

Implementasi Algoritma Simulated Annealing (SA) dalam Steganografi Discrete Cosine Transform (DCT)

I Nyoman Dwi Pradnyana Putra^{a1}, I Gede Santi Astawa^{a2},
I Ketut Gede Suhartana^{a3}, Gusti Ayu Vida Mastrika Giri^{a4}

^aProgram Studi Informatika, Universitas Udayana
Badung, Bali, Indonesia

¹dwipradnyana68@gmail.com

²santi.astawa@unud.ac.id

³kg.suhartana@unud.ac.id

⁴vida@unud.ac.id

Abstract

Maintaining confidential data security is very important in an era of rapid technological developments. One method used for this purpose is steganography, which is a technique that allows the hiding of sensitive messages in other media without the knowledge of unauthorized parties. In this context, the Discrete Cosine Transform (DCT) method is used to insert messages into the essential frequency components of the media, such as images. However, the use of DCT can damage image quality, and this is why optimization is necessary. In this research, the Simulated Annealing (SA) algorithm is applied as optimization in DCT steganography. The goal is to find the optimal parameter configuration that can maintain the quality of the steganography image while storing the message efficiently. This research uses 6 RGB and grayscale images each and 3 messages with different capacities. Evaluation is carried out by measuring the Peak Signal-to-Noise Ratio (PSNR) and Computational Cost values. This test obtained good results seen from the PSNR value with the highest value of 57,885 dB for RGB images and 58,402 dB for grayscale images, however testing the insertion process time had an influence on the SA algorithm which required quite a long computing time with 16 seconds for the insertion process. It is hoped that the results of this research will provide additional insight into the development of better and more effective information hiding techniques.

Keywords: *Steganography, Discrete Cosine Transform, Simulated Annealing, Digital Image, Peak Signal-to-Noise Ratio*

1. Pendahuluan

Teknologi yang terus berkembang saat ini memiliki pengaruh yang sangat besar terhadap berbagai aspek kehidupan manusia, salah satunya yaitu menjaga keamanan data yang bersifat rahasia. Salah satu teknik yang digunakan untuk menyembunyikan informasi yaitu steganografi. Proses penyembunyian pesan rahasia menggunakan sistem steganografi dapat dilakukan dengan menggunakan media gambar, audio, maupun video [1]. Pada implementasinya, steganografi memiliki banyak metode untuk menyembunyikan pesan rahasia, salah satunya yaitu metode *Discrete Cosine Transform* (DCT), metode ini mengubah sebuah sinyal ke dalam komponen frekuensi dasar [2]. Namun, semakin banyak karakter yang disisipkan pada gambar akan semakin turun nilai PSNR yang didapatkan atau kualitas gambarnya semakin menurun, maka dari itu algoritma optimasi juga perlu diimplementasikan untuk mempertahankan kualitas citra stego yang dihasilkan, salah satu algoritma optimasi yang dapat digunakan yaitu algoritma *Simulated Annealing* (SA), algoritma ini digunakan untuk menyelesaikan masalah yang dimana memerlukan ruang yang sangat luas serta perubahan keadaan dari suatu kondisi ke kondisi yang lainnya [3].

Penelitian mengenai steganografi menggunakan metode *Discrete Cosine Transform* pernah dilakukan oleh beberapa peneliti sebelumnya seperti [4] melakukan steganografi dengan menyisipkan citra (informasi rahasia) ke dalam sebuah citra menggunakan metode DCT yang

menyimpulkan bahwa resolusi gambar akan mempengaruhi penyisipan, waktu pengiriman, dan waktu ekstraksi. Nilai rata-rata PSNR 43,778 dB dan MSE 2,733. Waktu pengiriman paling lama terjadi pada pengiriman citra 48x48 piksel sebesar 13,668 detik. Penelitian ini mengimplementasikan metode *Discrete Cosine Transform* (DCT) dengan tambahan algoritma *Simulated Annealing* (SA) untuk meningkatkan keamanan dan kualitas citra steganografi. Data password dienkripsi terlebih dahulu untuk menambah lapisan keamanan. Setelah penyisipan pesan tersembunyi menggunakan DCT, optimasi dilakukan dengan algoritma SA untuk mencari konfigurasi optimal dari parameter penyisipan, memastikan data sensitif tetap terlindungi dari pihak yang tidak berwenang.

