

# PERANCANGAN DAN ANALISIS AUDIO WATERMARKING BERBASIS *DISCRETE SINE TRANSFORM (DST)* DENGAN METODE *LIFTING WAVELET TRANSFORM (LWT)* DAN *CENTROID*

## *DESIGNING AND ANALYZING DISCRETE SINE TRANSFORM (DST)-BASED AUDIO WATERMARKING USING LIFTING WAVELET TRANSFORM (LWT) AND CENTROID METHODS*

Muhammad Aprianda Rahmadi<sup>1</sup>, Gelar Budiman, 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

<sup>1</sup>[muhamad.aprianda@gmail.com](mailto:muhamad.aprianda@gmail.com), <sup>2</sup>[gelar.budiman@gmail.com](mailto:gelar.budiman@gmail.com), <sup>3</sup>[irmasaf@telkomuniversity.ac.id](mailto:irmasaf@telkomuniversity.ac.id)

### ABSTRAK

Penyebaran informasi dan data digital terutama lagu di internet telah berdampak negatif terhadap perlindungan hak cipta dari lagu tersebut. Untuk melindungi hak cipta pada lagu tersebut, maka di perlukan suatu teknik yang tidak hanya handal, namun tahan terhadap berbagai serangan yaitu teknik audio watermarking. Pada paper ini, dilakukan perancangan dan analisis *audio watermarking* dengan menggabungkan beberapa metode seperti *Discrete Sine Transform (DST)*, *Lifting Wavelet Transform (LWT)*, dan *Centroid*. Di metode LWT, Host audio akan di *segmentasi* melalui dua filter yaitu *High Pass Filter (HPF)* dan *Low Pass Filter (LPF)*. Setelah itu, *host* akan di *osilasi* dengan fungsi *sinus* amplituda dan frekuensi yang berbeda-beda sehingga korelasi unsur-unsur dari sinyal dapat dihilangkan dan diperoleh sinyal dalam domain frekuensi. Metoda *Centroid*, digunakan untuk mencari nilai amplituda titik pusat dari sinyal, setelah mendapatkan titik pusat dari *centroid* lalu *watermark* disisipkan dengan mengubah titik pusat nya menggunakan teknik QIM. Hasil penelitian ini, menunjukkan bahwa sistem audio watermarking yang dirancang memiliki parameter *robustness* yang baik terhadap serangan LPF, BPF (25-6k), Resampling, TSM, LSC, Kompresi MP3, dan kompresi MP4 dimana BER yang diperoleh <10 %. Memiliki kapasitas *watermark* yang besar yaitu 689,0625 bps, dan nilai SNR 24 dB.

**Keywords:** *Audio Watermarking, Lifting Wavelet Transform, Discrete Sine Transform, Centroid*

### ABSTRACT

Information distribution and digital data especially songs on the internet have bad impact on copyright protection for the songs themselves. To protect the copyright of the songs, an expert technique that is resistant to various attacks is required, namely the audio watermarking. In this paper, *audio watermarking* planning and analysis are conducted by combining several methods such as *Discrete Sine Transform (DST)*, *Lifting Wavelet Transform (LWT)*, and *Centroid*. In LWT method, Host audio will be segmented by two filters namely *High Pass Filter (HPF)* and *Low Pass Filter (LPF)*. After LWT decomposition, the selected subband is transformed to frequency domain by DST. *Centroid* method is used to find out the center point amplitude of the signal. then *watermark* is inserted by replacing the center point using the QIM technique. The result of the research shows that watermarking audio system which has arranged has good endurance parameters toward attacks of LPF, BPF (25-6k), Re-sampling, TSM, LSC, Compression MP3, and compression MP4 in which obtained BER is <10%, it has big payload parameter with 689,0625 bps, and 24 dB SNR value

