

Compressive Sensing Audio Watermarking **dengan Metode LWT dan QIM**

IRMA SAFITRI, NUR IBRAHIM, HERLAMBANG YOGASWARA

Universitas Telkom
Email: irmasaf@telkomuniversity.ac.id

Received 9 Juli 2018 | *Revised* 13 Agustus 2018 | *Accepted* 30 September 2018

ABSTRAK

*Penelitian ini mengembangkan teknik Compressive Sensing (CS) untuk audio watermarking dengan metode Lifting Wavelet Transform (LWT) dan Quantization Index Modulation (QIM). LWT adalah salah satu teknik mendekomposisi sinyal menjadi 2 sub-band, yaitu sub-band low dan high. QIM adalah suatu metode yang efisien secara komputasi atau perhitungan watermarking dengan menggunakan informasi tambahan. Audio watermarking dilakukan menggunakan file audio dengan format *.wav berdurasi 10 detik dan menggunakan 4 genre musik, yaitu pop, classic, rock, dan metal. Watermark yang disisipkan berupa citra hitam putih dengan format *.bmp yang masing-masing berukuran 32x32 dan 64x64 pixel. Pengujian dilakukan dengan mengukur nilai SNR, ODG, BER, dan PSNR. Audio yang telah disisipkan watermark, diuji ketahanannya dengan diberikan 7 macam serangan berupa LPF, BPF, HPF, MP3 compression, noise, dan echo. Penelitian ini memiliki hasil optimal dengan nilai SNR 85,32 dB, ODG $-8,34 \times 10^{-11}$, BER 0, dan PSNR ∞ .*

Kata kunci: *Audio watermarking, QIM, LWT, Compressive Sensing.*

ABSTRACT

*This research developed Compressive Sensing (CS) technique for audio watermarking using Wavelet Transform (LWT) and Quantization Index Modulation (QIM) methods. LWT is one technique to decompose the signal into 2 sub-bands, namely sub-band low and high. QIM is a computationally efficient method or watermarking calculation using additional information. Audio watermarking was done using audio files with *.wav format duration of 10 seconds and used 4 genres of music, namely pop, classic, rock, and metal. Watermark was inserted in the form of black and white image with *.bmp format each measuring 32x32 and 64x64 pixels. The test was done by measuring the value of SNR, ODG, BER, and PSNR. Audio that had been inserted watermark was tested its durability with given 7 kinds of attacks such as LPF, BPF, HPF, MP3 Compression, Noise, and Echo. This research had optimal result with SNR value of 85.32 dB, ODG value of -8.34×10^{-11} , BER value of 0, and PSNR value of ∞ .*

Keywords: *Audio watermarking, QIM, LWT, Compressive Sensing.*

1. PENDAHULUAN

Informasi dapat berupa *file* multimedia berbentuk gambar, suara, maupun video (Lalitha et al, 2016). Internet menjadikan penyebaran informasi menjadi sangat luas dan sangat mudah diakses di mana saja dan kapan saja. Kemudahan ini menyebabkan keamanan informasi tersebut menjadi sangat rentan terhadap terjadinya penyalahgunaan. Penyalahgunaan tersebut antara lain seperti penyebaran musik, lagu dan *file audio* lainnya tanpa seizin pemilik informasi.

Pelanggaran terhadap penyebaran *file audio* saat ini telah banyak terjadi, di antaranya adalah pelanggaran hak cipta dan penjualan ilegal. Hal ini dapat merugikan beberapa pihak yang mempunyai hak terhadap *file audio* tersebut. *Watermarking* menjadi salah satu solusi untuk memecahkan masalah tersebut karena dapat memberikan hak cipta yang dapat melekat pada *file audio* agar pelanggaran-pelanggaran tersebut tidak terjadi lagi. *Watermarking* adalah suatu proses untuk menyisipkan sinyal *digital* yang dimasukkan ke dalam sinyal *digital*, video, *audio* atau perangkat lunak (Jasmin & Prabha, 2014). *Watermarking* yang dihasilkan tidak selalu sempurna seperti yang diharapkan, karena penyisipan suatu informasi ke dalam suatu *file* akan menurunkan kualitas dari *file* tersebut. Penggunaan metode *watermarking* yang tepat dapat mengantisipasi penurunan kualitas tersebut.

Penelitian mengenai *audio watermarking* telah dilakukan untuk beberapa metode seperti *Discrete Cosine Transform* (DCT) (Budiman et al, 2016), *Discrete Fourier Transform* (DFT) (Dhar et al, 2010), *Discrete Wavelet Transform* (DWT) (Safitri et al, 2017), *Lifting Wavelet Transform* (LWT) (Lalitha et al, 2016) dan *Singular Value Decomposition* (SVD) (Lei et al, 2012). Sedangkan *Compressive Sensing* (CS) adalah teknik *sampling* baru yang bertentangan dengan kebijakan umum dalam akuisisi data. Metode tradisional yang digunakan untuk *sampling* berdasarkan teori Shannon. Dengan penggunaan teknik CS, sinyal dan gambar tertentu dapat dikembalikan dengan hanya menggunakan *sample* yang jauh lebih sedikit daripada yang digunakan oleh metode tradisional (Candes & Wakin, 2008). Untuk memungkinkan hal ini, CS bergantung pada dua prinsip yaitu *sparsity*, yang berkaitan dengan sinyal, dan inkohherensi, yang berhubungan dengan modalitas penginderaan (Candes & Wakin, 2008).

Dalam pengetahuan kami, CS pada *audio watermarking* masih belum dieksplor. Oleh karena itu, penelitian ini mengembangkan teknik CS pada *audio watermarking*. Tujuan dari penelitian ini antara lain adalah untuk membuat sistem kombinasi *Compressive Sensing* dan *Audio Watermarking* agar diperoleh efisiensi kompresi pada data *watermark*, serta menganalisis kualitas dan kinerja sistem *Audio Watermarking* dari segi ketahanan (*robustness*). Metode yang digunakan adalah *Lifting Wavelet Transform* (LWT), *QR decomposition* dan CS yang diterapkan pada *watermark*. LWT adalah suatu teknik untuk mendekomposisi suatu sinyal ke dalam 2 *sub-band*, yaitu *sub-band low* dan *high* (Jasmin & Prabha, 2014). CS adalah suatu metode kompresi dimana sampel dengan jumlah yang lebih kecil akan diperoleh dari seluruh panjang sinyal *sparse* dengan mengambil pengukuran *non-adaptive* linier yang mempertahankan struktur dari sinyal (Bash & Kayhan, 2015). *Watermark* yang telah di-*sparse* direkonstruksi untuk mengembalikan ke bentuk semula. QIM pada umumnya menggunakan 2 quantizer Q_1 dan Q_2 untuk mengkuantisasi sinyal *host* menjadi 2 *set disjoint points* dimana 1 *set* merepresentasikan bit "0" dan *set* lainnya merepresentasikan bit "1" (Li & Cox, 2007).