2. Metodologi Penelitian

2.1. Data dan Pengumpulan Data

Data yang digunakan pada penelitian ini yaitu data citra dan teks dengan metode pengumpulan data sekunder, data citra dengan format *.jpeg digunakan sebagai penampung pesan rahasia, sedangkan data teks dengan format *.txt digunakan sebagai pesan yang akan disisipkan ke dalam citra. Terdapat 6 citra digital digunakan yang memiliki ukuran dan jenis yang berbeda (RGB dan *grayscale*). Citra sebagai *cover-image* diambil dari situs web [pinterest.com](https://www.pinterest.com) dengan kata kunci "*digital art*". File teks sebagai pesan yang digunakan memiliki jumlah karakter yang berbeda, dengan penambahan 100 karakter pada setiap file teks berikutnya. Data teks yang digunakan yaitu berupa nilai hash SHA-512 dari sebuah *master password* dari *password manager*. Daftar spesifikasi citra dan pesan dapat dilihat pada Tabel 1 sampai dengan Tabel 3.

Tabel 1. Data Citra RGB

Nama File	Ukuran
rgb1.jpeg	74 KB
rgb2.jpeg	126 KB
rgb3.jpeg	151 KB
rgb4.jpeg	210 KB
rgb5.jpeg	373 KB
rgb6.jpeg	452 KB

Tabel 2. Data Citra *Grayscale*

Nama File	Ukuran
gray1.jpeg	62 KB
gray2.jpeg	100 KB
gray3.jpeg	129 KB
gray4.jpeg	190 KB
gray5.jpeg	332 KB
gray6.jpeg	385 KB

Tabel 3. Data Pesan Rahasia

Nama File	Ukuran
message1.txt	102 chars
message2.txt	203 chars
message3.txt	302 chars

2.2. *Discrete Cosine Transform*

Discrete Cosine Transform merupakan salah satu metode steganografi pada citra digital yang menggunakan teknik transform domain, metode ini mentransformasikan sebuah informasi dari domain ruang atau waktu ke dalam domain frekuensi. Metode DCT adalah perubahan basis yang mengambil fungsi yang bernilai riil dan mengubahnya dalam bentuk basis ortonormal cosinus.

Metode DCT yang banyak digunakan dalam aplikasi adalah DCT 2D. Persamaan untuk transformasi DCT 2D (citra berukuran $m \times n$) ditunjukkan pada persamaan (1) dan (2)[5]:

$$T(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \cos \frac{(2x+1)u\pi}{2N} \cos \cos \frac{(2y+1)v\pi}{2N} \quad (1)$$

dimana $\alpha(u) = \sqrt{\frac{1}{N}}$ untuk $u=0$

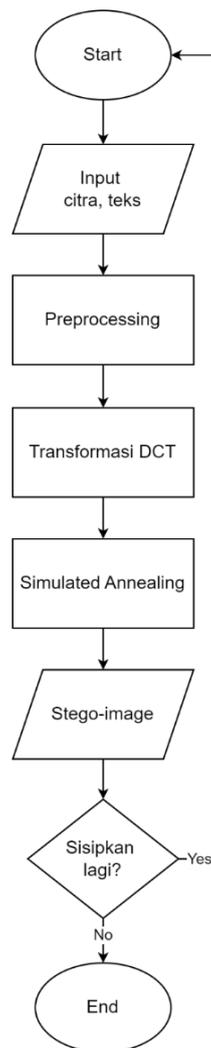
$$\alpha(u) = \sqrt{\frac{2}{N}}$$
 untuk $u = 1, 2, 3, \dots, n-1$

Untuk invers dari transformasi 2D DCT dapat dilihat pada persamaan di bawah ini:

$$f(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \alpha(u)\alpha(v)T(x, y) \cos \cos \frac{(2x+1)u\pi}{2N} \cos \cos \frac{(2y+1)v\pi}{2N} \quad (2)$$

$$\alpha(u) = \sqrt{2/N}$$
 untuk $u = 1, 2, 3, \dots, n-1$

a. Proses Penyisipan

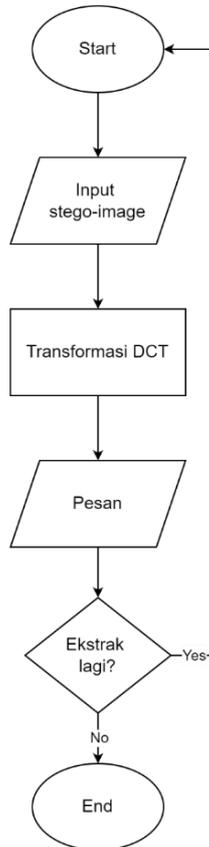


Gambar 1. Flowchart Penyisipan

Penyisipan merupakan salah satu fitur utama dari sistem ini, pertama sistem menerima input berupa masukan citra dan pesan yang ingin disisipkan, kemudian selanjutnya dilakukan *preprocessing* yang dimana proses tersebut dilakukan dengan mengkonversi gambar ke dalam domain DCT, kemudian proses transformasi dilakukan untuk pemilihan

blok gambar yang akan digunakan, pembentukan nilai koefisien untuk disisipkan pesan, kemudian algoritma *simulated annealing* untuk melakukan optimasi pemilihan koefisien yang cocok dan memiliki nilai koefisien secara minimal yang kemudian disisipkan pesan tersebut, proses invers matriks dilakukan untuk mengembalikan blok gambar menjadi *stego-image*. *Flowchart* penyisipan dapat dilihat pada Gambar 1.