**Keywords:** *Audio Watermarking, Lifting Wavelet Transform, Discrete Sine Transform, Centroid*

### 1. Pendahuluan

Saat ini media digital seperti video, audio, dan citra telah menggantikan peran media analog di berbagai aplikasi. Keberhasilan dari penerapan media digital ini disebabkan karena beberapa kelebihan yang dimiliki oleh media digital tersebut, seperti transmisi yang bebas noise, penyimpanan yang mudah, penyalinan yang sempurna, dan kemudahan untuk melakukan pengeditan. Di samping kelebihan yang dimiliki media digital tersebut, terdapat kelemahan dari penggunaan media digital yaitu mengenai hak cipta dan kebenaran konten dari suatu media digital tersebut. Digital *watermarking* dikembangkan sebagai solusi untuk mengatasi permasalahan tersebut [1]

Pada penelitian [2], diusulkan perancangan audio watermarking menggunakan algoritma LWT, dimana sinyal *embedded watermarking* disisipkan kedalam koefisien sub-band dengan memanfaatkan metode kuantisasi, deteksi audio watermarking dapat di implementasikan dengan cepat tanpa sinyal asli, namun metode tersebut tidak terbukti sangat

kuat terhadap berbagai serangan. Penelitian [3] metode watermarking audio yang di usulkan berdasarkan penyisipan ganda pada domain hibrida DWT-DST. Metode ini menggunakan variasi DWT dari dekomposisi domain waktu dan domain frekuensi untuk sinyal audio, dan memodifikasi koefisien DST dengan sebuah citra watermark sebelum rekonstruksi sinyal. Menghasilkan tingkat ketahanan yang tinggi dari berbagai serangan, Keefektifan algoritma berbasis DWT-DST ditunjukkan oleh 8 sinyal audio watermarking yang berbeda. Pada skema ini watermark tidak dapat di ekstraksi tanpa mengetahui kunci yang dihasilkan selama proses penyisipan.

Penelitian [4] di usulkan skema audio watermarking yang kokoh untuk proteksi hak cipta berdasarkan tingkat frame audio centroid. Watermark citra biner di sisipkan ke setiap segmen audio yang membawa frame centroid dengan menggunakan metode DCT dan metode kuantisasi ganjil. Ekstraksi watermark tidak memerlukan sinyal audio asli. Dimana hasil percobaan menunjukkan bahwa watermark yang kokoh tidak terdengar, level kestabilan frame centroid kokoh terhadap berbagai serangan dan mempunyai ketahanan yang tinggi.

Hal semacam distorsi, kapasitas penyisipan, dan ketahanan terhadap serangan menjadi indikator performansi dari sistem *watermarking* dengan penerapan metode *Quantisasi Index Modulation* (QIM) untuk penyisipan sinyal agar keamanan data meningkat [5]. Lalu pada penelitian [6] menghasilkan kapasitas yang besar dengan menggabungkan metode QIM dengan DWPT namun terdapat subband yang tidak digunakan dimana subband tersebut dapat meningkatkan embedding capacity yang lebih tanpa mengubah kualitas host audio dan menjaga ketepatan ekstraksi yang tinggi.

## 2. Basic Theory

### 2.1 Lifting Wavelet Transform (LWT)

Skema *lifting* diusulkan untuk mengurangi waktu komputasi, karena LWT menyederhanakan masalah dengan langsung menganalisis masalah dalam domain *integer* sehingga LWT menghitung lebih efektif dan hanya membutuhkan ruang memori yang kecil. Jadi dapat disimpulkan bahwa prinsip utama dari LWT adalah untuk membangun sebuah *wavelet* baru dengan karakteristik yang lebih baik berdasarkan pada *wavelet* yang sederhana. [7]

Berikut ini merupakan skema proses *Lifting Wavelet Transform* untuk domain audio [8][9]:

1. *Split*: merupakan proses pembagian sinyal ( $S_j = \{S_j, k\}$ ) dalam 2 bagian, bagian ganjil ( $O_{j-1}$ ) dan bagian genap ( $e_{j-1}$ ). Bagian yang terbagi merupakan dua titik potong dari sub-koleksi sinyal. Panjang dari tiap sub-koleksi sinyal yaitu setengah dari panjang aslinya.

