

## IMAGE WATERMARKING BERBASIS PENGINDERAAN KOMPRESIF MENGGUNAKAN SPEKTRAL TERSEBAR DAN TRANSFORMASI WAVELET

### COMPRESSIVE SENSING IMAGE WATERMARKING USING SPREAD SPECTRUM AND WAVELET TRANSFORM

Rangga Wahyuning Gusty<sup>1</sup>, Dr. Ida Wahidah, S.T., M.T.<sup>2</sup>, Irma Safitri, S.T, M.Sc.<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom  
Jln. Telekomunikasi No.1 Terusan Buah Batu Bandung 40257 Indonesia

<sup>1</sup>[ranggawahyuning@gmail.com](mailto:ranggawahyuning@gmail.com), <sup>2</sup>[idawahidah@telkomuniversity.ac.id](mailto:idawahidah@telkomuniversity.ac.id), <sup>3</sup>[irmasaf@telkomuniversity.ac.id](mailto:irmasaf@telkomuniversity.ac.id)

#### ABSTRAK

*Watermarking* bertujuan untuk melindungi hak cipta pada suatu karya dari hal-hal yang merugikan seperti pembajakan, penggandaan masal, dan lain-lain. CS (*Compressive Sensing*) bertentangan dengan kebijakan umum dalam akuisisi data yang digunakan untuk *sampling* berdasarkan teori Shannon pada metode tradisional. Dengan menggunakan teknik CS, sinyal dapat direkonstruksi dengan menggunakan *sample* yang jauh lebih sedikit dibandingkan pada metode tradisional. SS (*Spread Spectrum*) mengklaim dapat tahan terhadap beberapa serangan karena adanya penyebaran pada bit *watermark* yang memungkinkan bit asli pada citra *watermark* tidak terkena serangan. Proses *embedding* dan *extraction* pada penelitian ini dilakukan pendekatan DWT (*Discrete Wavelet Transform*) dan SS (*Spread Spectrum*). CS digunakan untuk menghilangkan ketidakefisienan saat pengumpulan data dengan mereduksi dimensi citra *watermark*. *Sparsity* pada citra *watermark* menggunakan metode DCT yang kemudian dilakukan pengukuran berdasarkan distribusi Gaussian. Selanjutnya *watermark* diletakkan pada koefisien citra *host* yang didapatkan dari hasil transformasi DWT dengan metode SS. Berdasarkan hasil pengukuran, citra *watermark* direkonstruksi dengan menggunakan BP yang tersedia dalam paket  $l_1$ -*magic*. Hasil penelitian dari tugas akhir ini berupa aplikasi *watermarking* pada MATLAB. CS berbasis DWT menggunakan SS dapat meningkatkan *robustness* dilihat dari nilai BER 0% dengan menggunakan CS, dan BER 19,63% tanpa menggunakan CS, dengan nilai PSNR yang relatif sama karena bit *watermark* disisipkan berulang-ulang ke dalam citra *host*. Dalam ruang warna RGB dihasilkan PSNR yang lebih baik dibandingkan YCbCr, namun YCbCr lebih tahan terhadap serangan.

**Kata kunci :** *Image Watermarking, Compressive Sensing, Basis Pursuit, Discrete Cosine Transform, Discrete Wavelet Transform, Spread Spectrum.*

#### ABSTRACT

*The purpose of watermarking method is to protect copyright from harmful things such as piracy, duplication, and others. CS (Compressive Sensing) replaces the general policy in data used for sampling based on Shannon's theory of traditional methods. Using CS techniques, signals can be reconstructed using fewer samples than traditional methods. SS (Spread Spectrum) claims that can robust some attacks because it has spread on the watermark bit that make the original bits on the watermark image not to be attacked. The process of embedding and extraction in this research was carried out on DWT (Discrete Wavelet Transform) and SS (Spread Spectrum). CS is used to eliminate inefficiencies when retrieving data by reducing the dimensions of a watermark image. DCT method is used for sparsity transform and then measured based on the Gaussian distribution. Then the watermark is placed on the host coefficient obtained from the DWT transformation results with the SS method. Based on the measurement results, the watermark image is reconstructed using BP in  $l_1$ -magic package. The results of this final project are watermark application in MATLAB. DWT-based CS using SS can increase robustness seen from the value of BER 0% using CS, and BER 19.63% without CS, with the PSNR value relatively the same because the watermark bit inserted repeatedly into the host image. In RGB color space has better PSNR value than YCbCr, but YCbCr is more robust to attacks.*

**Keywords :** *Image Watermarking, Compressive Sensing, Basis Pursuit, Discrete Cosine Transform, Discrete Wavelet Transform, Spread Spectrum.*

#### 1. PENDAHULUAN

Penggunaan komputer dan smartphone di era globalisasi saat ini semakin meluas, setiap orang dapat dengan mudah dan cepat mengakses berbagai informasi seperti pengunduhan konten digital. Kemudahan pengunduhan konten digital seperti teks, gambar, suara, video, dan lain-lain dapat memicu terjadi masalah-masalah tentang hak cipta atau *copyright*. Salah satu cara agar dapat melindungi hak cipta tersebut adalah dengan menyisipkan

informasi yang dapat melindungi hak cipta tentang kepemilikan konten digital tersebut dengan teknik watermarking.

CS (*Compressive Sensing*) merupakan teknik yang di klaim dapat melakukan reduksi dimensi sehingga dapat meng-kompres sinyal untuk mengurangi *sampling rate* pada besarnya tingkat kompleksitas komputasi dalam proses rekonstruksi [3]. Dalam CS, sampel diambil dalam jumlah sedikit yang bertentangan dengan kebijakan umum dalam akuisisi data pada metode tradisional yang digunakan untuk *sampling* berdasarkan teori Shannon. Sinyal dapat direkonstruksi dengan menggunakan sampel yang jauh lebih sedikit dibandingkan yang digunakan oleh metode tradisional berdasarkan perhitungan yang disebut *sensing matrix* dan sinyalnya harus bersifat *sparse* atau penjarang [4]. *Sparse image* mengandung beberapa koefisien *non-zero* dan koefisien lainnya adalah *zero*. DWT (*Discrete Wavelet Transform*) untuk mentransformasikan citra standar menjadi *sparse*. Algoritma penyisipan berbasis domain frekuensi digunakan karena dalam domain frekuensi sangat sulit untuk mengekstrak konten digital yang telah disisipkan *watermark* tanpa mengubah kualitas konten tersebut [1].

