruhan, performa teknik steganografi yang dihasilkan memiliki kualitas gambar yang berbeda-beda tergantung dengan jenis, format, dan ukuran gambar. Steganografi PVO pada gambar *grayscale* memiliki performa yang lebih baik daripada gambar RGB karena gambar *grayscale* hanya memiliki satu *channel* warna (*gray*) sedangkan gambar RGB memiliki tiga *channel* warna (*red, green, blue*). Oleh karena itu, informasi yang disisipkan pada gambar *grayscale* lebih fokus dan padat, sehingga lebih sulit untuk dideteksi. Pada pengujian citra RGB, meskipun PSNR sedikit lebih rendah dibanding *grayscale*, kualitas visual tetap berada pada tingkat yang tidak terdeteksi secara kasat mata oleh pengamat manusia. Tantangan utama pada RGB adalah distribusi *payload* ke tiga *channel* warna yang membuat kapasitas per *channel* lebih rendah, namun hal ini juga dapat dimanfaatkan untuk menyamarkan pola *embedding* dan meningkatkan ketahanan terhadap deteksi.

Kemudian, berdasarkan uji *robustness* yang telah dilakukan, dapat diambil kesimpulan bahwa performa PVO dalam melakukan implementasi *Digital Signature Algorithm* (DSA) dengan SHA-512 pada Citra Digital menggunakan metode *Reversible Image Steganography* berbasis *Pixel Value Ordering* (PVO) memiliki tingkat keberhasilan yang berbeda tergantung pada jenis dan format gambar, ukuran gambar, jumlah *file* yang disisipkan, jenis gangguan yang dilakukan dan spesifikasi gangguan yang digunakan.

REFERENSI

[1] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1996.

[2] NIST, *The Digital Signature Standard*. Gaithersburg, MD: NIST, 1994.

[3] S. Pramanik, S. K. Bandyopadhyay, and R. Ghosh, "*Signature image hiding in color image using steganography and cryptography based on digital signature concepts*," in Proc. 2nd Int. Conf. *Innovative Mechanisms for Industry Applications* (ICIMIA), Mar. 2020, pp. 665–669. doi: 10.1109/ICIMIA48430.2020.9074957.

[4] H. H. Liu and C. M. Lee, "*High-capacity reversible image steganography based on pixel value ordering*," EURASIP J. *Image Video Process*., vol. 2019, no. 1, Dec. 2019. doi: 10.1186/s13640-019-0458-z.

[5] H. X. Chi, J. H. Horng, and C. C. Chang, "*Reversible data hiding based on pixel-value-ordering and prediction-error triplet expansion*," *Mathematics*, vol. 9, no. 14, Jul. 2021. doi: 10.3390/math9141703.

[6] "Implementasi ekstraksi fitur jumlah *keypoint descriptor* pada pengenalan tanda tangan dengan algoritma *learning vector quantization*," [Online].

[7] Y. Tao, Y. Xia, T. Xu, and X. Chi, "*Research progress of the scale invariant feature transform* (SIFT) *descriptors*," J. Convergence Inf. Technol., vol. 5, no. 1, pp. 116–121, Feb. 2010. doi: 10.4156/jcit.vol5.issue1.13.

[8] J. Jiang, X. Li, and G. Zhang, "*SIFT hardware implementation for real-time image feature extraction*," IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 7, pp. 1209–1220, 2014. doi: 10.1109/TCSVT.2014.2302535.

[9] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. London: Springer, 2021. [*Online*]. *Available*: https://szeliski.org/Book

[10] "*Local features tutorial*," [*Online*]. *Available*: https://www.cs.ubc.ca/~lowe/vision.html

[11] M. M. Amin, M. Salleh, S. Ibrahim, M. R. Katmin, and M. Z. I. Shamsuddin, "*Information hiding using steganography*," in Proc. 4th Nat. Conf. Telecommun. Technol. (NCTT), 2003, pp. 21–25. doi: 10.1109/NCTT.2003.1188294.

[12] A. Pradhan, A. K. Sahu, G. Swain, and K. R. Sekhar, "*Performance evaluation parameters of image steganography techniques*," Int. J. Comput. Sci. Eng., vol. 5, no. 6, pp. 145–152, 2017.

[13] N. Provos, "*The basics of embedding*," IEEE Secur. Privacy Mag., vol. 1, no. 3, pp. 32–44, 2003.

[14] M. Begum and M. S. Uddin, "*Analysis of digital image watermarking techniques through hybrid methods*," Adv. Multimedia, vol. 2020, pp. 1–10, 2020. doi: 10.1155/2020/7912690.

[15] X. Li, J. Li, B. Li, and B. Yang, "*High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion*," *Signal Process*., vol. 93, no. 1, pp. 198–205, Jan. 2013. doi: 10.1016/j.sigpro.2012.07.025.