1.1 Compressive Sensing (CS)

Compressive Sensing (CS) adalah suatu metode kompresi dimana sejumlah sampel yang lebih kecil yang diperoleh dari seluruh panjang sinyal *sparse* dengan mengambil pengukuran *non-adaptive* linier yang mempertahankan struktur dari sinyal, dan menggunakan metode optimasi numerik (Neetha & Koya, 2015). CS dapat merekonstruksi sinyal dengan menggunakan sejumlah pengukuran acak dan sinyal yang direkonstruksi harus berbentuk sinyal *sparse*. Sinyal yang *sparse* mengandung beberapa koefisien sebagai *non-zero* dan koefisien lainnya adalah *zero* (Bash & Kayhan, 2015).

Sinyal $x \in R^N$ adalah *k-sparse* ketika elemen k dari x adalah *non-zero*. Jika $f \in R^N$ adalah sinyal *k-sparse* di ruang ψ yang merupakan kombinasi linier dari N , ψ merepresentasikan basis *orthonormal* dan f direpresentasikan sebagai (Bash & Kayhan, 2015):

$$f = \psi x \quad (1)$$

Kemudian sinyal x dapat direpresentasikan sebagai sinyal yang tersebar $x = \psi' y$ dan salah satu contohnya adalah (Bash & Kayhan, 2015):

$$y = \phi f \quad (2)$$

dimana $y \in R^M$ adalah perhitungan vektor dan ϕ adalah $M \times N$ matriks *sensing*. Kemudian persamaan sebelumnya dapat diubah menjadi (Bash & Kayhan, 2015):

$$y = \phi f = \phi \psi x = \phi x \quad (3)$$

Ketika ϕ adalah matriks *random Gaussian*, maka perhitungan $M = O(k) \log\left(\frac{N}{k}\right)$ merepresentasikan baris dari matriks *sensing*.

1.1.1 Algoritma Rekonstruksi

L1-Magic adalah salah satu metode rekonstruksi CS. Algoritma ini berasal dari kelas algoritma *interior point*, yaitu metode *primal-dual interior point* untuk pemrograman linier. Rumus standar dari program linier adalah (Vujovic & Dakovic, 2014):

$$\min_x \{c_0, X\} \text{ subjek untuk } y = AX; f_i(X) \leq 0 \quad (4)$$

dimana $X \in R^N, y \in R^M, A$ adalah matriks $M \times N$, pada setiap $f_i, i = 1, \dots, m$ adalah fungsi linier (Vujovic & Dakovic, 2014):

$$f_i(X) = \{c_i, X\} + d_i \quad (5)$$

untuk beberapa $c_i \in R^N, d_i \in R$, ketika \tilde{X} titik optimal, akan terdapat vektor ganda $v^* \in R^M$, dan $\lambda^* \in R^m$ yang memenuhi kondisi Karush-Kuhn-Tucker (Vujovic & Dakovic, 2014):

$$\begin{aligned} c_i + A^T v^* + \sum_i \lambda_i^* c_i &= 0 \\ \lambda_i^* f_i(\tilde{X}) &= 0, i = 1, \dots, m \\ A \tilde{X} &= y \\ f_i(\tilde{X}) &\leq 0, i = 1, \dots, m \end{aligned} \quad (6)$$

Algoritma ini dapat menemukan vektor \tilde{X} yang dicari (vektor ganda optimal) dengan memecahkan sistem dari persamaan *non*-linier.

1.2 Lifting Wavelet Transform (LWT)

LWT merupakan suatu metode untuk mendekomposisi sinyal ke dalam 2 *sub-band*, yaitu *low* dan *high*. Perubahan nilai *sampling* dapat diantisipasi dengan menyisipkan *watermark* pada *level* yang berbeda. Rekonstruksi *lifting wavelet* lebih baik karena meningkatkan kehalusan dan menurunkan efek *aliasing* dibandingkan dengan *general wavelet*. LWT dapat mengurangi hilangnya informasi, serta meningkatkan ketahanan dan keutuhan *watermark* (Jasmin & Prabha, 2014).

Matriks *polyphase* adalah matriks *Laurent Polynomials* dan dituntut determinannya harus sama dengan 1, dan diketahui pasangan filter (h, g) adalah pelengkap. Teorema *lifting* sekarang dinyatakan bahwa filter terbatas lainnya g^{new} melengkapi h dalam bentuk (Valens, 1999):

$$g^{new}(z) = g(z) + h(z)s(z^2) \quad (7)$$

Dimana, $s(z^2)$ adalah *Laurent polynomials*. Hal ini dapat dengan mudah dilihat dari g^{new} pada bentuk *polyphase* dan menghimpun matriks *polyphase* yang baru sebagai (Valens, 1999):

$$P^{new}(z) = \begin{bmatrix} h_e(z) & h_e(z) + g_e(z) \\ h_o(z) & h_o(z) + g_o(z) \end{bmatrix} = P(z) = \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \quad (8)$$

Seperti dapat dengan mudah dibuktikan bahwa determinan dari matriks *polyphase* yang baru juga sama dengan 1, seperti yang dibuktikan oleh Persamaan (6)

1.3 Quantization Index Modulation (QIM)

QIM adalah metode yang efisien secara komputasi untuk *watermarking* (Li & Cox, 2007). *Quantizer* adalah suatu fungsi yang memetakan nilai dengan yang terdekat dengan kelas *pre-defined discontinuous points*. Fungsi *round* (.) menunjukkan pembulatan nilai ke integer terdekat dan operasi kuantisasi standar dengan *step size* Δ yang didefinisikan sebagai berikut (Li & Cox, 2007):

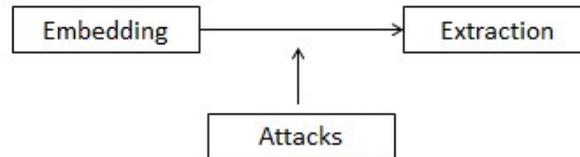
$$P^{new}(z) = \begin{bmatrix} h_e(z) & h_e(z) + g_e(z) \\ h_o(z) & h_o(z) + g_o(z) \end{bmatrix} = P(z) = \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \quad (9)$$