Dalam tugas akhir ini dilakukan penelitian *image watermarking* dengan metode DWT dengan tipe *daubechies wavelet* untuk mengubah domain pada citra *host* dari domain waktu ke domain frekuensi, sedangkan pada citra *watermark* dilakukan pendekatan CS dengan transformasi proyeksi berdasarkan distribusi Gaussian yang sebelumnya dilakukan transformasi DCT untuk mengubah sinyal menjadi bersifat *sparse*. Kemudian SS menyebarkan bit-bit pada citra *watermark* agar *watermark* sulit terdeteksi, selanjutnya citra *watermark* disisipkan dengan citra *host*. Berdasarkan hasil pengukuran, citra *watermark* direkonstruksi dengan menggunakan BP.

## 2. TINJAUAN PUSTAKA

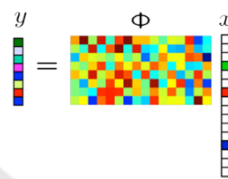
### 2.1 Image Watermarking

*Image watermarking* merupakan proses penyisipan informasi tambahan ke dalam sebuah citra, informasi yang disisipkan ini disebut *watermark*, secara umum *watermark* merupakan kode identifikasi baik terlihat maupun tidak terlihat dan dapat berisi informasi pemiliknya [6].

### 2.2 Compressive Sensing

CS (*Compressive Sensing*) atau penginderaan kompresif adalah metode kompresi yang diakibatkan adanya reduksi dimensi. Dalam prosesnya digunakan sampel dalam jumlah sedikit dan acak berdasarkan transformasi yang digunakan [6, 8]. Secara umum, ada dua transformasi penting dalam CS yaitu *sparsity transform* yang membuat suatu sinyal menjadi bersifat *sparse* dan *projection transform* yang merupakan transformasi pada suatu sinyal untuk di kompres atau di cuplik yang digunakan pada operasi pengukuran atau pengamatan.

Ketika data merupakan sinyal yang bersifat *sparse*, dapat langsung diperoleh representasi ringkasnya tanpa atau dengan sedikit kehilangan informasinya melalui reduksi dimensi linear [10]. Pada penelitian ini *sparse* sinyal dilakukan dengan mentransformasikan citra menggunakan DCT yang mampu memadatkan daya yang besar dan dapat menkonversikan citra dari domain spasial kedalam domain frekuensi.



Gambar 2.1. Compressive Sensing

Dari gambar 2.1 dapat dituliskan persamaan CS sebagai berikut,

$$y = \Phi x \quad (2.2)$$

dengan  $x$  merupakan sinyal *sparse* berukuran  $M \times 1$  ketika hampir seluruh elemen  $k$  dari  $x$  adalah *non-zero* dan  $k < M$ .  $\Phi$  merupakan matriks  $M \times N$  yang diperoleh dari hasil pengukuran *projection transform* dengan  $M \ll N$  kemudian dihasilkan  $y$  yang merupakan matriks  $M \times 1$  sebagai reduksi dimensi linear dari pengukuran  $\Phi$  dengan seluruh vektor  $x$ .

#### 2.2.1 Discrete Cosine Transform

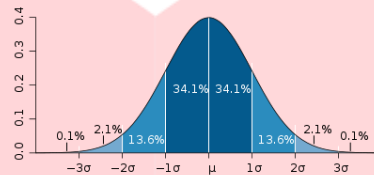
Discrete Cosine Transform (DCT) merupakan transformasi *bi-directional* yang dapat mengkonversikan citra dari domain spasial ke dalam domain frekuensi begitu sebaliknya selama *forward transform* dari citra terkonsentrasi ke dalam beberapa komponen frekuensi rendah [11].

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N} \quad (2.3)$$

$$\text{dengan } \alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M - 1 \end{cases} \quad \text{dan} \quad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N - 1 \end{cases} \quad (2.4)$$

### 2.2.2 Distribusi Gaussian

Distribusi Gaussian atau biasa disebut dengan distribusi normal digambarkan sebagai berikut.



Gambar 2.2. Kurva distribusi Gaussian

Hasil dari proyeksi Gaussian berupa matrik berukuran  $M \times N$  yang diperoleh dengan membangkitkan bilangan *random* berdasarkan distribusi Gaussian [8].

### 2.3 Basis Pursuit

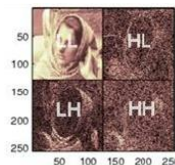
Metode basis pursuit dapat merumuskan pencarian basis ideal  $\varphi_M$  dalam kamus-basis  $\Phi$  berdimensi  $M \times N$  sebagai pemrograman [4, 8]. BP berhubungan dengan *Linear Programming* (LP) yang disebut sebagai bentuk standar dalam masalah optimasi terbatas dengan didefinisikan sebagai berikut,

$$\min \|x\|_1 \text{ subject to } Ax = y, \quad x \geq 0 \quad (2.5)$$

dengan  $\|x\|_1 = \sum_{i=1}^n |x_i|$ . Pada persamaan (2.5) dikenal sebagai masalah dalam BP, solusinya adalah vektor terkecil dari  $l_1$ -norm untuk menggambarkan  $y$  [11]. Pada  $l_1$ -magic didapatkan solusi pada persamaan (2.5) menggunakan teknik LP. Dalam tugas akhir ini, digunakan fungsi *l1eq\_pd* pada MATLAB yang merupakan algoritma *primal-dual* untuk LP.