$$Split(S_j) = (e_{j-1}, o_{j-1}) \quad 2.1$$

$$e_{j-1} = \{e_{j-1} = s_{j,2k}\} \quad 2.2$$

$$o_{j-1} = \{o_{j-1, k} = s_{j, 2k}\} \quad 2.3$$

2. *Predict*: langkah kedua merupakan prediksi. Prediksi digunakan untuk korelasi antara bagian genap dan bagian ganjil yang menggunakan salah satu bagian (biasanya bagian genap) untuk memprediksi bagian yang lain (bagian ganjil). Diferensial dari koleksi data  $d_{j-1}$  mengandung subset asli  $o_{j-1}$ . Jadi persamaan dari prediksi adalah sebagai berikut:

$$d_{j-1} = o_{j-1} - p(e_{j-1}) \quad 2.4$$

3. *Update*: Setelah proses *split*, salah satu fitur produk subset (seperti nilai rata-rata) mungkin tidak sama dengan data asli. Maka dari itu, untuk menjaga karakteristik umum dari data asli diperlukan adanya proses pembaharuan (*update*). Proses *update* ( $U$ ) bisa dijabarkan sebagai berikut

$$S_{j-1} = e_{j-1} + U(d_{j-1}) \quad 2.5$$

### 2.2 Discrete Sine Transform (DST)

*Discrete Sine Transform* (DST) merupakan teknik transformasi yang mengubah sinyal dari domain waktu/spasial ke domain frekuensi. Berbeda dengan transformasi *fourier* yang memiliki komponen sinus dan kosinus, DST hanya memiliki komponen sinus yang berosilasi dalam frekuensi yang berbeda beda. Dari definisi tersebut dapat di simpulkan bahwa DST hanya bekerja pada komponen real[10].

$$X(k) = \sqrt{\frac{2}{N+1}} \sum_{n=0}^{N-1} x(n) \cdot \sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right), k = 0, \dots, N - 1 \tag{2.6}$$

Dimana :

- x(n) merupakan sinyal dalam domain waktu
- n merupakan sample sinyal audio dalam domain waktu
- $\pi$  merupakan tetapan (baca *phi*) dengan nilai 3.14
- K merupakan sample sinyal audio domain frekuensi
- N merupakan panjang sinyal audio dalam proses DST
- Sementara untuk persamaan *inverst* DST adalah sebagai berikut:

$$x(n) = \sqrt{\frac{2}{N+1}} \sum_{k=0}^{N-1} X(k) \cdot \sin\left(\frac{\pi(k+1)(n+1)}{N+1}\right), k = 0, \dots, N - 1 \tag{2.7}$$

Dimana :

- X(k) merupakan sinyal dalam domain frekuensi
- n merupakan sample sinyal audio domain waktu
- $\pi$  merupakan tetapan (baca *phi*) dengan nilai 3.14
- K merupakan sample sinyal audio domain frekuensi
- N merupakan panjang sinyal audio dalam proses IDST.

### 2.3 Centroid

Dalam penelitian yang dilakukan Paliwal [11], di indkasikan bahwa Sub-band Spectrum Censored (SSC) merupakan yang paling mendekati puncak dari spectrum. Puncak katup dari spectrum [12] memiliki pengaruh yang relatif kecil saat telah di tempelkan, sehingga dapat di simpulkan bahwa spectrum centroid merupakan salah satu faktor yang mempengaruhi dimana harus di sisipkan audio *watermark*.