QIM menyematkan pesan dengan modulasi pertama dari indeks atau urutan indeks dengan pesan yang akan di-*embedding* kemudian kuantisasi sinyal *host* dengan *quantizer* yang cocok atau urutan *quantizer*. Misalkan Δ adalah kuantisasi *step size* dan L merupakan panjang dari sinyal *host* x dan pesan m yang di-*embedding* 1 bps. QIM pada umumnya menggunakan 2 *quantizer* Q_1 dan Q_2 untuk mengkuantisasi sinyal *host* menjadi 2 *set disjoint points* dimana 1 *set* merepresentasikan bit "0" dimana *set* lainnya merepresentasikan bit "1". Ketika pesannya di-*embedding*, Q_1 atau Q_2 dipilih berdasarkan bit pesan untuk kuantisasi x ke nilai kuantisasi terdekat. Sebagai contoh, Q_1 dan Q_2 dapat dipilih ketika Q_1 dikuantisasi ke integer genap dan Q_2 dikuantisasi ke integer ganjil. Jika bit "0" maka yang dipilih adalah Q_1 , jika bukan maka dipilih Q_2 . Persamaan sinyal *watermark* y (Li & Cox, 2007):

$$y = Q_m(x, |\Delta|) \quad (10)$$

2. METODOLOGI PENELITIAN

Sistem *Audio Watermarking* terdiri dari proses *embedding* dan *extraction*. Di antara kedua proses tersebut dilakukan beberapa serangan/gangguan terhadap sinyal *host* yang telah disisipkan data *watermark* tersebut. Proses *audio watermarking* secara umum dapat dilihat pada Gambar 1 berikut:

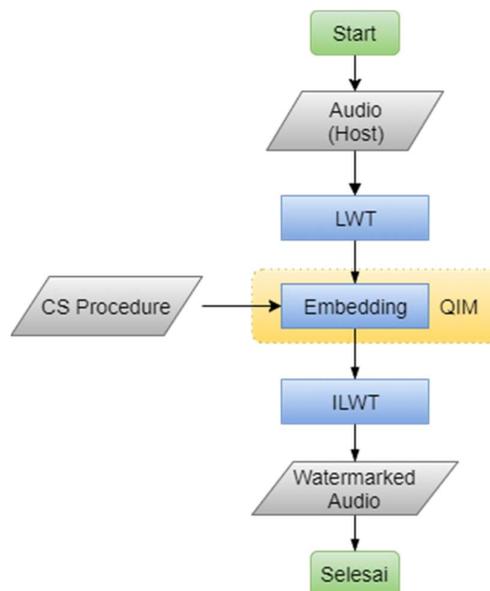


Gambar 1. Blok Diagram Umum Sistem *Audio Watermarking*

Pada penelitian ini, sistem *Compressive Sensing* (CS) dikombinasikan dengan sistem *Audio Watermarking* pada bagian *embedding* dan *extraction*. CS pada *embedding* diterapkan pada data *watermark* yang berupa sinyal *sparse* hasil kompresi data *watermark* tersebut. Sedangkan pada proses *extraction*, CS diterapkan untuk merekonstruksi data *watermark* hasil ekstraksi. Metode rekonstruksi yang digunakan adalah *L1-Magic*. Penjelasan detail teknik CS *audio watermarking* tercantum pada subbab 2.1 dan 2.2 berikut ini.

2.1 *Embedding*

Proses *embedding* merupakan proses penyisipan *watermark* ke dalam *host audio*. Dalam proses ini, hal yang perlu diperhatikan adalah nilai *intensity* dan *frame* yang digunakan. Nilai *intensity* mempengaruhi penyebaran bit *watermark* yang akan di sisipkan ke dalam *host*. *Frame* mempengaruhi nilai maksimum kapasitas dari *watermark*.



Gambar 2. Diagram Blok Proses *Embedding*

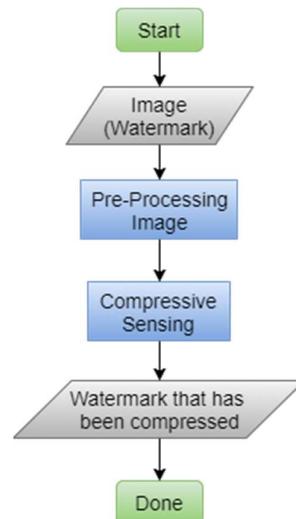
Sistem *compressive sensing audio watermarking* kami terdiri dari proses *embedding*. Diagram blok proses tersebut dapat dilihat pada Gambar 2.

Detail penjelasan Gambar 2 tersebut adalah sebagai berikut:

1. Membaca *file audio* yang dijadikan sebagai *host*.
2. *File audio* dipotong menjadi 10 detik, untuk memperkecil ukuran dan durasi.
3. *Host audio* dibagi ke dalam *frame* yang telah ditentukan. Setelah itu dilakukan proses LWT untuk dekomposisi level menjadi 2 *sub-band*, yaitu *low* dan *high*. *Sub-band low* dipilih untuk dilakukan penyisipan.
4. Sebelum dilakukan proses *embedding*, terlebih dahulu dilakukan prosedur CS.
5. Dilakukan proses *embedding* dengan menggunakan QIM, dengan aturan sebagai berikut (Li & Cox, 2007):

$$\begin{cases} S' = \text{round} \left(\frac{s}{Q} \right) * Q ; \text{jika } w(k) = 1 \\ S' = \text{floor} \left(\frac{s}{Q} \right) * Q + \left(\frac{Q}{2} \right) ; \text{jika } w(k) = 0 \end{cases} \quad (11)$$

6. ILWT mengembalikan *level sub-band*.
7. Diperoleh *audio* yang telah di *watermark*.



Gambar 3. Prosedur *Compressive Sensing* (CS)

Proses CS pada *watermark image* dapat dilihat pada Gambar 3. Penjelasan detail mengenai diagram blok pada Gambar 3 adalah sebagai berikut:

1. Membaca *file* citra yang dijadikan sebagai *watermark*.
2. Dilakukan *pre-processing* dengan mengubah ukuran citra menjadi $\left(\frac{n \times n}{8}, 8\right)$, setelah itu diubah menjadi bentuk matriks desimal berukuran $\left(\frac{n \times n}{8}, 1\right)$.
3. Setelah diperoleh bentuk matriks desimal, maka dilakukan proses mengubah bentuk matriks desimal menjadi matriks *sparse* berukuran $\left(\frac{n \times n}{8}, 256\right)$.
4. Citra *watermark* dilakukan proses *compressive sensing* yang mengubah matriks *sparse* ke dalam bentuk yang lebih kecil. Dengan rumus :

$$y = \phi x \quad (12)$$

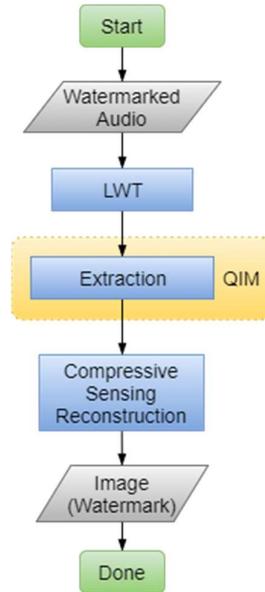
x adalah matriks *input* atau citra *watermark*, sedangkan ϕ adalah matriks *random* dengan ukuran $(\text{rasio} \times 256, 256)$.