### 2.4 Discrete Wavelet Transform

*Discrete Wavelet Transform* (DWT) adalah tahap dekomposisi dari transformasi wavelet, sinyal dalam domain waktu dilewatkan ke dalam *high-pass filter* dan *low-pass filter* untuk memisahkan komponen pada *sub-band* frekuensi tinggi dan frekuensi rendah. Berikut adalah visualisasi citra yang telah di dekomposisi dengan DWT:



Gambar 2.3. Visualisasi citra yang telah di dekomposisi dengan DWT.

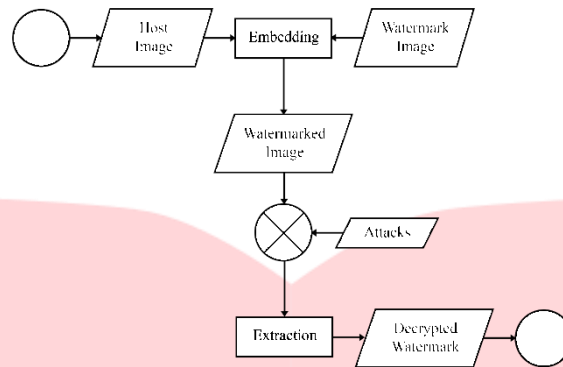
### 2.5 Spread Spectrum

*Spread Spectrum* (SS) merupakan sebuah teknik *watermarking* yang dilakukan dengan menyebarkan bit-bit *watermark* ke banyak komponen frekuensi. Secara umum, SS menghasilkan metode yang lebih tahan terhadap serangan seperti kompresi dan penapisan lolos-rendah, karena bit-bit *watermark* di sebar dalam ukuran kecil.

## 3. PERANCANGAN SISTEM

### 3.1. Desain Sistem

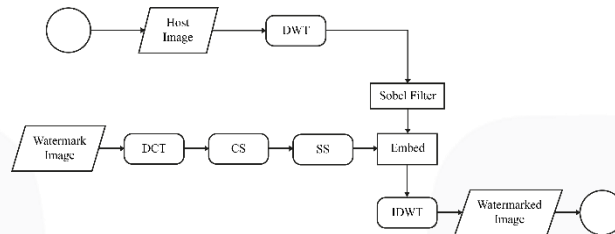
Secara umum, dalam *watermarking* terdapat 2 proses yaitu *embedding* dan *extraction*. Pada proses *embedding*, citra *watermark* disisipkan ke dalam citra *host* dan menghasilkan citra yang ter-*watermark*. Sedangkan *extraction* adalah proses ekstraksi citra ter-*watermark* dan menghasilkan citra *watermark* yang sudah terlepas. Sebelum proses ekstraksi, citra ter-*watermark* diberikan *attacks* untuk menguji kualitas *watermark*.



Gambar 3.1. Model Perancangan Penelitian

3.1.1 Embedding

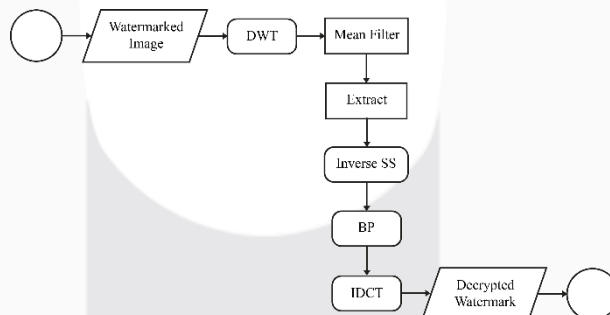
Proses *embedding* dilakukan dari citra *host* yang merupakan citra RGB berformat BMP, kemudian setelah citra melalui DWT untuk mengubah dari domain waktu ke domain frekuensi, citra *watermark* disisipkan ke dalam citra *host* dengan metode SS yang sebelumnya citra *watermark* di kompres dengan metode CS dengan DCT sebagai *sparsity transform*-nya. Setelah bit-bit citra *watermark* disisipkan pada domain *wavelet* dalam citra *host*, kemudian dilakukan *Inverse DWT* untuk mengembalikan domain citra dari domain frekuensi ke dalam domain waktu. Output dari IDWT merupakan citra yang telah disisipkan *watermark*.



Gambar 3.2. Proses embedding watermark dengan CS

3.1.2 Extraction

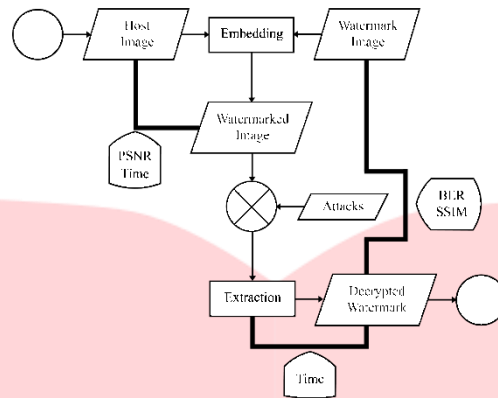
Proses *extraction* dilakukan dari citra RGB yang sudah disisipkan *watermark*, proses DWT untuk mengubah dari domain waktu ke dalam domain frekuensi. Berbeda dengan proses *embedding*, pada proses *extraction* dilakukan *inverse SS* untuk menghasilkan *watermark* yang sudah terdekripsi, kemudian BP digunakan untuk merekonstruksi dan membalikkan *sparsity transform*-nya dengan IDCT lalu menghasilkan *watermark* yang telah terdekripsi.



