

## PERANCANGAN DAN ANALISIS COMPRESSIVE SAMPLING PADA AUDIO WATERMARKING STEREO BERBASIS DISCRETE SINE TRANSFORM DENGAN METODE HYBRID LIFTING WAVELET TRANSFORM-CEPSTRUM.

### DESIGN AND PERFORMANCE ANALYZE COMPRESSIVE SAMPLING ON AUDIO WATERMARKING STEREO BASED ON DISCRETE SINE TRANSFORM WITH HYBRID LIFTING WAVELET TRANSFORM AND CEPSTRUM METHOD

Ramadhan Prasetya Dahlan<sup>1</sup>, Gelar Budiman, S.T., M.T.<sup>2</sup>, Nur Ibrahim, S.T., M.T.<sup>3</sup>

<sup>123</sup>Fakultas Teknik Elektro Universitas Telkom, Bandung, Jawa Barat 40257

<sup>1</sup>dahlanprasetya@gmail.com, <sup>2</sup>gelarbudiman@telkomuniversity.ac.id, <sup>3</sup>nuribrahim.nib@gmail.com

---

#### ABSTRAK

Seiring dengan perkembangan jaringan internet dan teknologi media mengakibatkan penyebaran informasi berupa data, teks, video, gambar, dan audio menjadi sangat mudah untuk dimodifikasi secara ilegal. Dengan masalah tersebut untuk menjaga hak kepemilikan data digital diperlukan teknik *digital watermark*. Makalah ini akan menggunakan beberapa metode *watermarking* dan menggabungkan metode tersebut untuk melindungi data digital, metode yang digunakan adalah LWT (*Lifting Wavelet Transform*), *Cepstrum*, DST (*Discrete Sine Transform*) dalam audio stereo dengan CS (*Compressive Sampling*) menggunakan teknik penyisipan SMM (*Statistical Mean Manipulation*) dan QIM (*Quantization Index Modulation*). Audio asli dan *watermark* akan diubah menjadi beberapa *sub-band* tertentu oleh LWT. Kemudian hasil dari pemilihan *sub-band* LWT akan digunakan untuk transformasi *cepstrum*. Kemudian dilakukan transformasi DST dari hasil keluaran *cepstrum*. Setelah proses penyisipan, CS akan mengkompresi data dan gambar direkonstruksi secara tepat untuk dapat melakukan transformasi yang lebih cepat. Metode tersebut dipilih karena data *watermark* yang disisipkan kuat terhadap serangan seperti *linear speed change* dan TSM, namun kurang tahan terhadap serangan seperti *resampling*, *pitch shifting*, *echo*, dan *noise*. Hasil dalam penelitian ini menghasilkan kualitas audio dengan memiliki rata-rata BER 0.36651, rata-rata SNR 28.87 dB, rata-rata ODG -2.86832, dan rata-rata MOS 4.08.

**Kata Kunci:** *Digital Watermarking, LWT, DST, Cepstrum, Compressive Sampling.*

---

#### ABSTRACT

Along with the development of internet network and multimedia technology resulted in the deployment of information in the form of data, texts, videos, images, and audios become easier to modify it illegally. So with this problem to maintain digital ownership rights it is required digital watermarking techniques. This paper propose to protect the digital watermarked data with combining several methods i.e. LWT-Cepstrum-DST in a stereo audio with Compressive Sampling (CS) and the embedding process in this paper is used Statistical Mean Manipulation (SMM) and Quantization Index Modulation (QIM). The original audio and watermark will be transformed into some spesific sub-band by LWT. Then, it uses to transform with cepstrum. DST will be transform the output cepstrum. After embedding process, compressive sampling will be compressing the data and proper reconstruction image for faster transformation. Those methods selected because the watermarked data is robust against attacks such as linear speed change and TSM, but less robust against attacks such as resampling, pitch shifting, echo, and noise. The result shows that in this paper produces audio quality with an average SNR of 28.87 dB, average BER 0.36651, average ODG -2.86832, and MOS 4.08 against all audio genres tested.