5. Diperoleh matriks citra yang sudah di kompresi berukuran $(\frac{n \times n}{8}, rasio \times 256)$ yang selanjutnya akan disisipkan ke dalam *host*.

2.2 Extraction

Proses *extraction* bertujuan untuk memisahkan *host audio* dengan *watermark* yang telah disisipkan. Pada proses ini *watermark* akan direkonstruksi untuk mengembalikan matriks *watermark* menjadi seperti semula sebelum disisipkan ke dalam *host*.

Proses *extraction* sistem *compressive sensing audio watermarking* kami dapat dilihat pada Gambar 4.



Gambar 4. Blok Diagram Umum Sistem

Detail penjelasan Gambar 4 tersebut adalah sebagai berikut:

1. Membaca *file audio* yang telah di-*watermark*.
2. LWT membagi level ke dalam 2 *sub-band*, *low* dan *high*. Proses *extraction* dilakukan pada *sub-band low*, karena penyisipan *watermark* ada di *sub-band low*.
3. *Extraction* dilakukan untuk memisahkan *host* dengan citra *watermark* dengan proses QIM dengan menggunakan rumus *dequantization* (Agradiya et al, 2017):

$$\begin{cases} w(k) = 0; & \text{jika } \left(\frac{Q}{4}\right) \leq \text{mod}(S'', Q) < (3 * \frac{Q}{4}) \\ w(k) = 1; & \text{lainnya} \end{cases} \quad (13)$$

4. Citra *watermark* dilakukan proses rekonstruksi untuk mengembalikan bentuk matriks citra *watermark* yang sebelum di-*embedding* telah melalui proses CS menjadi seperti semula. Rekonstruksi citra dilakukan dengan algoritma rekonstruksi *L1-Magic* :

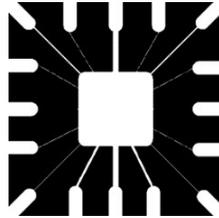
$$\min_x \{c_0, X\} \text{ subjek untuk } y = AX; \quad (14)$$

$$f_i(X) \leq 0$$

5. Diperoleh *watermark* yang asli.

3. HASIL DAN DISKUSI

Penelitian ini menggunakan *audio* dengan format *.wav dengan durasi selama 10 detik, dan 4 jenis aliran musik, yaitu *Pop*, *Classic*, *Rock*, dan *Metal*. *Watermark* yang digunakan berupa citra hitam putih dengan format *.bmp yang masing-masing berukuran 32x32 dan 64x64 *pixel*. Frekuensi *sampling* yang digunakan sebesar 44100 Hz, nilai *frame* yang digunakan sebesar 128, nilai *intensity* sebesar 0,001, dan penyisipan dilakukan pada *sub-band low*. Gambar 5 memperlihatkan *watermark* yang digunakan dalam penelitian ini.



Gambar 5. *Watermark*

Tabel 1. Pengaruh *Measurement Rate*

<i>Host</i>	<i>Watermark</i>	MR	RK	SNR	ODG	BER	PSNR
<i>Pop</i>	32x32	0,029	87,5%	78,68	-4,28E-10	0	∞
		0,025	75%	79,1	-1,98E-10	0	∞
		0.02	62,5%	80,02	-8,34E-11	0	∞
	64x64	0,029	87,5%	72,49	-0,036	0	∞
		0,025	75%	73,2	-0,036	0	∞
		0.02	62,5%	73,82	-0,037	0	∞
<i>Classic</i>	32x32	0,029	87,5%	65,96	-2,61E-08	0	∞
		0,025	75%	66,52	-1,38E-08	0	∞
		0.02	62,5%	67,59	-5,82E-09	0	∞
	64x64	0,029	87,5%	59,8	-0,26	0	∞
		0,025	75%	60,62	-0,25	0	∞
		0.02	62,5%	61,46	-0,23	0	∞
<i>Rock</i>	32x32	0,029	87,5%	80,86	-2,52E-07	0	∞
		0,025	75%	81,79	-1,35E-07	0	∞
		0.02	62,5%	82,22	-7,23E-08	0	∞
	64x64	0,029	87,5%	75,44	-0,016	0	∞
		0,025	75%	76,19	-0,017	0	∞
		0.02	62,5%	76,94	-0,017	0	∞
<i>Metal</i>	32x32	0,029	87,5%	83,89	-0,009	0	∞
		0,025	75%	84,25	-0,009	0	∞
		0.02	62,5%	85,32	-0,009	0	∞
	64x64	0,029	87,5%	78,05	-0,011	0	∞
		0,025	75%	78,69	-0,011	0	∞
		0.02	62,5%	79,74	-0,01	0	∞

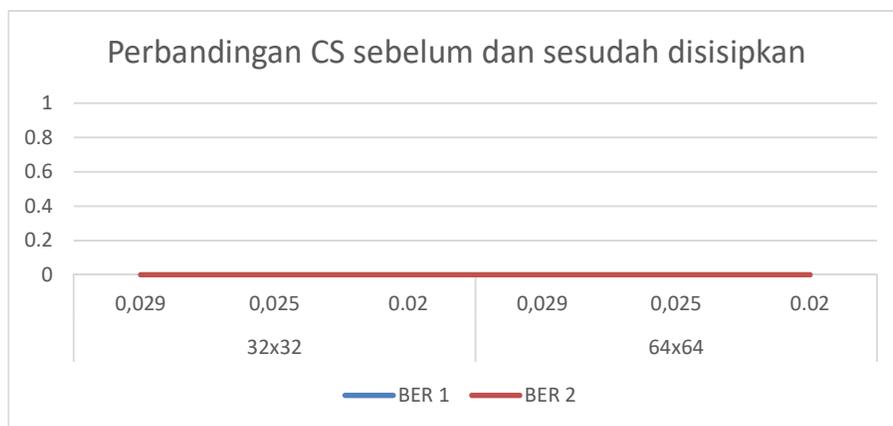
3.1 Pengujian dan Analisis Pengaruh *Measurement Rate*

Measurement rate (MR) dilakukan untuk mengetahui persentase rasio kompresi dan mengubah ukuran *watermark* agar menjadi lebih kecil untuk disisipkan ke dalam *host*. *Measurement rate* yang digunakan yaitu sebesar 0,02, 0,025, dan 0,029. Hasil pengujian dapat dilihat pada Tabel 1.