Gambar 3.3. Proses extraction watermark dengan CS

3.2 Parameter Input dan Output dari Simulasi

Tujuan dari penelitian ini adalah mengetahui kualitas *watermark* pada citra ter-*watermark* sebelum dan setelah diberikan serangan



Gambar 3.4. Parameter input dan output

Dalam pengujian kualitas watermark dilakukan beberapa serangan yang diterapkan antara lain JPEG *compression*, *rotate*, *scaling*, *cropping*, *median filter*, *salt and pepper*, AWGN, dan *histogram equalization*, kemudian output pada watermark yang telah diberi serangan dilakukan perhitungan untuk menentukan kualitas citra watermark setelah proses ekstraksi. Parameter kualitas watermark yang digunakan pada penelitian ini adalah *robustness* dan *transparency*. *Transparency* dapat dianalisis dengan membandingkan antara citra *host* sesudah dan sebelum proses *embedding* yang dapat dianalisis dari nilai PSNR. *Robustness* dapat dianalisis dari nilai BER dan SSIM dengan membandingkan antara citra watermark sebelum proses *embedding* dan setelah proses *extraction*. Waktu komputasi dianalisis selama proses *embedding* dan *extraction*.

#### 4. PENGUJIAN SISTEM DAN HASIL ANALISIS

##### 4.1 Desain Sistem

Dalam penelitian ini terdapat dua skenario pengujian sistem yaitu pengujian pada sistem penyisipan dan pengujian pada sistem ekstraksi.

##### 4.1.1 Pengujian Sistem Penyisipan

Pengujian pada sistem penyisipan watermark bertujuan untuk mengetahui pengaruh perubahan nilai dari parameter dalam proses penyisipan terhadap nilai PSNR. Nilai yang dihasilkan dapat berubah-ubah secara acak karena penyisipan citra watermark disisipkan secara menyebar pada citra *host*.

Tabel 4.1. Pengujian sistem penyisipan

Parameter	Nilai pada Parameter			
	YCbCr		RGB	
Ruang Warna pada citra <i>host</i>				
Resolusi citra <i>host</i> (px)	256 x 256	512 x 512	1024 x 1024	2048 x 2048
Resolusi citra watermark (px)	8 x 8		16 x 16	32 x 32
<i>Alpha</i>	0,2		0,6	1,0
Rasio Kompresi CS	Tanpa CS	CS 25%	CS 50%	CS 75%

##### 4.1.2 Pengujian Sistem Ekstraksi

Pengujian sistem ekstraksi bertujuan untuk mengetahui ketahanan citra watermark terhadap serangan yang diberikan setelah proses *embedding* dengan parameter paling optimal terhadap nilai BER, SSIM, dan waktu komputasi. Nilai yang dihasilkan dapat berubah-ubah secara acak karena proses ekstraksi mengambil kembali citra watermark yang telah disisipkan secara menyebar pada proses penyisipan.

Tabel 4.2. Pengujian sistem ekstraksi

Parameter	Nilai pada Parameter			
	25%	50%	75%	100%
JPEG <i>Compression</i>	25%	50%	75%	100%
<i>Rotate</i>	90°	180°		270°
<i>Scaling</i>	50%		150%	
<i>Cropping</i>	16 px		32 px	
<i>Median Filter</i>				-
AWGN	0,001	0,01		0,1
<i>Salt and Pepper Noise</i>	0,001	0,01		0,1
<i>Histogram Equality</i>				-

## 4.2 Analisis Pengaruh Parameter pada *Image Watermarking*

### 4.2.1 Analisis Pengaruh Parameter Ruang Warna pada Citra *Host*

Ruang warna pada citra *host* merupakan bagian untuk memvisualkan citra *host* yang disisipkan citra *watermark*. Pada penelitian ini menguji antara ruang warna RGB dan dengan konversi ke dalam YCbCr pada citra *host*. Nilai PSNR, BER, SSIM, dan waktu komputasi di dapat dengan mengambil nilai dari resolusi citra *host*  $2048 \times 2048$  piksel, resolusi citra *watermark*  $16 \times 16$  piksel,  $\alpha = 0,6$ , rasio kompresi CS sebesar 75% dengan hasil sebagai berikut.

**Tabel 4.3.** Pengujian sistem penyisipan

Ruang warna	PSNR (dB)		BER	SSIM	Waktu (s)	
	Host	Watermark			Embedding	Ekstraksi
YCbCr	41,4947	Inf	0	1	1,2022	50,7662
RGB	46,6790	Inf	0	1	1,3134	51,2996

Dari tabel di atas dapat disimpulkan nilai PSNR pada ruang warna RGB lebih tinggi dibandingkan pada ruang warna YCbCr. Hal ini disebabkan perbedaan *range* dari nilai pada tiap layer dalam ruang warna. Pada RGB, *layer* R, G, dan B memiliki *range* nilai dari 0-255. Sedangkan pada YCbCr, *layer* Y memiliki *range* nilai dari 16-235, *layer* Cb dan Cr memiliki *range* nilai dari 16-240. *Range* nilai yang lebih besar juga mempengaruhi waktu komputasi pada proses ekstraksinya.

### 4.2.2 Analisis Pengaruh Parameter Resolusi *Host*

Resolusi *host* merupakan banyak piksel dalam citra *host* yang disisipkan *watermark*. Pada penelitian ini menguji antara resolusi  $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$ ,  $512 \times 512$ , dan  $1024 \times 1024$  piksel. Nilai PSNR, BER, SSIM, dan waktu komputasi di dapat dengan mengambil nilai dari ruang warna pada citra *host* RGB, resolusi citra *watermark*  $16 \times 16$  piksel,  $\alpha = 0,6$ , dan rasio kompresi CS sebesar 75% dengan hasil sebagai berikut.