**Keywords :** *Digital Watermarking, LWT, DST, Cepstrum, Compressive Sampling.*

---

#### I. PENDAHULUAN

Perkembangan jaringan internet dan penyebaran informasi yang semakin mudah menyebabkan semakin mudah untuk memodifikasi suatu data dari data aslinya. Hal ini telah menimbulkan berbagai macam masalah seperti pelanggaran hak cipta dan pembajakan suatu data. Dengan *digital watermarking*, masalah ini dapat diatasi dengan menyisipkan informasi digital (gambar, audio, atau video) ke dalam data asli. Dalam *digital watermarking*, ketika informasi disisipkan ke dalam audio digital, maka disebut

sebagai *audio watermarking*. *Audio watermarking* adalah teknik menyembunyikan data ke dalam host audio.

Umumnya, *audio watermarking* memiliki beberapa persyaratan dasar : *imperceptibility*, *robustness* (ketahanan), *capacity* (kapasitas), dan *security* (keamanan).

*Imperceptibility* [1] menunjukkan bahwa data *watermark* harus tidak terdengar oleh sistem pendengaran manusia, namun kebanyakan masih belum sempurna sehingga biasanya dilakukan pengujian dengan relawan. *Imperceptibility* akan menunjukkan kualitas audio dengan menghitung SNR. *Robustness* [2] adalah kemampuan untuk mengekstrak data *watermark* setelah dilakukan beberapa serangan pemrosesan sinyal seperti *noise* aditif, *cropping*, kompresi, *resampling*, dan lain-lain. *Capacity* [3] berarti data *watermark* harus dapat menampung sebagian besar bit yang disematkan. *Security* [4] harus terdeteksi hanya oleh orang yang berwenang. Makalah ini mengusulkan *audio watermarking* dengan menggunakan metode LWT-*cepstrum*-DST dengan *compressive sampling*.

Metode LWT (*Lifting Wavelet Transform*) [5] digunakan karena membuat *watermark* yang tertanam akan memiliki ketahanan yang lebih baik terhadap *watermark* yang disisipkan tahan terhadap serangan seperti *linear speed change*, TSM, dan khususnya dapat menahan penanaman secara acak. Namun kurang tahan terhadap serangan seperti *resampling*, *pitch shifting*, *echo*, dan *noise*. LWT [6] juga dirancang untuk mengurangi waktu komputasi dan kebutuhan memori. Prinsip utama LWT adalah untuk membangun *wavelet* baru dengan karakteristik yang lebih baik berdasarkan *wavelet* sederhana. *Cepstrum* [7] diperlukan untuk meningkatkan daya *imperceptibility* dan ketahanan pada *watermark* yang ditanam. Pemilihan *cepstrum* yang tepat akan membuat *watermarking* memiliki kemampuan anti serangan yang kuat dan kemampuan *anti-jamming*. DST [8] digunakan untuk menghemat waktu komputasi dengan menghilangkan kebutuhan waktu perhitungan, untuk menghemat secara maksimal waktu dan biaya komputasi. Proses *embedding* dalam makalah ini menggunakan teknik SMM dan QIM.

Data *watermark* akan disisipkan oleh SMM ke dalam host audio yang sudah dilakukan transformasi *cepstrum* sebelumnya dan data *watermark* yang akan disisipkan oleh QIM ke dalam host audio sudah dilakukan transformasi DST sebelumnya.. Untuk meningkatkan kapasitas *watermark*, sebelum proses *embedding* dilakukan pengompresian data menggunakan *compressive sampling*. Dalam *digital watermarking*, selain algoritma *embedding*, proses ekstraksi akan dilakukan setelah audio yang *watermark* diserang oleh serangan pemrosesan sinyal untuk memastikan bahwa audio yang *watermark* kuat terhadap berbagai macam serangan sinyal.

Algoritma transformasi dalam makalah ini menggunakan *cepstrum*. Ada beberapa keunggulan *cepstrum* seperti *blind detection*, menghitung cepat, memiliki jumlah kapasitas yang besar dalam proses penyimpanan data *watermarking*, ketahanan yang baik, dan juga memiliki kemampuan anti *jamming* yang baik. Disamping kelebihan tersebut terdapat masalah utama yaitu *cepstrum* tidak tahan terhadap kompresi. Li et al [9] mengusulkan metode pemetaan *watermarking cepstrum* dan SMM sebagai algoritma *embedding*. Hasilnya menunjukkan bahwa *watermark* yang tertanam kuat terhadap kompresi MP3 dan *noise* aditif. Di sisi lain ada paradigma baru yang disebut *compressive sampling*. Pengambilan sampel kompresif [10] dapat dengan sempurna mengembalikan sinyal dari sejumlah pengukuran linear non-adaptif yang tidak koheren, asalkan sinyal tersebut mengakui representasi langka dalam beberapa basis ortonormal, yang ditunjukkan dengan jumlah kecil koefisien nol dari beberapa ekspansi.