Measurement rate, perbedaan aliran musik dan ukuran *watermark* berpengaruh pada nilai SNR dan ODG, hal ini dapat dilihat dari Tabel 1. Semakin kecil nilai *measurement rate* maka SNR akan semakin besar. ODG yang dihasilkan mengalami perubahan yang sangat kecil dan tidak signifikan. Nilai BER dan PSNR yang dihasilkan yaitu 0 dan ∞ untuk seluruh jenis *host* dan *watermark*. Hal ini dikarenakan ukuran dari *watermark* diubah menjadi semakin kecil dari bentuk aslinya, sehingga ketika disisipkan ke dalam *host* bentuknya sudah lebih sederhana dan tidak membuat struktur dari *host* berubah secara signifikan. Semakin kecil nilai MR maka akan semakin kecil rasio kompresi (RK), artinya kompresi pada citra *watermark* memiliki bentuk yang semakin kecil. RK yang diperoleh paling optimal yaitu bernilai 62,5%. Perbedaan aliran musik menyebabkan nilai SNR dan ODG yang dihasilkan berbeda-beda. Hasil yang paling optimal dari pengujian ini yaitu dengan nilai SNR 85,32 dB, ODG $-8,34E-11$, BER 0, dan PSNR ∞ .

3.2 Pengujian Pengaruh Nilai K pada Klasifikasi K-NN

Analisis ini dilakukan untuk membandingkan nilai BER yang dihasilkan dari perbandingan antara *watermark* yang sudah dilakukan CS yang sebelum disisipkan ke dalam *host* dan yang sesudah disisipkan ke dalam *host*. Nilai *measurement rate* yang digunakan sebesar 0,02, 0,025, dan 0,029. Hasil pengujian dapat dilihat pada Gambar 6.



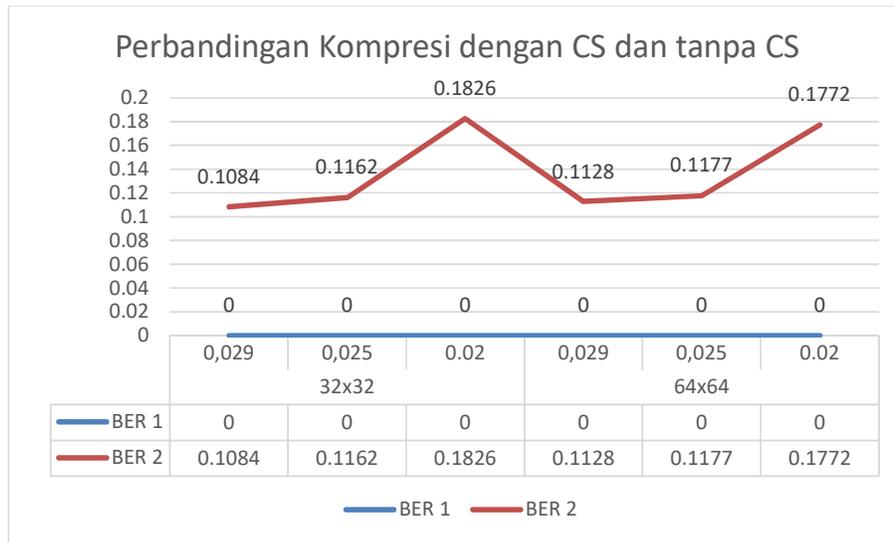
Gambar 6. Grafik BER *Watermark* Sebelum dan Sesudah di Sisipkan

Dari Gambar 6, BER 1 menunjukkan citra *watermark* sebelum disisipkan dan BER 2 sesudah disisipkan ke dalam *host*. Berdasarkan gambar tersebut, nilai BER sebelum disisipkan dan sesudah disisipkan bernilai 0. Hal ini menunjukkan bahwa tidak terjadi perubahan terhadap struktur dari *watermark*, selain itu proses CS dapat direkonstruksi dengan baik.

3.3 Perbandingan Kompresi tanpa CS dan dengan CS

Analisis ini dilakukan untuk membandingkan nilai BER yang dihasilkan dari *watermark* yang dikompresi tanpa menggunakan CS dan *watermark* yang dikompresi dengan menggunakan CS. Kompresi tanpa CS dilakukan dengan mengubah ukuran matriks citra *watermark* sesuai dengan ukuran matriks citra *watermark* yang dihasilkan dengan menggunakan CS pada *measurement rate* 0,02, 0,025, dan 0,029.

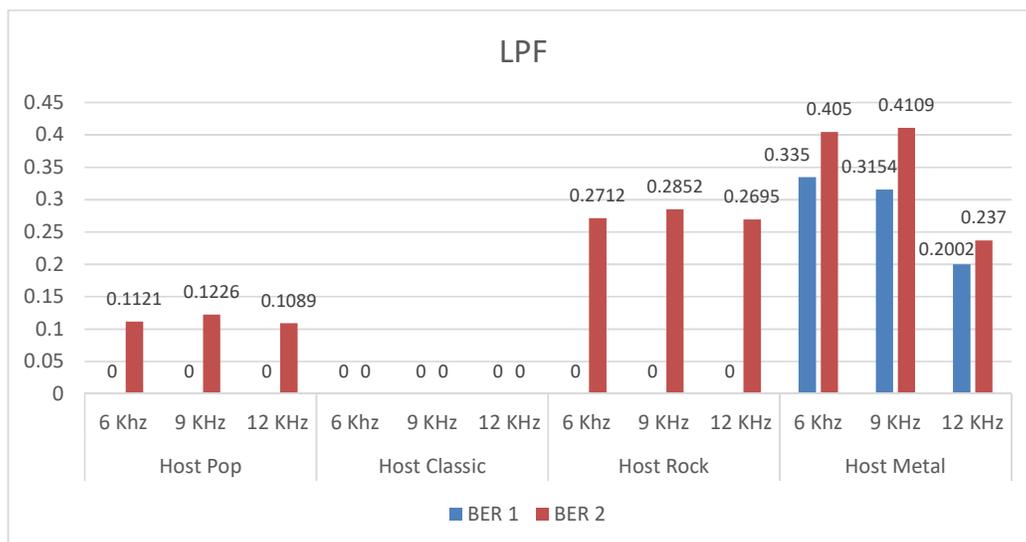
Dari Gambar 7, BER 1 menunjukkan nilai BER kompresi dengan menggunakan CS dan BER 2 menunjukkan nilai BER kompresi tanpa CS. Hasil kompresi sistem *audio watermarking* yang menggunakan CS lebih baik daripada kompresi yang tanpa menggunakan CS. Hal ini dapat dilihat dari nilai BER=0 untuk sistem yang menggunakan CS yaitu bernilai 0, sedangkan kompresi pada sistem yang tanpa menggunakan CS memiliki BER > 0.