model campuran Gaussian Pada penelitian [13], di masukkan kedalam pengklasifikasian dan identifikasi masalah dari proses sinyal audio. Beragam karakteristik dan parameter telah dipilih dan dibandingkan, seperti spectrum centroid menjadi salah satu parameter yang di pilih dan hasilnya tingkat kesuksesannya lebih tinggi dibanding dengan parameter lainnya. Dapat di simpulkan bahwa spectrum centroid dalam hal hal parameter karakteristik memiliki tingkat karakteristik yang stabil. Secara singkat Centroid merupakan titik kesetimbangan dari vector atau pusat gravitasi dari sebuah spectrum. Centroid dapat di hitung dengan menggunakan rumus berikut:

$$C_i = \frac{\sum_{k=1}^{fL} k \frac{X_i(k)}{\sum_{k=1}^{fL} X_i(k)}}{\sum_{k=1}^{fL} X_i(k)} \tag{2.8}$$

Dimana nilai  $fL$  merupakan banyak data dalam satu frame dan nilai  $X_i(k)$  bisa di dapatkan dari rumus berikut:

$$X_i(k) = f(k) \times k \quad (k = 1, 2, 3, \dots, fL) \tag{2.9}$$

### 2.4 Quantization Index Modulation (QIM)

Kuantisasi pada dasarnya dapat diterapkan di domain waktu atau frekuensi atau setelah proses transformasi. Penyisipan dilakukan data *watermark* pada nilai maksimal sinyal. Penggunaan parameter dengan nilai yang diubah-ubah untuk mencari hasil penyisipan terbaik, parameter itu adalah  $\Delta$ . Nilai  $\Delta$  didapat dari banyaknya bit kuantisasi yang mana  $\Delta$  itu merupakan jarak antar nilai kuantisasi. Rumusan QIM untuk proses penyisipan dan proses ekstraksi pada *watermark* yang akan digunakan ialah sebagai berikut [14]:

#### A. Proses Penyisipan

Jika bit *watermark* yang disisipkan adalah bit '1', maka:

$$F'(0) = A_k \text{ dan } \arg \min |F(0) - A_k| \tag{2.10}$$

Jika bit *watermark* yang disisipkan adalah bit '0', maka:

$$F'(0) = B_k \text{ dan } \arg \min |F(0) - B_k| \tag{2.11}$$

Persamaan nilai  $A_k$  dan  $B_k$  adalah:

$$A_k = (2k + \frac{1}{2})\Delta \tag{2.12}$$

$$B_k = (2k - \frac{1}{2})\Delta \tag{2.13}$$

dimana  $k = 0, \pm 1, \pm 2, \dots$

B. Proses Ekstraksi

$$\tilde{v}(k) = \text{mod}(\text{ceil}(\frac{F(0)}{\Delta}), 2) \tag{2.14}$$

$$\Delta = \frac{1}{2^{(nbit-1)}} \tag{2.15}$$

$F(0)$  = *sample host* audio sebelum di kuantisasi

$F'(0)$  = *sample host audio* setelah disisipkan

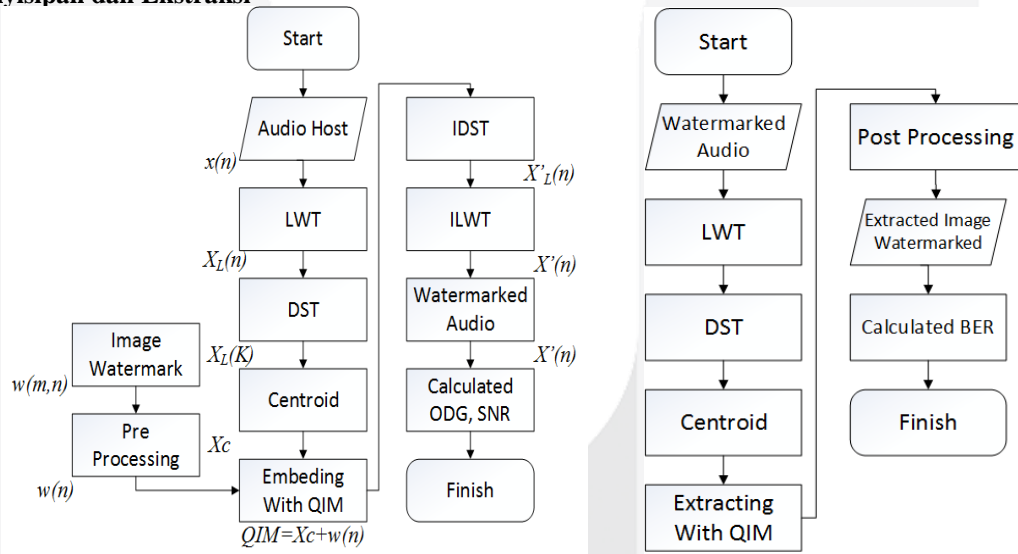
$\Delta$  = delta

$k$  = indeks kuantisasi

*nbit* = bit kuantisasi pada metode QIM

### 3. System Audio Watermarking Design

#### 3.1 Proses Penyisipan dan Ekstraksi



Gambar 3.2 Diagram alir Penyisipan (kiri) dan Ekstraksi (Kanan)

Tahap penyisipan dimulai dengan penginputan audio dalam format \*.wav. audio tersebut akan melewati beberapa proses untuk mendapatkan *watermarked audio*.

1. Membaca host audio  $x(n)$
2. Host audio  $x(n)$  di dekomposisi menggunakan LWT, untuk menghasilkan beberapa sub-band pada frekuensi rendah  $X_L(n)$  dan frekuensi tinggi  $X_H(n)$ . Dari beberapa sub-band yang di peroleh *watermark* akan di sisipkan pada frekuensi rendah  $X_L(n)$  karena pada frekuensi rendah akan menghasilkan daya rata-rata di atas *threshold*.

$$P_{low} = \frac{1}{N} \sum_{i=1}^n X_L^2(i) \qquad P_{high} = \frac{1}{N} \sum_{i=1}^n X_H^2(i)$$

3. Host audio pada frekuensi rendah  $x_L(n)$  di transformasi menggunakan DST untuk merubah dari domain waktu ke domain frekuensi  $X_L(k)$ .
4. Setelah mendapatkan koefisien-koefisien sinyal dalam domain frekuensi, melakukan proses centroid yaitu proses mencari titik berat sinyal  $X_C$ . Dimana titik berat sinyal tersebut akan di modifikasi untuk penyisipan.
5. Melakukan proses *pre-processing* yaitu mengubah watermark dari 2 dimensi  $w(m,n)$  ke 1 dimensi  $w(n)$ . Tujuannya adalah agar bisa di sisipkan ke dalam host audio yang mempunyai karakteristik 1 dimensi.
6. Host audio  $X_c$  dan *watermark*  $w(n)$  disisipkan menggunakan QIM.
7. Melakukan proses IDST, dan ILWT untuk memperoleh *watermarked audio*  $x'(n)$ .
8. Menghitung SNR, ODG, dan Kapasitas

Proses ekstraksi merupakan suatu proses memisahkan antara *host audio* dan *watermark* pada *watermarked audio* yang telah di rusak dengan sengaja. Secara umum, proses ekstraksi hampir sama dengan proses penyisipan hanya saja pada proses ekstraksi tidak lagi dilakukan proses *inverse*.

1. Membaca *watermarked audio*
2. *Watermarked audio* di dekomposisi menggunakan LWT, untuk menghasilkan beberapa sub-band pada frekuensi rendah (L) dan frekuensi tinggi (H). Dari beberapa sub-band yang di peroleh *watermark* akan di proses pada frekuensi rendah  $x_L(n)$  karena pada frekuensi rendah akan menghasilkan daya rata-rata di atas *threshold*.
3. Sinyal audio pada frekuensi rendah  $x_L(n)$  di transformasi menggunakan DST untuk merubah dari domain waktu ke domain frekuensi  $X_L(k)$ .
4. Setelah mendapatkan koefisien-koefisien sinyal dalam domain frekuensi, melakukan proses centroid yaitu proses mencari titik berat sinyal  $S_h$ . Dimana titik berat sinyal tersebut akan di modifikasi untuk penyisipan.
5. *Watermarked audio* selanjutnya akan di ekstraksi dengan QIM.
6. Menghitung BER