Makalah ini akan disusun dengan bagian-bagian sebagai berikut : Bagian II dasar teori; Bagian III algoritma metode yang diusulkan; Bagian IV analisis; terakhir, Bagian V kesimpulan dari makalah ini.

## II. DASAR TEORI

### 2.1 *Lifting Wavelet Transform*

LWT (*Lifting Wavelet Transform*) pada dasarnya dirancang untuk mengurangi waktu komputasi dan mengurangi kebutuhan kapasitas memori. LWT memiliki beberapa keunikan diantaranya :

1. Memiliki kemampuan implementasi yang cepat pada *wavelet transform* sehingga dapat dilakukan komputasi yang efisien dan mengurangi kebutuhan ruang pada memori.
2. Sangat mudah dalam membangun *non-linear wavelet transform*.
3. Memiliki kemampuan untuk lokalisasi frekuensi, sehingga dapat mengatasi kelemahan pada *traditional wavelet transform*.

Prinsip utama pada LWT untuk membangun sebuah *wavelet* yang baru dengan karakteristik lebih baik berdasarkan *simple wavelet*. Dan LWT memiliki tiga langkah untuk dapat menyelesaikan transformasi *wavelet*, yaitu: *split*, prediksi, dan *update*, yang merupakan dasar dari algoritma *lifting* [11].

### 1. *Split*

*Splitting* pada LWT adalah membagi data menjadi dua bagian yang lebih kecil (detail), yaitu  $e_{j-1}$  untuk urutan genap dan  $o_{j-1}$ , dan berikut adalah sinyal dalam bentuk sistematis :

$$S_j = \{S_j, k\} \quad (1)$$

Kemudian persamaan 1 dibagi menjadi dua bagian subset. Panjang masing-masing subset adalah setengah dari subset yang asli. Persamaan proses *split* pada LWT didefinisikan sebagai berikut :

$$(S_j) = e_{j-1}, o_{j-1}, \quad (2)$$

$$e_{j-1} = \{e_{j-1} = S_j, 2k\}, \quad (3)$$

$$o_{j-1} = \{o_{j-1}, k = S_j, 2k + 1\}. \quad (4)$$

Dan persamaan *Inverse LWT* (ILWT) digunakan untuk memproses sinyal kembali menjadi sinyal asli atau disebut proses ekstraksi. Persamaan ILWT didefinisikan sebagai berikut :

$$S_{j-1} = U(d_{j-1}) \quad (5)$$

### 2. *Predict*

*Predict* digunakan sebagai korelasi antara urutan genap dan urutan ganjil. Kemudian, *replace* detail sebagai perbedaan antara data dan prediksi, jika prediksi bersifat *reasonable*, perbedaan pada pengumpulan data ( $d_{j-1}$ ) akan menjadi lebih kecil, dan berisi informasi lebih sedikit daripada subset asli ( $o_{j-1}$ ). Persamaan proses prediksi didefinisikan sebagai berikut :

$$d_{j-1} = o_{j-1} - P(e_{j-1}), \quad (6)$$

Dimana ( $p_k$ ) merupakan fungsi antisipasi yang dapat dinyatakan operator prediksi ( $p$ ), fungsi  $p_k$  dapat memilih data yang sesuai ( $e_{j-1}$ ) itu sendiri. Dan persamaan prediksi di ILWT adalah sebagai berikut :

$$d_{j-1} = P(S_{j-1}) \quad (7)$$

### 3. *Update*

*Update* dilakukan untuk memperbarui dan mempertahankan karakteristik data asli. Proses *update* didefinisikan sebagai berikut :

$$S_{j-1} = e_{j-1} + U(d_{j-1}), \quad (8)$$

Dimana  $S_{j-1}$  adalah frekuensi rendah dari  $S_j$ .  $S_j$  sama halnya dengan fungsi antisipasi yang mana pembaruan aritmatik dapat mengambil fungsi yang berbeda.  $U$  juga mengambil fungsi yang berbeda, dan menghasilkan transformasi *wavelet* yang dibangun juga berbeda. Persamaan proses *update* di ILWT dapat didefinisikan sebagai berikut:

$$S_j = \text{Merge}(S_{j-1}, d_{j-1}). \quad (9)$$

## 2.2 *Cepstrum*

Transformasi *cepstrum* memiliki perhitungan yang kompleks, namun dapat mempertahankan semua sinyal dan semua informasi, jadi *cepstrum* dapat merekonstruksi sinyal. Digunakannya transformasi *cepstrum* adalah untuk menganalisis sinyal audio. *Cepstrum* akan menambah ketidaktersebutan imperseptibilitas dan menambah ketahanan pada *watermark* yang disisipkan. Pemilihan *cepstrum* yang tepat akan membuat *watermark* yang disisipkan memiliki kemampuan *anti-attack* dan kemampuan *anti-jamming*. Kemampuan ketahanan *audio watermarking* pada domain *cepstrum* [12] didasarkan pada persepsi *human auditory*. Dalam transformasi sinyal *cepstrum*, domain waktu diskrit akan dikonversi menjadi domain *cepstrum* diskrit.

$$\hat{s}(n) = \text{IDFT}(\log(\text{DFT}(s(n)))) \quad (10)$$

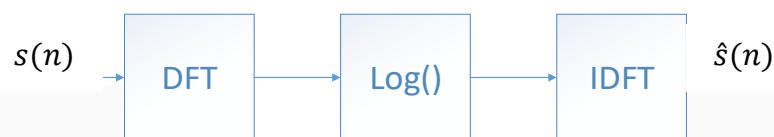
$$s(n) = IDFT(e^{DFT(\hat{s}(n))}) \quad (11)$$

Dimana  $s(n)$  adalah sinyal dalam domain dan  $\hat{s}(n)$  adalah sinyal dalam domain *cepstrum*. DFT ( $x(n)$ ) adalah fungsi dari transformasi *fourier* diskrit  $x(n)$  dengan formulasi sesuai dengan persamaan 12 dan IDFT ( $X(k)$ ) adalah *inverse* dari fungsi transformasi *fourier* diskrit terhadap  $X(k)$  dengan formulasi sesuai dengan persamaan 13.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi \frac{kn}{N}} \quad (12)$$

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{kn}{N}} \quad (13)$$

Dimana  $x(n)$  adalah sinyal dalam domain waktu diskrit,  $X(k)$  adalah sinyal dalam domain frekuensi diskrit,  $n$  adalah satuan diskrit dalam domain waktu dengan kisaran  $0 \leq n \leq N - 1$ ,  $k$  adalah unit diskrit dalam domain frekuensi, dan  $N$  adalah jumlah titik DFT yang digunakan.



Gambar 2. 1 Transformasi Cepstrum



Gambar 2. 2 Inverse Transformasi Cepstrum

### 2.3 Discrete Sine Transform

Transformasi DST memiliki fungsi atau sinyal dari segi jumlah sinusoid yang berbeda pada frekuensi dan amplitudo. DST beroperasi pada fungsi dijumlah titik data yang terbatas [8]. DST merupakan kelas matriks gabungan yang digunakan untuk mewakili sinyal atau fungsi tertentu. Hal ini juga diketahui bahwa satu fungsi dimensi atau sinyal yang menggambarkan serangkaian ortogonal dari fungsi dasar. Karena DST diterapkan pada durasi yang terbatas, sinyal atau fungsi waktu masukan berupa diskrit, maka diperlukan untuk menentukan apakah sinyal masukan ganjil atau genap pada kedua sisi kiri dan kanan dari batas domain.

DST identik dengan pergeseran *Discrete Fourier Transform* dari sinyal yang merupakan modifikasi permutasi tertentu dari sinyal asli, DST memiliki persamaan yang didefinisikan sebagai berikut [13]:

$$y(k) = \sum_{n=0}^{N-1} x(n) \sin\left[\frac{\pi kn}{N+1}\right], k = 0, \dots, N - 1 \quad (14)$$

DST juga memiliki pemadatan energi yang digunakan untuk *audio watermarking*. Persamaan IDST (*Inverse Discrete Sine Transform*) yang dapat di definisikan sebagai berikut :

$$x(n) = \frac{2}{N+1} \sum_{k=0}^{N-1} y(k) \cdot \sin\left[\frac{\pi kn}{N+1}\right], n = 0, \dots, N - 1 \quad (15)$$

## 2.4 Compressive Sampling

*Compressive sampling* pada dasarnya digunakan untuk menyederhanakan sinyal. CS (*Compressive sampling*) [14] memperkenalkan cara alternatif dari sampling sinyal yang berbeda dari pengambilan sampling standar, berdasarkan teorema *Shannon Nyquist*. Menurut teori CS, sample sinyal dapat diperoleh secara acak, pada tingkat jauh di bawah *Nyquist*. CS didasarkan oleh algoritma matematis yang kuat dan digunakan untuk rekonstruksi konten yang hilang. Untuk mendapatkan sinyal rekonstruksi dengan tingkat akurasi yang tinggi dapat diperoleh dari teknik rekonstruksi CS.