b. Proses Ekstraksi



Gambar 2. *Flowchart* Ekstraksi

Ekstraksi merupakan salah satu fitur utama dari sistem ini, pertama sistem menerima input berupa masukan citra yang ingin diekstraksi, kemudian selanjutnya dilakukan *preprocessing* yang dimana proses tersebut dilakukan dengan mengkonversi gambar ke dalam domain DCT, kemudian proses transformasi dilakukan untuk pemilihan blok gambar yang akan, kemudian pesan tersebut diekstraksi dari *stego-image*, pesan-pesan tersebut akan disimpan menjadi file baru. *Flowchart* penyisipan dapat dilihat pada Gambar 2.

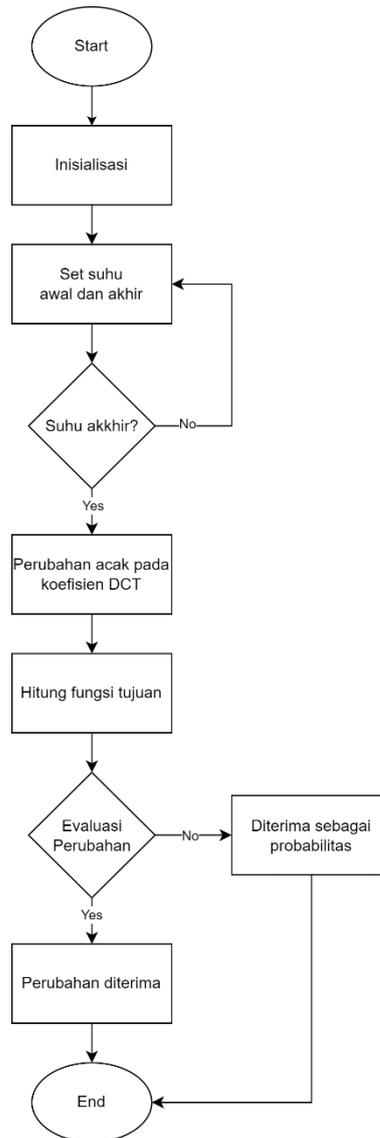
2.3. Simulated Annealing

Simulated Annealing adalah meta-algoritma probabilistik generik untuk masalah optimasi global. SA telah terbukti kuat dan mampu menangani data dunia nyata yang *noise* dan tidak lengkap. Dimulai dari titik yang dipilih secara acak dalam ruang pencarian. Algoritma dimulai dengan suhu tinggi yang kemudian dikurangi secara perlahan, biasanya secara bertahap. Pada setiap langkah, suhu harus dipertahankan konstan untuk periode waktu yang tepat (yaitu jumlah iterasi) agar algoritma dapat menetap dalam kesetimbangan termal, yaitu keadaan seimbang [6].

Dengan analogi dengan proses metalurgi, setiap langkah dari algoritma SA menggantikan solusi aktual dengan solusi yang dihasilkan secara acak dari lingkungan, dipilih dengan probabilitas tergantung pada perbedaan antara nilai fungsi yang sesuai dan pada parameter global, yang disebut suhu - T. Suhu menurun selama proses. Solusi saat ini berubah hampir secara acak ketika T besar, tetapi semakin menurun ketika T menjadi nol. Penyisihan untuk gerakan menanjak

menyelamatkan metode agar tidak macet di minimum lokal. *Simulated Annealing* adalah algoritma stokastik tergantung pada parameter yang dapat dilihat pada persamaan (3):

$$SA = M, x_0, N, f, T_0, T_f, \alpha, n_T \quad (3)$$



Gambar 3. Flowchart Algoritma *Simulated Annealing*

Simulated annealing merupakan algoritma yang digunakan untuk mengoptimasi penyisipan agar memiliki hasil *stego-image* yang lebih optimal, proses pertama yaitu mengatur suhu awal dan suhu akhir, kemudian melakukan iterasi hingga suhu mencapai suhu akhir dengan melakukan perulangan, jika suhu sudah mencapai akhir maka perubahan acak dilakukan pada koefisien DCT yang dipilih untuk menyembunyikan pesan, kemudian hitung fungsi tujuan untuk mengevaluasi kualitas pesan, pada evaluasi perubahan, jika perubahan meningkatkan kualitas pesan tersembunyi, maka perubahan diterima. Jika tidak, perubahan diterima dengan probabilitas tertentu yang dipengaruhi oleh suhu dan perbedaan nilai fungsi tujuan, langkah yang terakhir yaitu menurunkan suhu dengan faktor pendinginan. Flowchart algoritma *simulated annealing* dapat dilihat pada Gambar 3.