**Tabel 4.4.** Analisis pengaruh parameter ruang warna pada citra *host*

Ukuran Citra Host (px)	PSNR (dB)		BER	SSIM	Waktu (s)	
	Host	Watermark			Embedding	Ekstraksi
256 x 256						
512 x 512	37,0378	3,1133	0,4883	0,0093	0,3301	2,9494
1024 x 1024	39,7493	3,7482	0,4219	0,0981	0,4696	12,6644
2048 x 2048	46,6790	Inf	0	1	1,3134	51,2996

Dari tabel di atas dapat disimpulkan semakin besar resolusi *host* yang disisipkan bit *watermark* maka semakin baik nilai PSNR, BER, dan SSIM. Hal ini disebabkan karena semakin banyak piksel dari citra *host* maka semakin besar juga cakupan yang disisipkan dalam satu bit citra *watermark* hal ini juga menyebabkan proses ekstraksi menjadi lebih lama. Pengaruh dari perubahan ukuran *host* tidak terlalu jauh berbeda dibandingkan dengan resolusi *host* yang lebih kecil.

### 4.2.3 Analisis Pengaruh Parameter *Alpha*

*Alpha* atau faktor input merupakan bilangan yang menentukan banyaknya bilangan pada matriks *host* untuk disisipkan yang merupakan matriks yang telah di transformasi DWT pada komponen *red* dalam citra *host* ruang warna RGB. Citra *watermark* memiliki isi semua koefisien *zero* kecuali diagonal utamanya sejumlah lebar sisi. Nilai *alpha* 1 berarti seluruh bilangan selain 0 disisipkan citra *watermark*. Pada penelitian ini menguji nilai *alpha* 0,2, 0,6, dan 1,0. Nilai PSNR, BER, SSIM, dan waktu komputasi di dapat dengan mengambil nilai dari ruang warna pada citra *host* RGB, resolusi citra *host*  $2048 \times 2048$  piksel, resolusi citra *watermark*  $16 \times 16$  piksel, dan rasio kompresi CS sebesar 75% dengan hasil sebagai berikut.

**Tabel 4.5.** Analisis pengaruh parameter *alpha*

Alpha	PSNR (dB)		BER	SSIM	Waktu (s)	
	Host	Watermark			Embedding	Ekstraksi
0,2	55,2675	3,1833	0,4805	0,0338	1,0467	51,1142
0,6	46,6790	Inf	0	1	1,3134	51,2996
1,0	43,2192	Inf	0	1	1,4803	52,7503

Dari tabel di atas dapat disimpulkan semakin kecil nilai *alpha* maka nilai PSNR-nya semakin baik dan waktu komputasi lebih cepat. Semakin besar nilai *alpha* maka semakin buruk nilai PSNR-nya namun memiliki

nilai BER dan SSIM yang lebih baik namun dengan waktu komputasi yang lebih lama. Pada nilai  $\alpha = 0,6$ , citra *watermark* masih dapat dipulihkan secara sempurna dengan nilai PSNR yang lebih baik dibandingkan nilai  $\alpha = 1$ .

#### 4.2.4 Analisis Pengaruh Parameter Ukuran Citra *Watermark*

Resolusi host merupakan banyak piksel dalam citra *watermark* yang disisipkan. Pada penelitian ini menguji antara resolusi  $8 \times 8$ ,  $16 \times 16$ , dan  $32 \times 32$  piksel. Nilai PSNR, BER, SSIM, dan waktu komputasi di dapat dengan mengambil nilai dari ruang warna pada citra *host* RGB, resolusi citra *host*  $2048 \times 2048$  piksel,  $\alpha = 0,6$ , dan rasio kompresi CS sebesar 75% dengan hasil sebagai berikut.

**Tabel 4.6.** Analisis pengaruh parameter ukuran citra *watermark*

Ukuran Citra Watermark (px)	PSNR (dB)		BER	SSIM	Waktu (s)	
	Host	Watermark			Embedding	Ekstraksi
<b>8 x 8</b>	46,7005	18,0618	0,0156	0,9604	1,0133	51,1790
<b>16 x 16</b>	46,6790	Inf	0	1	1,3134	51,2996
<b>32 x 32</b>	46,7992	Inf	0	1	1,5345	51,3869

Dari tabel di atas dapat disimpulkan bahwa ukuran citra *watermark* tidak banyak mempengaruhi nilai PSNR, hal ini disebabkan proses penyisipan *watermark* yang dilakukan secara berulang-ulang hingga kapasitas dari citra *host* penuh. Semakin besar citra *watermark* maka semakin baik nilai BER dan SSIM-nya namun waktu komputasi semakin lama, hal ini disebabkan karena pada proses rekonstruksi BP, semakin banyak jumlah bit sampel yang di ambil dalam proses kompresinya maka semakin baik hasil dari rekonstruksinya. Perubahan nilai bit pada piksel disisipkan tidak jauh berbeda.

#### 4.2.5 Analisis Pengaruh Parameter Rasio Kompresi

Parameter rasio kompresi merupakan besar kompresi yang dilakukan pada proses CS. Penelitian ini menguji antara rasio kompresi 25%, 50%, dan 75%. Nilai PSNR, BER, SSIM, dan waktu komputasi di dapat dengan mengambil nilai dari ruang warna pada citra *host* RGB, resolusi citra *host*  $2048 \times 2048$  piksel, resolusi citra *watermark*  $16 \times 16$  piksel, dan  $\alpha = 0,6$  dengan hasil sebagai berikut.

**Tabel 4.7.** Analisis pengaruh parameter rasio kompresi *watermark*

Kompresi Watermark	PSNR (dB)		BER	SSIM	Waktu (s)	
	Host	Watermark			Embedding	Ekstraksi
<b>Tanpa CS</b>	46,7433	7,4548	0,1797	0,5964	1,0428	52,6273
<b>CS 25%</b>	46,6822	7,3614	0,1836	0,5406	0,9971	52,9644
<b>CS 50%</b>	46,6400	16,3009	0,0234	0,9357	1,0067	52,8604
<b>CS 75%</b>	46,6790	Inf	0	1	1,0134	52,2996
<b>CS 100%</b>	46,6704	Inf	0	1	1,0476	52,4474