Secara perhitungan matematis, suatu sinyal disebut sebagai "*K-sparse*" jika  $\|x\|_0 \leq K \ll n$ , dimana  $l_0$  merupakan *quasi-norm* yang dapat dihitung dengan  $\|x\|_0 = \#\{j: x_j \neq 0\}$ . Pada CS, setiap pengukuran didapatkan dengan proyeksi sinyal aktual ke fungsi uji  $a_i$  [15] :

$$y_i = \langle x, a_i \rangle = a_i^T x, 1 \leq i \leq m \ll n \quad (16)$$

Atau secara sederhana dapat ditulis menjadi :

$$y = Ax \quad (17)$$

Proses rekonstruksi CS dapat menggunakan persamaan  $l_1$  norm yang didefinisikan sebagai berikut :

$$\hat{x} = \operatorname{argmin} \|x\|_1 \quad (18)$$

Tanda  $\| \cdot \|$  berarti norm atau panjang dari elemen yang terdapat didalamnya, atau jika  $\| \cdot \|$  seperti pada persamaan berarti norm dari elemen dari suatu ruang vektor *normed*. Dimana nilai tersebut merupakan panjang matriks yang dihasilkan setelah dilakukan proses ekstraksi.

## 2.5 Statistical Mean Manipulation

SMM [16] akan dilakukan dalam proses penyisipan dan ekstraksi. Sinyal dalam domain *cepstrum* dikurangi dengan nilai rata-ratanya dan ditambahkan atau dikurangkan dengan suatu faktor modifikasi rata-rata sesuai bit *watermark* yang disisipkan. Proses SMM dilakukan dalam setiap bit *watermark* yang disisipkan sesuai persamaan 2.19 berikut :

$$x_i^w = x_i - \mu_i + \alpha \cdot w_i \quad (19)$$

dimana  $i$  adalah segmen atau *frame* yang memiliki range :  $0 \leq i \leq N$ ,  $N$  adalah jumlah poin DFT yang merupakan jumlah sampel per segmen yang diproses pada setiap penyisipan bit *watermark*,  $x_i$  adalah sinyal dalam domain *cepstrum* pada *frame* atau segmen ke- $i$  sebelum disisipkan,  $x_i^w$  adalah sinyal dalam domain *cepstrum* pada segmen ke- $i$  setelah disisipkan,  $\mu_i$  adalah rata-rata sinyal  $x_i$ ,  $\alpha$  adalah faktor modifikasi rata-rata pada SMM, dan  $w_i$  adalah *watermark* yang dapat bernilai 1 atau -1. Citra *watermark* biner yang terbaca sebagai tipe *return to zero* (RZ) atau angka 0 dan 1, harus dikonversi dahulu ke tipe *non return to zero* (NRZ) atau 1 dan -1 agar dapat dilakukan penyisipan.

Proses ekstraksi *watermark* dilakukan dengan merata-ratakan sinyal pada satu segmen, jika hasilnya diatas 0 maka *watermark* yang dideteksi bernilai "1", namun jika hasil rata-ratanya di bawah 0 maka *watermark* yang dideteksi bernilai "0". Sehingga tidak diperlukan konversi dari NRZ ke RZ karena hasil akhir sudah diperoleh tipe bilangan RZ sesuai *watermark* aslinya. Berikut adalah proses ekstraksi SMM :

$$w_i = \begin{cases} 1, & \mu_i \geq 0 \\ 0, & \mu_i < 0 \end{cases} \quad (20)$$

dimana  $w_i$  adalah *watermark* hasil ekstraksi pada segmen ke- $i$  dan  $\mu_i$  adalah rata-rata sinyal pada segmen ke- $i$  pada domain *cepstrum* pada penerima atau pada proses ekstraksi.