2.4. Peak Signal to Noise Ratio (PSNR)

Penelitian ini menggunakan pengujian dengan parameter *Peak Square Noise Ratio* dan *Mean Square Error*, PSNR dapat didefinisikan sebagai perkiraan kualitas gambar secara statistik digunakan untuk mengukur distorsi antara cover dan *stego-image* dengan membandingkan

antara nilai maksimum dari sinyal yang diukur dengan besarnya noise dalam satuan desibel (dB). MSE merupakan adalah kesalahan kuadrat rata-rata sinyal-sinyal *pixel* citra hasil pemrosesan sinyal terhadap sinyal asli [7]. Persamaan yang digunakan untuk menentukan nilai berdasarkan PSNR dapat dilihat pada persamaan (4) dan (5):

$$PSNR = 10 \log_{10} \frac{255}{MSE} \quad (4)$$

Dimana nilai MSE didapatkan dengan persamaan berikut:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (S(i, j) - C(i, j))^2 \quad (5)$$

Keterangan:

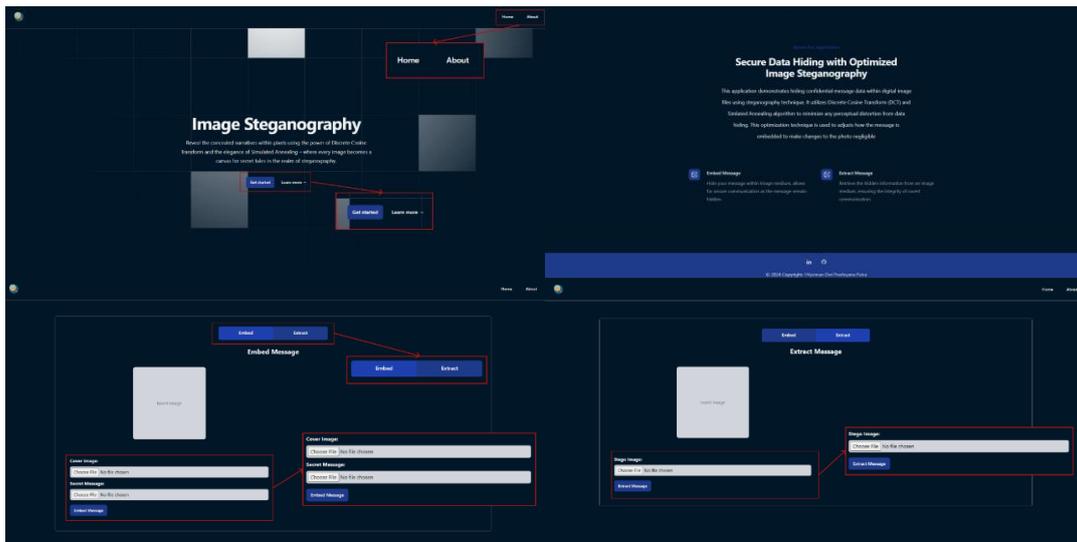
- m : Baris matriks citra hasil pemrosesan
- n : Kolom matriks citra hasil pemrosesan
- S : Pixel citra asli
- C : Pixel citra hasil pemrosesan

Pengujian ini bertujuan untuk mengetahui perbedaan citra setelah disisipkan sebuah pesan dilihat dari jangkauan PSNR berdasarkan pada persamaan (4). Sistem akan menyesuaikan panjang karakter maksimal yang disisipkan dilihat dari nilai PSNR pada gambar yang dipakai. Semakin besar nilai PSNR dan semakin kecil nilai MSE yang didapatkan, maka citra tersebut dapat dikatakan penyisipan dapat dilakukan dengan baik, kualitas citra terbaik berdasarkan nilai MSE ketika mendekati 0, kualitas citra dikatakan baik/sedang apabila memiliki nilai PSNR antara 30 hingga 40 dB, sangat baik > 40 dB, dan kurang baik apabila < 30 dB [8].