Gambar 7. Grafik BER Kompresi dengan CS dan tanpa CS

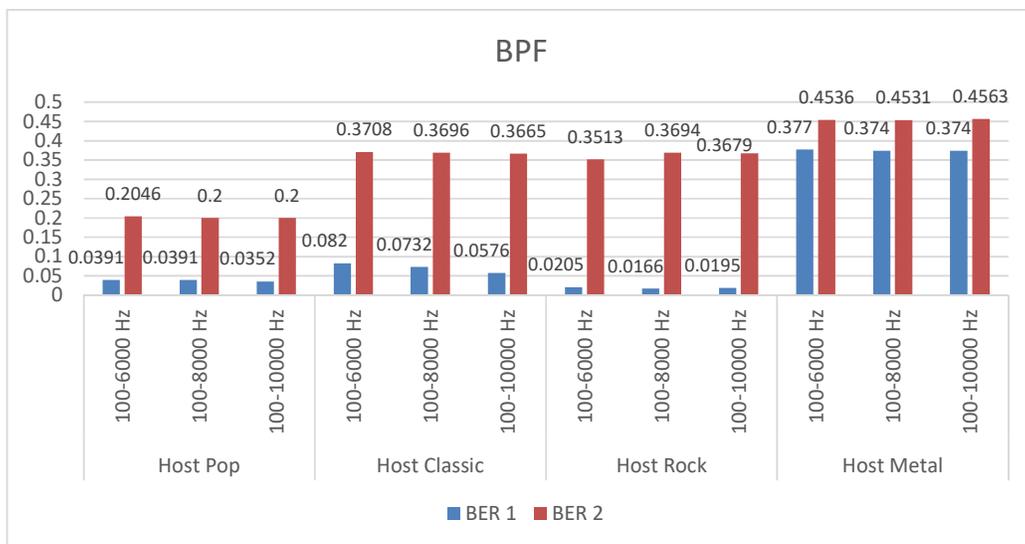
3.4 Kualitas Ketahanan *Watermark* Terhadap Serangan

Audio yang telah disisipkan dengan *watermark* diuji ketahanannya dengan cara diberikan serangan. Serangan tersebut berupa LPF, BPF, HPF, MP3 *Compression*, *Noise*, dan *Echo*. Pengujian ini dilakukan dengan mengukur nilai BER sebagai parameter uji kualitas. Hasil yang paling optimal dari masing-masing pengujian akan diambil. BER 1 menunjukkan *watermark* ukuran 32x32 dan BER 2 *watermark* ukuran 64x64.



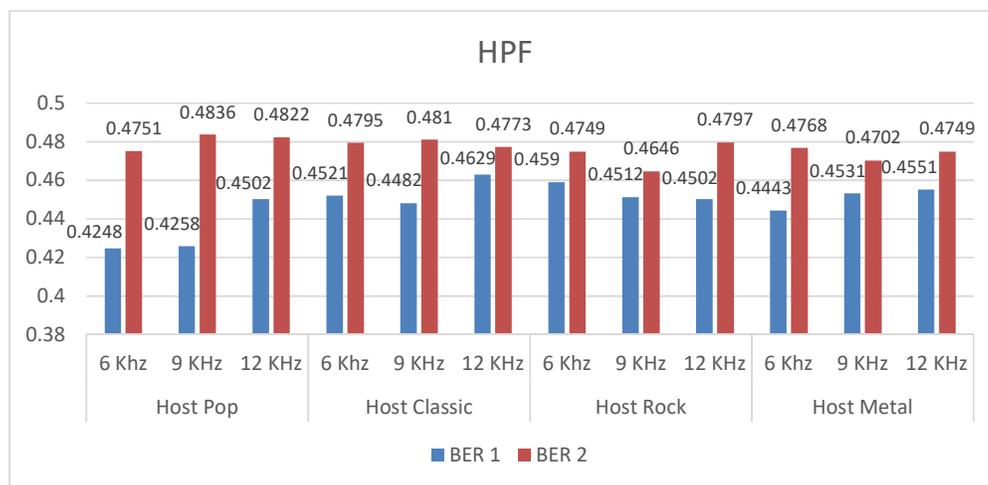
Gambar 8. Grafik BER Kompresi dengan CS dan tanpa CS

Sistem *compressive sensing audio watermarking* diuji dengan serangan berupa LPF dengan tiga macam frekuensi *cut-off*, yaitu 6 kHz, 9 kHz, dan 12 kHz. Hasil performansinya dapat dilihat dari nilai BER pada Gambar 8. Berdasarkan Gambar 8 tersebut, *watermark* 32x32 secara umum lebih tahan terhadap serangan LPF. Nilai BER=0 untuk *watermark* ukuran ini pada tiga jenis genre musik, yaitu *Pop*, *Classic* dan *Rock* pada semua frekuensi *cut-off* yang diuji, yaitu 6 kHz, 9 kHz, dan 12 kHz. Namun untuk *Metal*, nilai BER meningkat. Sedangkan *watermark* ukuran 64x64 hanya tahan pada serangan LPF pada genre musik *Classic*, nilai BER=0. Untuk jenis musik lainnya yaitu *Pop*, *Rock* dan *Metal*, *watermark* ukuran ini memiliki BER di atas nol sehingga kurang tahan terhadap serangan LPF.