## 2.6 Quantization Index Modulation

QIM [17] meyisipkan pesan dengan terlebih dahulu memodulasi urutan indeks dengan pesan yang akan disematkan, lalu menghitung sinyal host dengan urutan kuantitatif yang sesuai. Sebuah quantizer adalah fungsi yang memetakan nilai ke titik terdekat. QIM dibagi menjadi dua bagian, yaitu *encoding* dan *decoding*.

**2.6.1 QIM Encoding**

Fungsi *round* (.) menunjukkan nilai pembulatan ke bilangan bulat terdekat. Operasi kuantisasi dengan ukuran langkah  $\Delta$  didefinisikan sebagai berikut :

$$Q(x, \Delta) = \text{round} \left( \frac{x}{\Delta} \right) \Delta \tag{21}$$

Jika diasumsikan  $x$  sebagai sinyal host yang terdiri dari sampel  $N$  dan  $L$  mewakili panjang sinyal host,  $m$  adalah pesan yang disisipkan satu bit per sampel, maka kuantisasi dapat dituliskan dalam persamaan berikut :

$$d[n, 1] = \begin{cases} d[n, 0] + \frac{\Delta}{2}, & d[n, 0] < 0 \\ d[n, 0] - \frac{\Delta}{2}, & d[n, 0] > 0 \end{cases} \tag{22}$$

$$n = 1, 2, \dots, N/L$$

$d[n, 0]$  dan  $d[n, 1]$  digunakan untuk menyisipkan bit pesan "1" atau "0". Sinyal *watermark* dituliskan sebagai berikut :

$$y_n(x_n, m_n) = Q(x_n + d[n, m_n], \Delta) - d[n, m_n] \tag{23}$$

**2.6.2 QIM Decoding**

Selama proses deteksi, detektor menghitung dua sinyal yang terdiri dari  $S_r(n, 0)$  dan  $S_r(n, 1)$  dengan menyematkan "0" dan "1" pada sinyal yang diterima  $r$  secara terpisah dengan cara yang sama pada persamaan 23. Detektor mengumpulkan dua jarak *Euclidean* untuk sampel  $N$  kemudian menentukan bit pesan yang terdeteksi sebagai berikut:

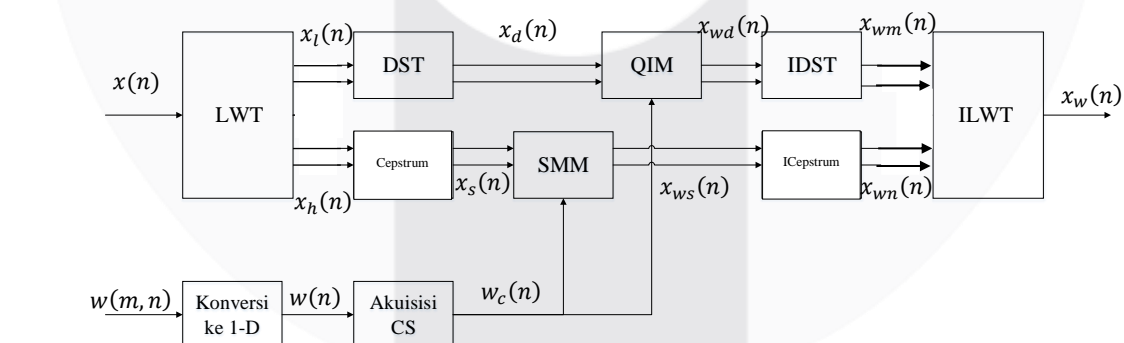
$$\hat{m}_n = \underset{l \in \{0,1\}}{\text{argmin}} \sum_{h=(n-1)N+1}^{nN} (r_h - S_r(h, l))^2 \tag{24}$$

$$n = 1, 2, \dots, L/N$$

**III. METODE WATERMARKING**

**3.1 Perancangan Sistem *Embedding* (Penyisipan)**

Proses *embedding* (penyisipan) sebuah data digital dalam audio memiliki beberapa tahap yang terdapat pada diagram blok di gambar 3.2 berikut :



Ket :

2 panah digunakan karena keluaran dari LWT memiliki *multi subband*

Gambar 3. 1 Proses embedding pada audio watermarking

Proses *embedding* sebuah informasi digital mempunyai beberapa tahap, yaitu :