3. Hasil dan Pembahasan

3.1. Hasil Desain Sistem

Pada tahap ini, tampilan desain sistem yang dirancang sebelumnya berupa wireframe akan diimplementasikan menggunakan HTML, CSS, dan *Tailwind* CSS. Berikut beberapa tampilan halaman pada website steganografi yang telah diimplementasikan.



Gambar 4. Hasil Desain Sistem

Berdasarkan Gambar 4 terdapat 4 buah halaman yang diimplementasikan, halaman pertama yaitu halaman *Home* yang berisi sebuah *navigation bar* yang mengarah ke halaman *home* dan *about*, selain itu halaman ini menampilkan pengenalan aplikasi web serta terdapat tombol “Get Started” yang nantinya akan mengarah ke halaman penyisipan. Halaman kedua yaitu halaman *about* yang memuat informasi mengenai aplikasi web, fitur yang tersedia, serta *footer* yang memuat informasi pengembang aplikasi.

Halaman ketiga yaitu penyisipan, terdapat 2 menu yang berada di tengah halaman yang menunjukkan fitur yang ingin digunakan. Kemudian terdapat *form* untuk memasukan file citra dan pesan untuk dilakukan penyisipan, serta sistem akan menampilkan gambar yang dimasukan pada halaman web. Halaman terakhir yaitu halaman ekstraksi, tampilan ini tidak jauh berbeda dengan halaman penyisipan, perbedaannya terdapat pada form input file yang hanya satu untuk memasukan citra *stego-image*.

3.2. Pengujian PSNR

Pengujian PSNR digunakan untuk mengukur nilai perbedaan citra antara *cover-image* dengan *stego-image*, berikut penggalan kode Python yang digunakan pada pengujian hasil steganografi.

Tabel 4. Hasil Pengujian PSNR Steganografi DCT

Citra	PSNR (dB)		
	message1.txt	message2.txt	message3.txt
rgb1.jpeg	32.697	32.625	32.540
rgb2.jpeg	33.963	33.931	33.904
rgb3.jpeg	33.985	33.962	33.940
rgb4.jpeg	35.340	35.216	35.165
rgb5.jpeg	35.882	35.854	35.823
rgb6.jpeg	38.586	38.565	38.552
gray1.jpeg	33.781	33.675	33.615
gray2.jpeg	37.851	37.756	37.687
gray3.jpeg	35.180	35.156	35.128
gray4.jpeg	35.183	35.151	35.114
gray5.jpeg	37.074	37.031	36.997
gray6.jpeg	40.970	40.948	40.932

Berdasarkan Tabel 4 menunjukkan bahwa metode steganografi DCT cukup efektif dalam menyisipkan pesan, konsistensi nilai PSNR terhadap 3 pesan yang berbeda menunjukkan bahwa metode yang digunakan stabil dan tidak banyak merusak kualitas gambar. Nilai PSNR tertinggi pada rgb6.jpeg pada pesan 1, dan gray6.jpeg pada pesan 1, hal ini disebabkan oleh ukuran citra yang semakin besar akan memberikan hasil PSNR lebih tinggi tergantung seberapa banyak pesan yang disisipkan.

Tabel 5. Hasil Pengujian PSNR Steganografi DCT dan SA

Citra	PSNR (dB)		
	message1.txt	message2.txt	message3.txt
rgb1.jpeg	44.836	41.886	40.221
rgb2.jpeg	46.366	44.041	42.806
rgb3.jpeg	50.238	46.791	45.072
rgb4.jpeg	50.468	47.663	46.028
rgb5.jpeg	52.129	49.067	47.379
rgb6.jpeg	57.885	55.120	53.443
gray1.jpeg	45.392	42.695	41.059
gray2.jpeg	50.262	47.818	46.094
gray3.jpeg	51.041	47.886	46.019
gray4.jpeg	51.710	48.364	46.661
gray5.jpeg	54.026	50.866	49.330
gray6.jpeg	58.402	56.021	54.862

Berdasarkan Tabel 5 hasil PSNR setelah proses penyisipan menggunakan algoritma *simulated annealing* bahwa teknik steganografi ini dapat meminimalkan noise pada stego-image yang dimana hasil PSNR menunjukan di atas 40 dB dengan nilai tertinggi 57.885 dB untuk *rgb6.jpeg* terhadap *message1.txt* dan 58.402 dB untuk *gray6.jpeg* terhadap *message1.txt*, citra yang akan disisipkan pesan sangat berpengaruh terhadap jumlah karakter atau ukuran bit pada pesan, dimana semakin besar pesan, maka nilai PSNR semakin sedikit.