**4. Hasil dan Analisis**

**4.1 Parameter Sistem Sebelum Diserang**

Ada 5 parameter yang perlu diperhatikan untuk memperoleh SNR, BER, dan Kapasitas yang baik pada sistem ini. Parameter tersebut adalah level dekomposisi (N), panjang frame (N Frame), jumlah bit kuantisasi (nbit), type wavelet (typew), dan *threshold* (thr). Kelima parameter tersebut harus diatur untuk memperoleh hasil yang baik. parameter tersebut didapatkan dari pengujian pada host audio conversation.wav, dimana parameter tersebut akan dipakai untuk 5 host audio yang berbeda yaitu (conversation.wav, piano.wav, gitar.wav, drum.wav, dan bass.wav) untuk di lakukan pengujian dengan diserang oleh 12 jenis serangan. berikut ini merupakan tabel 4.1 parameter optimal yang di dapatkan sebelum dilakukan proses serangan, dan tabel 4.2 nilai SNR, BER, dan Capacity yang diperoleh.

Tabel 4.1 parameter optimal sistem sebelum di serang

| N | N Frame | nbit | thr | Typew |
|---|---------|------|-----|-------|
| 5 | 512     | 1    | 1   | 1     |

Tabel 4.2 hasil yang diperoleh dari parameter optimal sebelum diserang

| SNR    | BER | C      |
|--------|-----|--------|
| 32,832 | 0   | 2,6917 |

**4.2 Hasil Analisis Parameter Awal Setelah diserang oleh beberapa serangan**

Parameter optimal hasil pengujian conversation.wav pada sub bab 4.1, dilakukan percobaan serangan untuk 5 host audio yang berbeda (conversation.wav, piano.wav, gitar.wav, drums.wav, dan bass.wav). masing-masing host, diambil satu hasil yang menunjukkan BER yang kurang baik, sehingga akan dilakukan optimaasi parameter untuk setiap host dan serangan yang dipilih. Untuk serangan pada masing-masing host yang akan di optimasi dapat dilihat pada tabel 4.3 dibawah.

Tabel 4.3 host yang akan di optimasi parameter berdasarkan BER

| Serangan            | Parameter | BER      |           |           |           |          |
|---------------------|-----------|----------|-----------|-----------|-----------|----------|
|                     |           | host,wav | Piano,wav | Gitar,wav | drums,wav | bass,wav |
| BPF                 | 100-6K    | 0,36     | 0,41      | -         | -         | -        |
| Kompresi MP3        | 32k       | -        | -         | 0,36      | 0,37      | -        |
| Linier Speed Change | 5%        | -        | -         | -         | -         | 0,43     |

### 4.3 Hasil Optimasi Parameter 5 Host Audio























Tabel 4.5

| Hasil Optimasi | N | N Frame | n bit | thr | typew | SNR     | BER  | C        |
|----------------|---|---------|-------|-----|-------|---------|------|----------|
| parameter 1    | 3 | 32      | 2     | 1   | 1     | 17,7416 | 0,17 | 172,2656 |
| parameter 2    | 1 | 64      | 1     | 1   | 4     | 27,4743 | 0,06 | 344,5313 |
| parameter 3    | 2 | 64      | 1     | 1   | 1     | 17,2452 | 0,13 | 172,2656 |
| parameter 4    | 1 | 32      | 1     | 1   | 1     | 24,3685 | 0,08 | 689,0625 |
| parameter 5    | 1 | 256     | 1     | 1   | 1     | 46,4643 | 0,05 | 86,1328  |

Berdasarkan hasil optimasi parameter 1 sampai dengan parameter 5 diatas, masing-masing host audio akan di serang lagi oleh 12 jenis serangan. dimana yang akan diambil adalah host yang memiliki rata-rata BER paling kecil diantara host audio lain nya. Dalam hal ini, yang akan dipilih adalah drum.wav karena memiliki rata – rata 0,19. Berikut tabel 4.5 hasil optimasi yang dipilih untuk host audio (drum.wav)