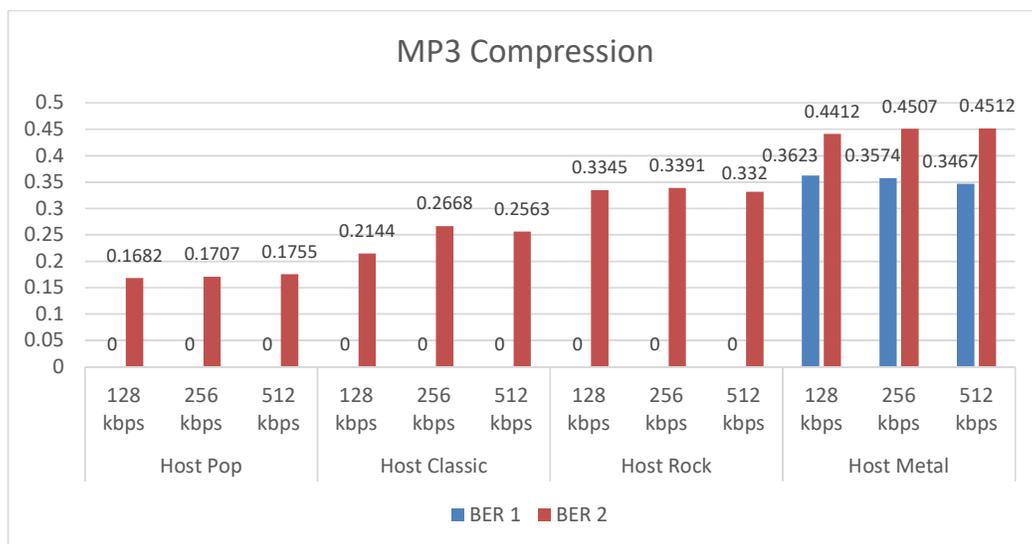
Gambar 9. Grafik BER terhadap Serangan BPF

Selain itu, sistem kami diuji pula dengan serangan BPF dengan tiga macam *frequency band*, yaitu 100-6000 Hz, 100-8000 Hz dan 100-10000 Hz. Hasil uji ketahanan sistem direpresentasikan dalam nilai BER seperti pada Gambar 9. Berdasarkan Gambar 9 tersebut, baik *watermark* ukuran 32x32 dan ukuran 64x64 memiliki BER>0 untuk semua jenis musik dan pada semua *frequency band* yang disimulasikan. Namun, *watermark* dengan ukuran 32x32 memiliki BER yang jauh lebih kecil daripada *watermark* ukuran 64x64. Dari hasil tersebut, *watermark* dengan ukuran lebih kecil lebih tahan terhadap serangan BPF.



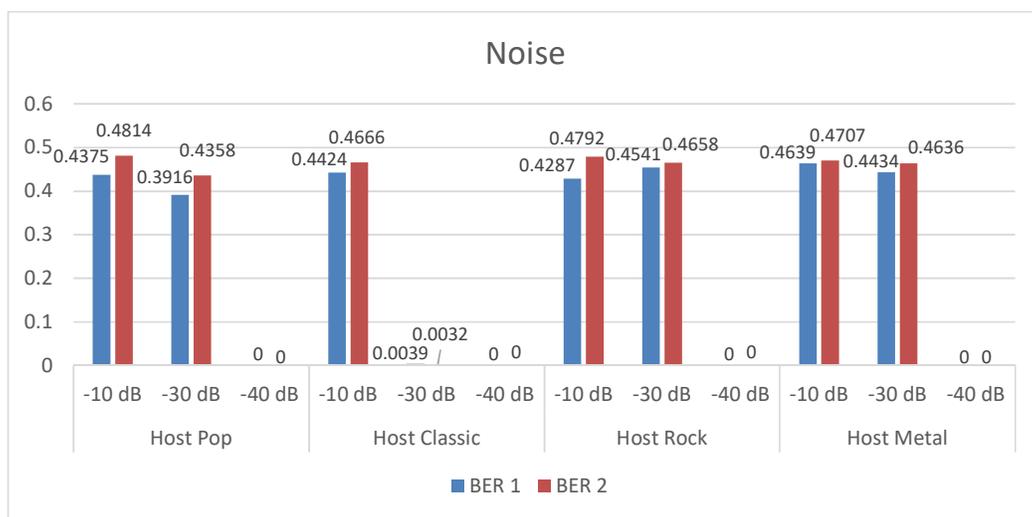
Gambar 10. Grafik BER terhadap Serangan HPF

Serangan HPF diterapkan pada sistem *compressive sensing audio watermarking* dengan tiga macam frekuensi *cut-off*, yaitu 6 kHz, 9 kHz, dan 12 kHz. Hasil uji ketahanan sistem direpresentasikan dalam nilai BER seperti pada Gambar 10. Berdasarkan gambar tersebut, baik *watermark* ukuran 32x32 dan ukuran 64x64 memiliki BER>0 untuk semua jenis musik dan pada semua frekuensi *cut-off*. Nilai BER tersebut jauh lebih buruk daripada jika sistem diserang dengan LPF dan BPF. Kedua macam ukuran *watermark* tidak tahan terhadap serangan HPF.



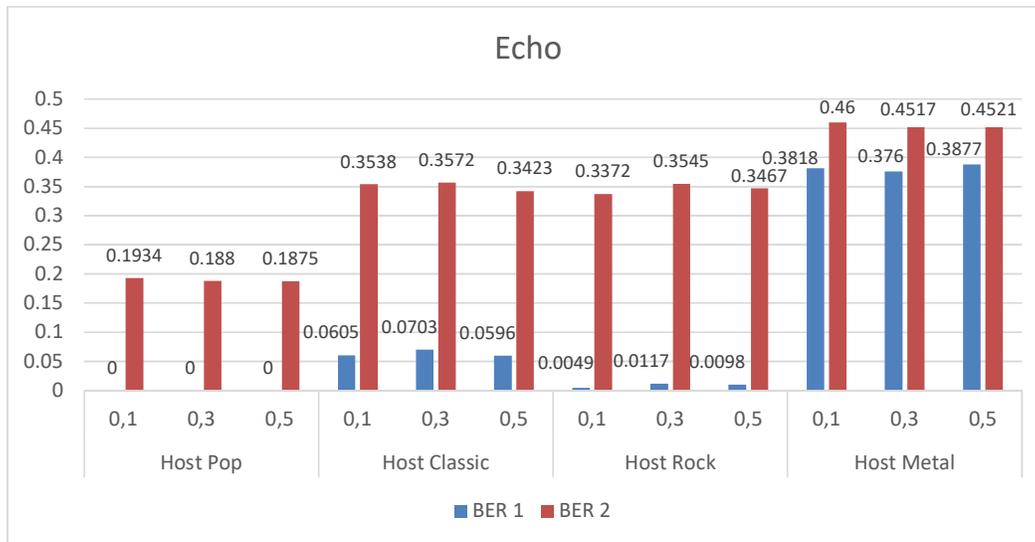
Gambar 11. Grafik BER terhadap Serangan MP3 Compression

Serangan lain yang ditambahkan pada sistem kami adalah berupa MP3 *compression* dengan tiga macam *bit rate*, yaitu 128 kbps, 256 kbps dan 512 kbps. Hasil performansi dapat dilihat dari nilai BER pada Gambar 11. Berdasarkan tersebut, *watermark* 32x32 secara umum lebih tahan terhadap serangan MP3 *compression*. Nilai BER=0 untuk *watermark* ukuran ini pada tiga jenis genre musik, yaitu *Pop*, *Classic* dan *Rock* pada semua *bit rate* yang diuji. Namun untuk *Metal*, nilai BER memburuk. Sedangkan *watermark* ukuran 64x64 memiliki BER>0 untuk semua jenis musik. *Watermark* dengan ukuran 64x64 lebih tidak tahan terhadap serangan MP3 *compression* dibandingkan dengan *watermark* ukuran 32x32.