Secara umum, citra *grayscale* cenderung memiliki PSNR yang lebih tinggi dibandingkan dengan citra berwarna untuk pesan yang sama. Misalnya, *gray6.jpeg* memiliki PSNR 58.402 dB, lebih tinggi daripada *rgb6.jpeg* yang memiliki PSNR 57.885 dB untuk *message1.txt*. Hal ini disebabkan karena citra *grayscale* lebih sederhana dan memiliki informasi yang lebih sedikit dibandingkan dengan citra berwarna sehingga lebih sedikit gangguan yang terjadi ketika pesan disisipkan.

3.3. Pengujian Waktu Proses

Pengujian waktu proses pada sistem ini digunakan untuk mengukur seberapa cepat metode steganografi dapat melakukan proses penyisipan dan ekstraksi, berikut merupakan potongan kode Python yang digunakan pada pengujian ini.

Tabel 6. Hasil Waktu Proses Penyisipan Steganografi DCT

Citra	Waktu Proses (s)		
	message1.txt	message2.txt	message3.txt
<i>rgb1.jpeg</i>	0.410	0.421	0.438
<i>rgb2.jpeg</i>	1.124	1.187	1.205
<i>rgb3.jpeg</i>	1.547	1.602	1.629
<i>rgb4.jpeg</i>	2.891	2.935	2.978
<i>rgb5.jpeg</i>	3.102	3.284	3.391
<i>rgb6.jpeg</i>	4.521	4.724	4.809
<i>gray1.jpeg</i>	0.407	0.416	0.462
<i>gray2.jpeg</i>	0.934	0.952	0.963
<i>gray3.jpeg</i>	1.518	1.641	1.677
<i>gray4.jpeg</i>	2.837	2.910	2.989
<i>gray5.jpeg</i>	3.059	3.245	3.412
<i>gray6.jpeg</i>	4.211	4.349	4.612

Berdasarkan Tabel 6 menunjukan adanya sedikit konsistensi antara waktu proses *message1.txt*, *message2.txt*, dan *message3.txt* menunjukkan bahwa metode steganografi DCT memiliki efisiensi yang serupa terlepas dari pesan yang disisipkan. Hasil waktu proses penyisipan yang dimana proses penyisipan paling lama hingga 4.8 detik yaitu pada citra dengan ukuran terbesar, dan penyisipan paling cepat pada citra dengan ukuran terkecil dengan lama 0.4 detik.

Tabel 7. Hasil Waktu Proses Ekstraksi Steganografi DCT

Citra	Waktu Proses (s)		
	message1.txt	message2.txt	message3.txt
<i>rgb1.jpeg</i>	0.002	0.003	0.005
<i>rgb2.jpeg</i>	0.009	0.011	0.016
<i>rgb3.jpeg</i>	0.014	0.021	0.024
<i>rgb4.jpeg</i>	0.019	0.025	0.034
<i>rgb5.jpeg</i>	0.036	0.048	0.052
<i>rgb6.jpeg</i>	0.070	0.089	0.100
<i>gray1.jpeg</i>	0.001	0.002	0.004
<i>gray2.jpeg</i>	0.004	0.012	0.015
<i>gray3.jpeg</i>	0.009	0.018	0.025

gray4.jpeg	0.016	0.020	0.027
gray5.jpeg	0.029	0.033	0.040
gray6.jpeg	0.066	0.087	0.096

Tabel 7 menunjukkan waktu proses ekstraksi pesan pada setiap citra hanya menggunakan metode DCT. Waktu proses ekstraksi untuk citra RGB dan *grayscale* memiliki perbedaan yang tidak begitu jauh. Waktu proses ekstraksi ini tergantung pada ukuran dan kompleksitas citra, dengan citra yang lebih kompleks atau berukuran lebih besar cenderung membutuhkan waktu lebih lama. Secara keseluruhan, waktu proses ekstraksi berlangsung sekitar dari 0.001 detik hingga 0.100 detik.

Tabel 8. Hasil Waktu Proses Penyisipan Steganografi DCT dan SA

Citra	Waktu Proses (s)		
	message1.txt	message2.txt	message3.txt
rgb1.jpeg	0.476	0.486	0.530
rgb2.jpeg	1.266	1.314	1.348
rgb3.jpeg	1.641	1.697	1.806
rgb4.jpeg	3.142	3.186	3.369
rgb5.jpeg	7.608	8.085	8.594
rgb6.jpeg	15.481	15.723	16.111
gray1.jpeg	0.401	0.612	0.890
gray2.jpeg	0.863	0.935	0.987
gray3.jpeg	1.575	1.602	1.684
gray4.jpeg	2.911	2.970	3.213
gray5.jpeg	7.587	7.743	8.180
gray6.jpeg	13.399	13.485	13.733

Tabel 8 menunjukkan hasil waktu proses penyisipan pada setiap citra dan pesan. Citra RGB memerlukan waktu lebih lama daripada *grayscale* untuk proses penyisipannya. Waktu proses penyisipan cenderung meningkat seiring dengan ukuran dan kompleksitas citra. Citra dengan waktu yang lebih besar seperti rgb6.jpeg dan gray6.jpeg memiliki waktu proses yang paling lama, menunjukkan bahwa citra tersebut lebih kompleks atau berukuran lebih besar.