Dari tabel di atas dapat disimpulkan bahwa besar rasio kompresi tidak banyak mempengaruhi nilai PSNR, hal ini disebabkan karena pengaruh proses penyisipan menggunakan repetisi dengan menyisipkan citra *watermark* secara berulang-ulang hingga kapasitas dari citra *host* penuh. Nilai BER dan SSIM tanpa menggunakan CS lebih baik dibandingkan menggunakan CS dengan rasio kompresi 25%, namun CS dengan rasio kompresi lebih besar dari 25% nilai BER dan SSIM semakin baik dengan waktu komputasi yang lebih lama. Hal ini disebabkan CS dengan rasio kompresi lebih kecil mengambil sedikit sample bit yang lebih sulit untuk direkonstruksi, jika pengambilan sampel bit lebih besar dari 25% maka CS dengan algoritma rekonstruksi BP dapat merekonstruksi citra *watermark* dengan lebih baik. Rasio kompresi CS 100% menunjukkan nilai BER dan SSIM lebih baik dibandingkan tanpa menggunakan CS, hal ini disebabkan pada proses kompresi CS digunakan DCT sebagai *sparsity transform* dan digunakan *projection transform* sedangkan tanpa CS, citra *watermark* dilewatkan tanpa adanya proses *sparsity transform* dan *projection transform*.

### 4.3 Analisis Ketahanan *Watermark* Terhadap Serangan

Serangan yang digunakan pada penelitian ini adalah JPEG *compression*, *rotate*, *scaling*, *cropping*, *median filter*, *motion filter*, *salt and pepper*, AWGN, dan *histogram equation*. Parameter yang digunakan yaitu nilai  $\alpha = 1$ , ukuran citra *host*  $2048 \times 2048$  piksel, ukuran citra *watermark*  $16 \times 16$  piksel, dan rasio kompresi sebesar 75%.

#### 4.3.1 Ketahanan *Watermark* Terhadap Serangan JPEG *Compression*

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan JPEG *compression* pada ruang warna citra *host* RGB dan YCbCr.

Tabel 4.8. Ketahanan *watermark* terhadap serangan *JPEG compression*

Nilai Serangan	Ruang Warna	PSNR (dB)		BER	SSIM	Waktu Ekstraksi (s)
		Serangan	Watermark			
25%	YCbCr	34,4903	4,1701	0,3828	0,1755	52,0737
	RGB	35,1277	2,7791	0,5273	0,0215	52,0793
50%	YCbCr	36,7882	Inf	0	1	51,2729
	RGB	36,7155	3,2188	0,4766	0,0699	51,2768
75%	YCbCr	38,8508	Inf	0	1	51,6121
	RGB	37,9208	3,5134	0,4453	0,1271	52,1511
100%	YCbCr	49,5196	Inf	0	1	51,2292
	RGB	44,5685	Inf	0	1	52,8008

Dari tabel di atas dapat disimpulkan bahwa ruang warna YCbCr lebih tahan terhadap serangan dibandingkan ruang warna RGB, hal ini disebabkan karena citra *watermark* disisipkan ke dalam komponen Y pada ruang warna YCbCr merupakan komponen *luminance* yang lebih tahan terhadap serangan dibandingkan komponen *red* pada RGB. Semakin tinggi nilai variansi serangan pada *JPEG compression* maka semakin tinggi tingkat kemiripan antara citra ter-*watermark* tanpa serangan dan dengan serangan maka semakin baik juga nilai BER dan SSIM-nya namun dalam waktu komputasi, ruang warna YCbCr lebih cepat dibandingkan ruang warna RGB.

#### 4.3.2 Ketahanan *Watermark* Terhadap Serangan *Rotate*

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan *rotate*.

Tabel 4.9. Ketahanan *watermark* terhadap serangan *rotate*

Nilai Serangan	Ruang warna	PSNR (dB)		BER	SSIM	Waktu Ekstraksi (s)
		Serangan	Watermark			
90°	YCbCr	8,9545	3,0787	0,4922	0,0362	51,1386
	RGB	8,9738	3,0444	0,4961	0,0323	51,2371
180°	YCbCr	8,3185	3,0787	0,4922	0,0145	51,6776
	RGB	8,3358	3,1833	0,4935	0,0113	51,8641
270°	YCbCr	8,9545	3,2906	0,4688	0,0717	51,6841
	RGB	8,738	3,1133	0,4883	0,0092	51,8778

Dari tabel di atas dapat disimpulkan bahwa citra *watermark* tidak tahan terhadap serangan rotasi hal ini dikarenakan tidak adanya *sync pattern* dalam proses penyisipan dan ekstraksi yang berguna untuk mendeteksi serangan rotasi. Namun dari data di atas, ruang warna YCbCr memiliki nilai BER, SSIM, dan waktu komputasi yang lebih baik dibandingkan ruang warna pada RGB. Hal ini disebabkan karena komponen Y pada ruang warna YCbCr merupakan komponen *luminance* yang lebih tahan terhadap serangan dibandingkan komponen *red* pada RGB.

#### 4.3.3 Ketahanan *Watermark* Terhadap Serangan *Scaling*

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan *scaling*.

Tabel 4.10. Ketahanan *watermark* terhadap serangan *scaling*

Nilai Serangan	Ruang warna	PSNR (dB)		BER	SSIM	Waktu Ekstraksi (s)
		Serangan	Watermark			
50%	YCbCr	38,1421	Inf	0	1	51,5836
	RGB	38,2618	Inf	0	1	51,8105
150%	YCbCr	Inf	Inf	0	1	53,5344
	RGB	Inf	Inf	0	1	54,6623

Dari tabel di atas dapat disimpulkan bahwa citra *watermark* tahan terhadap serangan *scaling* lebih besar maupun lebih kecil. Waktu komputasi pada ruang warna YCbCr lebih cepat dibandingkan ruang warna RGB.

#### 4.3.4 Ketahanan *Watermark* Terhadap Serangan *Cropping*

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan *cropping*.