Tabel 4.5 BER hasil optimasi pada instrumen drum.wav

| Attacks    | Fs        | BER (before) | BER (after) | Extracted Watermark |
|------------|-----------|--------------|-------------|---------------------|
| LPF        | 3000      | 0,42         | 0,03        |                     |
|            | 6000      | 0,37         | 0,02        |                     |
|            | 9000      | 0,22         | 0           |                     |
| BPF        | Fs        | BER (before) | BER (after) | Extracted Watermark |
|            | 100-6000  | 0,37         | 0,48        |                     |
|            | 50-6000   | 0,37         | 0,34        |                     |
| Noise      | Parameter | BER (before) | BER (after) | Extracted Watermark |
|            | 0 dB      | 0,51         | 0,47        |                     |
|            | 10 dB     | 0,48         | 0,49        |                     |
| Resampling | Fs        | BER (before) | BER (after) | Extracted Watermark |
|            | 22050     | 0,28         | 0,01        |                     |
|            | 11025     | 0,23         | 0,01        |                     |
| TSM        | Parameter | BER (before) | BER (after) | Extracted Watermark |
|            | 0,99      | 0,49         | 0,01        |                     |

|                        |                  |                     |                    |   |
|------------------------|------------------|---------------------|--------------------|---|
|                        | 0,98             | 0,48                | 0                  |    |
|                        | 0,96             | 0,51                | 0,01               |    |
| <b>LSC</b>             | <b>Parameter</b> | <b>BER (before)</b> | <b>BER (after)</b> | <b>Extracted Watermark</b>  |
|                        | 0,99             | 0,36                | 0,01               |    |
|                        | 0,95             | 0,32                | 0,01               |    |
|                        | 0,9              | 0,31                | 0,01               |    |
| <b>Pitch Shifting</b>  | <b>Parameter</b> | <b>BER (before)</b> | <b>BER (after)</b> | <b>Extracted Watermark</b>  |
|                        | 0,99             | 0,54                | 0,42               |    |
|                        | 0,98             | 0,58                | 0,49               |    |
|                        | 0,96             | 0,46                | 0,43               |    |
| <b>Equalizer</b>       |                  | <b>BER (before)</b> | <b>BER (after)</b> | <b>Extracted Watermark</b>  |
|                        |                  | 0,38                | 0,26               |    |
| <b>Echo</b>            |                  | <b>BER (before)</b> | <b>BER (after)</b> | <b>Extracted Watermark</b>  |
|                        |                  | 0,38                | 0,34               |    |
| <b>MP3 Compression</b> | <b>Fs</b>        | <b>BER (before)</b> | <b>BER (after)</b> | <b>Extracted Watermark</b>  |
|                        | 32000            | 0,37                | 0,08               |    |
|                        | 64000            | 0,26                | 0,06               |   |
|                        | 128000           | 0,05                | 0,05               |  |
|                        | 192000           | 0,02                | 0                  |  |
| <b>AAC Compression</b> | <b>Fs</b>        | <b>BER (before)</b> | <b>BER (after)</b> | <b>Extracted Watermark</b>  |
|                        | 32000            | 0,42                | 0,55               |  |
|                        | 64000            | 0,42                | 0,55               |  |
|                        | 128000           | 0,42                | 0,55               |  |
|                        | 192000           | 0,42                | 0,55               |  |
| <b>MP4 Compression</b> | <b>Fs</b>        | <b>BER (before)</b> | <b>BER (after)</b> | <b>Extracted Watermark</b>  |
|                        | 32000            | 0,32                | 0,05               |  |
|                        | 64000            | 0,32                | 0,05               |  |
|                        | 128000           | 0,32                | 0,05               |  |
|                        | 192000           | 0,32                | 0,05               |  |

### 5. Conclusion

Pada penelitian ini berhasil merancang skema *audio watermarking* dengan menggabungkan metode LWT, DST, Centroid, dan QIM dimana dari hasil analisis skema yang dirancang memiliki parameter ketahanan yang baik terhadap serangan LPF, BPF (25-6k), Resampling, TSM, LSC, Kompresi MP3, dan kompresi MP4 dimana BER yang diperoleh <10 %. Memiliki kapasitas watermark yang besar yaitu 689,0625 bps, dan nilai SNR >20 dB.