Faktor lain yaitu karena adanya tambahan algoritma SA yang melibatkan analisis dan modifikasi pixel yang lebih kompleks. Proses identifikasi area-area ini memerlukan waktu pemrosesan tambahan yang mengakibatkan lamanya waktu proses penyisipan.

Tabel 9. Hasil Waktu Proses Ekstraksi Steganografi DCT dan SA

Citra	Waktu Proses (s)		
	message1.txt	message2.txt	message3.txt
rgb1.jpeg	0.002	0.004	0.006
rgb2.jpeg	0.010	0.014	0.018
rgb3.jpeg	0.018	0.023	0.027
rgb4.jpeg	0.020	0.029	0.035
rgb5.jpeg	0.031	0.036	0.043
rgb6.jpeg	0.072	0.095	0.102
gray1.jpeg	0.001	0.004	0.005
gray2.jpeg	0.006	0.014	0.017
gray3.jpeg	0.010	0.021	0.027
gray4.jpeg	0.017	0.023	0.029
gray5.jpeg	0.038	0.052	0.054

gray6.jpeg	0.070	0.090	0.100
------------	-------	-------	-------

Pada Tabel 9 menunjukkan waktu proses ekstraksi pesan pada setiap citra. Waktu proses ekstraksi untuk citra RGB dan *grayscale* bervariasi, tetapi tidak menunjukkan perbedaan pola yang signifikan. Waktu proses ekstraksi ini tergantung pada ukuran dan kompleksitas citra, dengan citra yang lebih kompleks atau berukuran lebih besar cenderung membutuhkan waktu lebih lama. Secara keseluruhan, waktu proses ekstraksi sekitar dari 0.001 detik hingga 0.102 detik, menunjukkan bahwa proses ekstraksi cukup cepat untuk semua citra dan pesan yang diuji. Grafik untuk pengujian waktu proses ekstraksi dapat dilihat pada tabel di bawah ini.

3.4. Pengujian *Black-box*

Pengujian *black-box* dilakukan untuk menguji fungsi yang terdapat pada aplikasi yang telah dibangun. Pengujian ini dilakukan dengan cara melakukan demonstrasi menggunakan aplikasi sebagai user untuk mengetahui apakah fungsi yang terdapat pada aplikasi dapat berjalan dengan baik sesuai dengan skenario.

Tabel 10. Hasil Pengujian *Black-box* Proses Penyisipan

Nama Pengujian	Pengujian Proses Penyisipan		
Tujuan	Menguji Fungsi Proses Penyisipan		
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu penyisipan 2. Pengguna memasukan citra sebagai <i>cover-image</i> 3. Pengguna memasukan file teks sebagai pesan 4. Pengguna menekan tombol <i>Embed</i> 5. Pengguna men-download <i>stego-image</i> 		
Hasil Pengujian			
Masukan Pengguna	Keluaran yang diharapkan	Pengamatan	Kesimpulan
<ol style="list-style-type: none"> 1. Citra digital 2. File teks 	<ol style="list-style-type: none"> 1. Dapat menampilkan citra yang diinput 2. Dapat menghasilkan <i>stego-image</i> 3. Dapat men-download <i>stego-image</i> 	<ol style="list-style-type: none"> 1. Menampilkan citra yang diinput 2. Menghasilkan <i>stego-image</i> 3. <i>Stego-image</i> dapat di-download 	Sesuai

Pada tabel 10 dapat dilihat hasil yang diharapkan atau skenario oleh sistem sudah sesuai dengan hasil pengamatan yang dilakukan terhadap sistem dengan melakukan demonstrasi. Hal ini menunjukkan bahwa proses penyisipan telah berjalan sesuai skenario dan mendapatkan hasil yang diharapkan.