**Tabel 4.11.** Ketahanan *watermark* terhadap serangan *cropping*

Nilai Serangan	Ruang warna	PSNR Watermark (dB)	BER	SSIM	Waktu (s)
16 px	YCbCr	3,0444	0,4961	0,0093	52,0452
	RGB	2,9097	0,5117	0,0343	52,5417
32 px	YCbCr	3,1482	0,4844	0,0431	51,8127
	RGB	3,2906	0,4688	0,0731	54,6688

Dari tabel di atas dapat disimpulkan bahwa *watermark* tidak tahan terhadap serangan *cropping*. Namun *cropping* lebih besar menghasilkan nilai BER yg lebih baik. Hal ini karena proses penyisipan *watermark* menggunakan repetisi dengan mengambil bit paling sering muncul yang mengakibatkan nilai BER dan SSIM tidak menentu jika *watermark* diberikan serangan *cropping* dengan nilai variansi berbeda.

#### 4.3.5 Ketahanan *Watermark* Terhadap Serangan *Median Filter*

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan *median filter*.

**Tabel 4.12.** Ketahanan *watermark* terhadap serangan *median filter*

Ruang warna	PSNR (dB)		BER	SSIM	Waktu Ekstraksi (s)
	Serangan	Watermark			
YCbCr	36,5796	Inf	0	1	41,9215
RGB	40,1599	Inf	0	1	44,4767

Dari tabel di atas dapat disimpulkan bahwa *watermark* tahan terhadap serangan *median filter* meskipun ruang warna YCbCr memiliki nilai PSNR yang lebih buruk dibandingkan ruang warna RGB. Waktu komputasi pada ruang warna YCbCr lebih cepat dibandingkan RGB.

#### 4.3.6 Ketahanan *Watermark* Terhadap Serangan *Salt and Pepper*

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan *salt and pepper*.

**Tabel 4.13.** Ketahanan *watermark* terhadap serangan *salt and pepper*

Nilai Serangan	Ruang warna	PSNR (dB)		BER	SSIM	Waktu Ekstraksi (s)
		Serangan	Watermark			
0,001	YCbCr	35,0200	Inf	0	1	41,2283
	RGB	34,9332	Inf	0	1	41,2974
0,01	YCbCr	24,9609	Inf	0	1	41,4830
	RGB	24,9708	Inf	0	1	41,5506
0,1	YCbCr	14,9591	3,7886	0,4180	0,1788	42,7371
	RGB	14,9805	3,6292	0,4336	0,1260	50,1435

Dari tabel di atas dapat disimpulkan bahwa *watermark* tahan terhadap serangan dengan variansi 0,001 dan 0,01. Pada nilai variansi serangan 0,1 ruang warna YCbCr lebih baik dibandingkan dengan RGB. Waktu komputasi pada YCbCr lebih cepat dibandingkan RGB. Nilai PSNR yang lebih baik mempengaruhi hasil dari BER dan SSIM yang lebih baik.

#### 4.3.7 Ketahanan *Watermark* Terhadap Serangan AWGN

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan AWGN.

**Tabel 4.14.** Ketahanan *watermark* terhadap serangan AWGN

Nilai Serangan	Ruang warna	PSNR (dB)		BER	SSIM	Waktu Ekstraksi (s)
		Serangan	Watermark			
0,001	YCbCr	30,0385	Inf	0	1	42,1248
	RGB	30,0350	Inf	0	1	42,7311
0,01	YCbCr	20,3122	Inf	0	1	45,5833
	RGB	20,3083	3,2545	0,4727	0,0502	45,6527
0,1	YCbCr	11,6698	3,1833	0,4805	0,0342	46,3679
	RGB	11,6703	3,2188	0,4766	0,0167	47,1007

Dari tabel di atas dapat disimpulkan bahwa *watermark* tahan terhadap serangan dengan variansi 0,001 dan 0,01. Pada nilai variansi serangan 0,1 ruang warna RGB lebih baik dibandingkan dengan YCbCr. Waktu komputasi pada YCbCr lebih cepat dibandingkan RGB. Nilai PSNR yang lebih baik mempengaruhi hasil dari BER dan SSIM yang lebih baik.

#### 4.3.8 Ketahanan *Watermark* Terhadap Serangan *Histogram Equality*

Berikut merupakan sampel nilai yang didapatkan dari hasil serangan *Histogram Equality*.

**Tabel 4.15.** Ketahanan *watermark* terhadap serangan *histogram equality*

Ruang warna	PSNR (dB)		BER	SSIM	Waktu Ekstraksi (s)
	Serangan	Watermark			
YCbCr	23,9258	Inf	0	1	44,9448
RGB	23,8612	Inf	0	1	45,5235

Dari tabel di atas dapat disimpulkan bahwa *watermark* tahan terhadap serangan *histogram equation* meskipun pada ruang warna RGB nilai PSNR-nya lebih buruk dibandingkan ruang warna YCbCr. Waktu komputasi pada ruang warna YCbCr lebih cepat dibandingkan ruang warna RGB.

#### 4.6 Analisis Pengaruh *Image Watermark* Terhadap Permasalahan Hak Cipta

Citra *watermark* pada penelitian ini lebih tahan terhadap serangan seperti *JPEG Compression 50%, 75%, dan 100%, scaling 50% dan 150%, median filter, salt and pepper* dengan nilai variansi 0,001 dan 0,01, AWGN dengan nilai variansi 0,001 dan 0,01, dan *histogram equation* pada citra *host* berukuran  $2048 \times 2048$  piksel disisipkan citra *watermark* berukuran  $16 \times 16$  piksel dengan rasio kompresi CS sebesar 75%, dan nilai  $\alpha = 1$  dalam ruang warna YCbCr.

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Dari hasil pengujian pada penelitian ini di dapat beberapa kesimpulan sebagai berikut.