1. Baca *file watermark*  $w(m, n)$  lalu ubah matriks dua dimensi menjadi satu dimensi matriks  $w(n)$ .
2. Lakukan kompresi  $w(n)$  dengan CS menggunakan persamaan 17 dan menghasilkan  $w_c(n)$  sebagai keluaran.
3. Baca *host audio*  $x(n)$  kemudian audio tersebut akan dibatasi sepanjang *watermark* ( $pjghost$ ) :

$$pjghost = length\ wt \times Nframe \times 2^N \tag{25}$$

Dimana,

- $N$  = tingkat dekomposisi dari LWT
- $length\ wt$  = panjang *watermark*
- $pjghost$  = panjang minimum *host audio*

4. Lakukan transformasi LWT sebanyak 2 tingkat dekomposisi yang akan menghasilkan *sub-band* sebanyak :

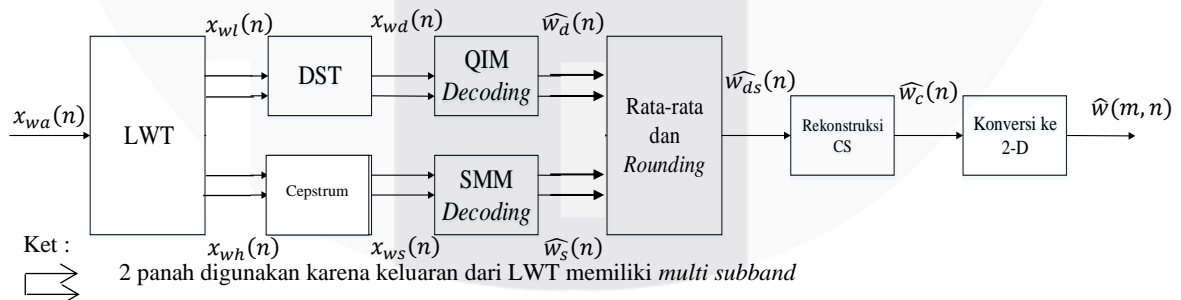
$$Subband = 2^N = 2^2 = 4 \tag{26}$$

Dimana LWT menghasilkan 4 *sub-band*. 2 *sub-band* dengan frekuensi tinggi yang diasumsikan sebagai  $x_h(n)$  dan 2 *sub-band* dengan frekuensi rendah yang diasumsikan sebagai  $x_l(n)$ .

5.  $x_l(n)$  digunakan sebagai masukan untuk transformasi DST, karena saat proses *embedding* dengan QIM harus menggunakan frekuensi rendah yang memiliki amplituda tinggi [18]. Kemudian DST akan mengubah menjadi domain frekuensi dari domain waktu berdasarkan persamaan 14. Keluaran dari proses ini akan diasumsikan sebagai  $x_d(n)$ .
6.  $x_h(n)$  digunakan sebagai masukan untuk transformasi *cepstrum*, yang akan mengubah menjadi domain *cepstrum* dari domain waktu berdasarkan persamaan 10. Keluaran dari proses ini akan diasumsikan sebagai  $x_s(n)$ .
7.  $w_c(n)$  disisipkan kedalam matriks  $x_d(n)$  pada posisi frekuensi tinggi dengan metode QIM yang menggunakan persamaan 23. Keluaran dari proses ini diasumsikan sebagai  $x_{wd}(n)$ .
8.  $w_c(n)$  disisipkan kedalam matriks  $x_s(n)$  pada posisi frekuensi rendah dengan metode SMM yang menggunakan persamaan 19. Keluaran dari proses ini diasumsikan sebagai  $x_{ws}(n)$ .
9. *Inverse DST* adalah proses mengembalikan dari domain frekuensi menjadi domain waktu menggunakan persamaan 15 dan menghasilkan  $x_{wm}(n)$ .
10. *Inverse cepstrum* adalah proses mengembalikan dari domain *cepstrum* menjadi domain waktu menggunakan persamaan 11 dan menghasilkan  $x_{wn}(n)$ .
11. Menggabungkan *sub-band*  $x_{wm}(n)$  dan  $x_{wn}(n)$  dengan *inverse LWT* akan menghasilkan *audio ter-watermark* yang diasumsikan sebagai  $x_w(n)$ . Kemudian kualitas dari *audio ter-watermark* dapat dihitung SNRnya dan ODGnya.