## 6. Daftar Pustaka

- [1] M. Rumondang, "Perlindungan Hak Cipta Pada Data Audio Menggunakan Teknik Watermarking Phase Coding Abstrak," *Perlindungan Hak Cipta Pada Data Audio Menggunakan Tek. Watermarking Phase Coding*, pp. 1–18, 2007.
- [2] Z. Tao, H. Zhao, J. Wu, J. Gu, Y. Xu, and D. Wu, "A Lifting Wavelet Domain Audio Watermarking Algorithm Based on the Statistical Characteristics of Sub-Band Coefficients," *Arch. Acoust.*, vol. 35, no. 4, pp. 481–491, 2010.
- [3] H. Yassine, B. Bachir, and K. Aziz, "A Secure and High Robust Audio Watermarking System for Copyright Protection," *Int. J. Comput. Appl.*, vol. 53, no. 17, pp. 33–39, 2012.
- [4] M. Fan and H. Wang, "Centroid-based robust audio watermarking scheme," *ICALIP 2008 - 2008 Int. Conf. Audio, Lang. /?/ Process. Proc.*, no. 1, pp. 476–479, 2008.
- [5] H. T. Hu, S. H. Chen, and L. Y. Hsu, "Incorporation of perceptually energy-compensated qim into dwt-dct based blind audio watermarking," *Proc. - 2014 10th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process. IHH-MSP 2014*, vol. 1, no. 1, pp. 748–752, 2014.
- [6] A. El Bahi and A. Adib, "A High Capacity Quantization-Based Audio Watermarking Technique Using the DWPT," *Int. Conf. Multimed. Comput. Syst.*, pp. 1–5, 2014.
- [7] J. Patel and K. Pathak, "Implementation of the 5 / 3 Lifting 2D Discrete Wavelet Transform," vol. 2, no. 3, pp. 2953–2957, 2014.
- [8] C. Xuesongl, C. Haiman, and W. Fenglee, "A Dual Digital Audio Watermarking Algorithm Based on LWT," no. Mic, pp. 721–725, 2012.
- [9] E. Nur, F. Astuti, G. Budiman, and L. Novamizanti, "PERANCANGAN TEKNIK LWT-DCT-SVD PADA AUDIO WATERMARKING STEREO DENGAN SINKRONISASI DAN COMPRESSIVE SAMPLING," pp. 1–6.
- [10] B. N. Madhukar and S. Jain, "A duality theorem for the discrete sine transform (DST)," *Proc. 2015 Int. Conf. Appl. Theor. Comput. Commun. Technol. iCATccT 2015*, pp. 156–160, 2016.
- [11] K. K. Paliwal, "SPECTRA : ^ SUB," pp. 7–10, 1998.
- [12] Q. Zhang, Z. Liu, and Y. Huang, "Adaptive Audio Watermarking Algorithm Based on Sub-band Feature," *J. Inf. Comput. Sci.*, vol. 2, no. February, pp. 305–314, 2012.
- [13] A. Ramalingam and S. Krishnan, "Gaussian mixture modeling of short-time fourier transform features for audio fingerprinting," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 4, pp. 457–463, 2006.
- [14] "USING PERCEPTUAL MODELS TO IMPROVE FIDELITY AND PROVIDE INVARIANCE TO VALUMETRIC SCALING FOR QUANTIZATION INDEX MODULATION WATERMARKING Qiao Li , Ingemar J . Cox Departments of Computer Science and Electronic and Electrical Engineering , University Colleg."