1. Citra *watermark* dapat melindungi hak cipta jika tanpa diberikan serangan pada citra ter-*watermark* dan diberikan serangan *JPEG Compression 50%, 75%, dan 100%, scaling 50% dan 150%, median filter, salt and pepper* dengan nilai variansi 0,001 dan 0,01, AWGN dengan nilai variansi 0,001 dan 0,01, dan *histogram equation*.
2. Semakin besar rasio kompresi pada CS maka semakin baik juga nilai BER dan SSIM dan waktu komputasinya namun dengan waktu yang lebih lama. *Watermark* tanpa menggunakan CS memiliki nilai BER yang lebih buruk dibandingkan menggunakan CS.
3. Citra ter-*watermark* dalam ruang warna RGB memiliki nilai PSNR yang lebih baik yaitu sebesar 46,6238 dB dibandingkan mengkonversikan terlebih dahulu ke dalam ruang warna YCbCr yang memiliki nilai PSNR 41,4328 dB. Namun dengan mengkonversikan ke dalam ruang warna YCbCr, *watermark* lebih tahan terhadap serangan karena *watermark* disisipkan dalam komponen Y yang merupakan luminance yang mampu meningkatkan *embedding capacity*.
4. Ukuran citra *host* berbanding lurus dengan nilai PSNR, BER dan SSIM namun dengan waktu komputasi yang lebih lambat.
5. Ukuran citra *watermark* dan nilai  $\alpha$ , berbanding lurus dengan nilai BER dan SSIM yang baik namun berbanding terbalik dengan nilai PSNR dan waktu komputasinya.

### 5.2 Saran

Pengujian penelitian ini masih terdapat kekurangan, untuk itu perlu dilakukan pengujian dengan metode lain.

1. Menggunakan variasi ruang warna selain RGB dan YCbCr, seperti *grayscale, biner, HSV, dan NTSC*.
2. Menambahkan teknik penyisipan seperti QIM dengan SS dilakukan setelah proses CS.
3. Menggunakan transformasi *pre-processing* pada citra *host* selain DWT, seperti DCT, SWT, LWT, dan lain-lain.
4. Menggunakan transformasi *pre-processing* pada citra *watermark* selain DCT, seperti DWT.
5. Menggunakan jenis *host file* selain citra, seperti audio dan video.
6. Menggunakan jenis *watermark file* selain citra, seperti audio dan video.

## 6. REFERENSI

- [1] A. Samčović and M. Milovanović, "Robust Digital Image Watermarking Based on Wavelet Transform And Spread Spectrum Techniques," in *Telecommunications Forum Telfor (TELFOR), 2015 23rd*, Belgrade, Serbia, 2015.
- [2] A. Hardyatmaka, "Otentikasi Citra Digital dengan Watermarking Menggunakan Spread Spectrum, Discrete Wavelet Transform, dan Principal Component Analysis," *Informatika Teori dan Pemrograman*, 2013.
- [3] Y. Zhang, L. Y. Zhang and J. Zhou, "A Review of Compressive Sensing in Information Security Field," *Special Section On Green Communication And Networking For 5G Wireless*, vol. 4, pp. 2507-2519, 2016.
- [4] I. Safitri, N. Ibrahim and H. Yogaswara, "Compressive Sensing Audio Watermarking dengan Metode LWT dan QIM," *Elkomika*, vol. 6, pp. 405-419, 2018.
- [5] I. Wahidah and A. B. Suksmono, "Reconstruction Algorithms For Compressive Video Sensing Using Basis Pursuit," *Measurement*, vol. 1, 2008.
- [6] A. Pangestu, "Analisis Image Watermarking Menggunakan Compressive Sensing Algoritma Orthogonal Matching Pursuit Dengan Pendekatan Berbasis Discrete Cosine Transform Menggunakan Singular Value Decomposition," Bandung, 2017.
- [7] A. E. F. Pratono, "Implementasi Image Watermarking Dengan Metode Spread Spectrum dan DWT Berbasis Android," Telkom University, Bandung, 2015.
- [8] D. D. Setyanto, "Analisis Transformasi Wavelet Diskrit Untuk Penginderaan Video Grayscale Kompresif," Institut Teknologi Telkom, Bandung, 2011.
- [9] A. Haryadi and Y. Suyanto, "Perbandingan PSNR, Bitrate, dan MOS pada Pengkodean H.264 Menggunakan Metode Prediksi Temporal," *IJEIS*, vol. 2, p. 155, October 2012.
- [10] R. Baraniuk, J. Romberg and M. Wakin. [Online]. Available: <https://pdfs.semanticscholar.org/presentation/8fdf/5503707dc7170f4bedfb5b40fcd07af0785b.pdf>.
- [11] Jain and A. K., "Fundamental of Digital Image Processing," *Englewood Cliffs*, pp. 150-153, 1989.
- [12] Y. J. Fan and C. Kamath, "A Comparison of Compressed Sensing and Sparse Recovery Algorithms," *Statistics, Optimization And Information Computing*, vol. 4, pp. 194-213, 2016.
- [13] G. Hua, Y. Xiang and G. Bi, "When Compressive Sensing Meets Data Hiding," *IEEE Signal Processing Letters*, vol. 23, pp. 473-477, April 2016.
- [14] A. Draganic, I. Orovic and S. Stankovic, "On Some Common Compressive Sensing Recovery Algorithms And Applications," 2017.
- [15] A. Bose and S. P. Maity, "Improved Spread Spectrum Compressive Image Watermark Detection with Distortion Minimization," in *2016 International Conference on Signal Processing and Communications (SPCOM)*, Shibpur, Howrah, 711103, India, 2016.
- [16] A. Bose and S. P. Maity, "Spread Spectrum Watermark Detection on Degraded Compressed Sensing," vol. 1, 2017.
- [17] A. M. Rohman, "Analisis Transformasi Proyeksi Gaussian Untuk Penginderaan Citra Kompresif," Institut Teknologi Telkom, Bandung, 2011.
- [18] D. M. A. Shakir, "Compressive Sensing Based Signal Recovery With Different Transforms," *Pure and Eng. Sciences*, vol. 20, pp. 129-141, 2017.