Tabel 11. Hasil Pengujian *Black-box* Proses Ekstraksi

Nama Pengujian	Pengujian Proses Ekstraksi		
Tujuan	Menguji Fungsi Proses Ekstraksi		
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu ekstraksi 2. Pengguna memasukan citra sebagai <i>stego-image</i> 3. Pengguna menekan tombol <i>Extract</i> 4. Pengguna men-download teks yang telah diekstraksi 		
Hasil Pengujian			
Masukan Pengguna	Keluaran yang diharapkan	Pengamatan	Kesimpulan

1. Citra digital	1. Dapat menampilkan citra yang diinput	1. Menampilkan citra yang diinput	Sesuai
	2. Dapat menampilkan pesan yang diekstraksi	2. Menampilkan pesan yang diekstraksi	
	3. Dapat men-download pesan yang diekstraksi	3. Teks yang telah diekstraksi dapat di-download	

Pada tabel 11 dapat dilihat hasil yang diharapkan atau skenario oleh sistem sudah sesuai dengan hasil pengamatan yang dilakukan terhadap sistem dengan melakukan demonstrasi. Hal ini menunjukkan bahwa proses ekstraksi telah berjalan sesuai skenario dan mendapatkan hasil yang diharapkan.

4. Simpulan

Berdasarkan dari penelitian yang telah dilakukan pada implementasi algoritma *simulated annealing* dalam steganografi *discrete cosine transform* dapat disimpulkan bahwa:

- Penerapan algoritma *simulated annealing* dalam steganografi metode *discrete cosine transform* (DCT) menunjukkan peningkatan yang signifikan pada nilai *peak signal-to-noise ratio* (PSNR) dibandingkan hanya menggunakan metode DCT. Berdasarkan pengujian, rata-rata nilai PSNR yang dihasilkan yaitu 48 dB dengan nilai tertinggi 57 dB pada citra RGB dan 58 dB pada citra *grayscale* serta nilai terendah 40 dB pada citra RGB dan 41 dB citra *grayscale*. Hal ini menunjukkan bahwa algoritma *simulated annealing* efektif dalam meminimalkan distorsi visual atau noise yang diakibatkan oleh proses penyisipan pesan pada gambar sehingga kualitas *stego-image* lebih tinggi dibandingkan dengan metode tanpa optimasi.
- Penggunaan algoritma *simulated annealing* dalam steganografi DCT memberikan dampak pada meningkatnya *computational cost* terkhusus pada waktu proses penyisipan. Berdasarkan pengujian, waktu proses yang dihasilkan pada proses penyisipan dengan waktu terlama yaitu 15 detik pada citra RGB dan 13 detik pada citra *grayscale* serta waktu tercepat 0.41 detik pada citra RGB dan 0.40 detik pada citra *grayscale*. Proses yang dilakukan oleh algoritma ini membutuhkan waktu dan sumber daya komputasi yang lebih besar. Namun, penggunaan algoritma ini sebanding dengan peningkatan kualitas gambar yang dihasilkan.

Referensi

- [1] S. Nur'aini, "Steganografi Pada Digital Image Menggunakan Metode Least Significant Bit Insertion," *Walisongo Journal of Information Technology*, vol. 1, no. 1, p. 73, 2019
- [2] T Yuniati, "IMPLEMENTASI AUDIO STEGANOGRAFI MENGGUNAKAN ALGORITMA DISCRETE COSINE TRANSFORM Tugas Akhir," *ejurnal.teknokrat.ac.id*, vol. 17, no. 1, 2023.
- [3] R. Hidayati, I. Guntero, and S. Junianti, "PENGUNAAN METODE SIMULATED ANNEALING UNTUK PENYELESAIAN TRAVELLING SALESMAN PROBLEM," 2019.
- [4] M. Hamdani, D. Gloria, and N. Samosir, "IMPLEMENTASI STEGANOGRAFI UNTUK KEAMANAN PENGIRIMAN CITRA DIGITAL MENGGUNAKAN METODA DCT (DISCRETE COSINE TRANSFORM)," 2018.
- [5] U. Ramadass, "FPGA Implementation of 2-D DCT for JPEG Image Compression," 2011, [Online]. Available: <https://www.researchgate.net/publication/228615235>

- [6] G. R. Karimi and A. A. Verki, *Simulated Annealing - Single and Multiple Objective Problems*. Rijeka, Croatia: InTech, 2012.
- [7] M. A. H. M. H. A. M. Hassaballah, *Digital Media Steganography Principles, Algorithms, and Advances*. Elsevier Science, 2020.
- [8] N. Sari and R. Rosnelly, "Sistem Perbaikan Citra Berkas Pelanggan Kredit Sepeda Motor Menggunakan Metode Gaussian Filter Dan Metode High-Pass Filter Motorcycle Credit Customer File Image Improvement System Using Gaussian Filter Method and High-Pass Filter Method," *Januari*, no. 2, p. 143, 2024.