Gambar 12. Grafik BER terhadap Serangan Noise

Pada sistem CS *audio watermarking* kami, serangan *noise* ditambahkan, dengan tiga macam *level noise* yaitu -10dB, -30 dB dan -40 dB. Performansi serangan *noise* terhadap sistem CS *audio watermarking* dapat dilihat pada Gambar 12. Pada gambar tersebut, nilai BER untuk *level noise* -10 dB sangat buruk untuk semua jenis musik. Sedangkan nilai BER untuk -30dB sangat buruk pada jenis musik *Pop*, *Rock* dan *Metal*. Terakhir, untuk *level noise* -40 dB, nilai BER=0 pada semua jenis musik. Sistem kami tahan terhadap serangan *noise* jika *level noise* tersebut adalah -40 dB dan tidak tahan terhadap serangan *noise* jika *level noise* -10 dB dan -30 dB.



Gambar 13. Grafik BER terhadap Serangan *Echo*

Serangan terakhir yang diterapkan pada sistem kami adalah serangan *echo* dengan tiga parameter *echo*, yaitu 0,1, 0,3 dan 0,5. Hasil performansi dapat dilihat dari nilai BER pada Gambar 13. Berdasarkan tersebut, *watermark* 32x32 memiliki nilai BER=0 pada jenis musik *Pop*. Nilai BER *watermark* dengan ukuran 32x32 jauh lebih baik daripada *watermark* dengan ukuran 64x64. Sehingga *watermark* 32x32 lebih tahan terhadap serangan *echo* dibandingkan dengan *watermark* 64x64. Namun pada jenis musik *Metal*, kedua *watermark* tersebut memiliki BER yang buruk.

4. KESIMPULAN

Setelah dilakukan pengujian dan analisis dari skema yang telah dirancang, maka dapat ditarik kesimpulan sebagai berikut :

1. Penerapan *Compressive Sensing* dengan mengubah nilai *measurement rate* berdampak terhadap Nilai SNR dan ODG, sedangkan nilai BER dan PSNR tetap, yaitu sebesar 0 dan ∞ . Dari hasil pengujian diperoleh nilai yang paling optimal yaitu dengan SNR 90,75 dB, ODG -6,93E-10, BER 0, dan PSNR ∞ , pada *Jazz watermark* 1 ukuran 16x16 *pixel* dengan MR 0,02.
2. CS pada citra *watermark* sebelum dan sesudah disisipkan ke dalam *host* memiliki nilai BER 0, tidak ada perubahan nilai BER yang terjadi. Rekonstruksi citra ke bentuk aslinya dikatakan berhasil dan baik.

3. Hasil kompresi yang diperoleh dengan menggunakan CS lebih baik daripada hasil kompresi yang diperoleh tanpa CS. Kompresi menggunakan CS menghasilkan BER bernilai 0, sedangkan kompresi tanpa CS bernilai di atas 0.
4. Secara umum *watermark* dengan ukuran 32x32 lebih tahan terhadap serangan dibandingkan dengan *watermark* dengan ukuran 64x64. Sistem kami secara umum tahan terhadap 6 macam serangan dari 7 macam serangan yang telah diuji. Sistem ini tahan terhadap serangan LPF, BPF, MP3 *Compression*, *Noise* dan *Echo*. Sedangkan sistem ini tidak tahan terhadap serangan HPF.

DAFTAR RUJUKAN

- Agradriya, B. A., Perdana, F. K., Safitri, I., & Novamizanti, L. (2017). Audio Watermarking Technique Based on Arnold Transform. *International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System and Information Technology (ICACOMIT)*, (pp. 17-21).
- Budiman, G., Suksmono, A. B., & Danudirdjo, D. (2016). Fibonacci Sequence Based FFT and DCT Performance Comparison in Audio Watermarking. *Pertanika Journal of Science & Technology (JST)*, 24(1), 1-10.
- Valens, C. (1999). The Fast Lifting Wavelet Transform. *The Math Forum*, (pp. 10-12).
- Candes, E. J., & Wakin, M. B. (2008). An Introduction to Compressive Sampling. *IEEE Signal Processing Magazine*, pp. 21-30.
- Jasmine, J. S., & Prabha, L. (2014). An Efficient Secure Image Watermarking using Wavelet Transform. *International Journal of Computer Trends and Technology (IJCTT)*, 17(3), 133-137.
- Bash, A. A., & Kayhan, S. K. (2015). Watermarked Compressive Sensing Measurement Reconstructed by the Greedy Algorithms. *International Journal of Computer Theory and Engineering*, 7(3), 219-222.
- Neetha, K. K., & Aneesh, M. K. (2015). A Compressive Sensing Approach to DCT Watermarking System. *International Conference on Control, Communication & Computing India (ICCC)*, (pp. 495-500).
- Dhar, P. K., Khan, M. I., & Kim, J. (2010). A New Audio Watermarking System using Discrete Fourier Transform for Copyright Protection. *International Journal of Computer Science and Network Security*, 10(6), 35-40.
- Lei, B., Soon, I. Y., Zhou, F., Li, Z., & Lei, H. (2012). A Robust Audio Watermarking Scheme based on Lifting Wavelet Transform and Singular Value Decomposition. *Signal Processing*, 92(9), 1985-2001.

- Li, Q., & Cox, I. J. (2007). Using Perceptual Models to Improve Fidelity and Provide Resistance to Volumetric Scaling for Quantization Index Modulation Watermarking. *IEEE Transaction of Information Forensics and Security*, 2(2), 127-139.
- Vujović, S., Daković, M., & Stanković, L. (2014). Comparison of the L1-Magic and the Gradient Algorithm for Sparse Signal Reconstruction. *22nd Telecommunications Forum TELFOR 2014* (pp. 577-580). IEEE.
- Safitri, I., Ginanjar, R. R., Rizal, A., & Azizah. (2017). Adaptive Multilevel Wavelet BCH Code Method in the Audio Watermarking System. *International Conference on Controls, Electronics, Renewable Energy and Communications (ICCEREC)*, (pp. 55-59).
- Lalitha, N. V., Prasad, P. V., & Rao, S. U. (2016). Performance Analysis of DCT and DWT Audio Watermarking based on SVD. *International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, (pp. 1-5).