### 3.2 Perancangan Sistem *Extracting* (Ekstraksi)

Proses ekstraksi pada *watermark* mempunyai beberapa tahap yang terdapat pada diagram blok di gambar 3.4 :



Gambar 3.2 Proses *extracting* pada *audio watermarking*

Proses *extracting* adalah proses pengambilan *watermark* setelah diserang oleh beberapa serangan pemrosesan sinyal untuk menghitung nilai BER dan memiliki beberapa tahap, yaitu :

1. Membaca file audio yang sudah telah diserang sebagai  $x_w(n)$ .
2. Lakukan transformasi LWT sebanyak 2 tingkat dekomposisi yang akan menghasilkan *sub-band* sebanyak :

$$\text{Subband} = 2^N = 2^2 = 4 \quad (27)$$

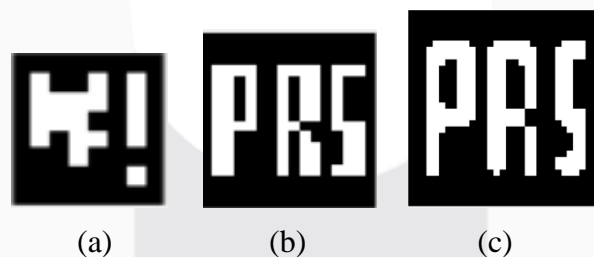
Dimana LWT menghasilkan 4 *sub-band*. 2 *sub-band* dengan frekuensi tinggi yang diasumsikan sebagai  $x_{wh}(n)$  dan 2 *sub-band* dengan frekuensi rendah yang diasumsikan sebagai  $x_{wl}(n)$ .

3.  $x_{wl}(n)$  ditransformasikan oleh DST yang menggunakan persamaan 15. Keluaran dari proses ini akan diasumsikan sebagai  $x_{wd}(n)$ .
4.  $x_{wh}(n)$  ditransformasikan oleh *cepstrum* yang menggunakan persamaan 11. Keluaran dari proses ini akan diasumsikan sebagai  $x_{wd}(n)$ .
5.  $x_{wd}(n)$  disisipkan kedalam posisi frekuensi tinggi berdasarkan prinsip ekstraksi QIM pada persamaan 24. Keluaran dari proses ini diasumsikan sebagai  $\widehat{w}_d(n)$ .
6.  $x_{ws}(n)$  disisipkan kedalam posisi frekuensi tinggi berdasarkan prinsip ekstraksi SMM pada persamaan 20. Keluaran dari proses ini diasumsikan sebagai  $\widehat{w}_s(n)$ .
7. Kemudian hitung rata-rata dan hitung *rounding* dengan QIM menggunakan persamaan 21. keluaran dari proses ini diasumsikan sebagai  $\widehat{w}_{ds}(n)$ .
8. Proses rekonstruksi CS menggunakan  $l_1$  norm yang dapat dihitung dengan persamaan 18 dan hasilnya diasumsikan sebagai  $\widehat{w}_c(n)$ .
9. *Final Processing*. Bit *watermark* yang terekstraksi, disusun menjadi matriks dua dimensi untuk mendapatkan *image* hasil ekstraksi  $\widehat{w}(m, n)$ .
10. Hitung nilai BER.

#### IV. ANALISIS

##### 4.1 Analisis Kinerja CS Terhadap *Watermark*

Dalam subbab ini akan menjelaskan pengaruh kinerja CS terhadap *watermark*. Pada pengujian ini, *watermark* dan bit compression diaplikasikan beberapa kali untuk mendapatkan hasil rasio kompresi dan *elapsed time* (s). Pemilihan rasio kompresi ini terlihat dari nilai *elapsed time* (s) terkecil. Pengujian CS menggunakan sisi 8, 16, 32 piksel dan beberapa kali mengubah rasio kompresi bit 0,023, 0,024, 0,025, 0,026, 0,027, 0,028, 0,029, dan 0,03. Hasil pengujian menunjukkan bahwa jika semakin besar rasio kompresi bit maka semakin besar rasio kompresi, namun akan membuat *elapsed time* (s) lebih kecil. Pada hasil sisi pixel tinggi memiliki kualitas gambar yang lebih baik namun hasil ekstraksi *watermark* lebih besar, sehingga waktu yang dibutuhkan pada setiap uji lebih panjang. Hasil ekstraksi untuk setiap sisi pixel dapat dilihat pada gambar 4.1.



Gambar 4. 1 (a) 8-Pixel, (b) 16-Pixel, (c) 32 Pixel

Pada penelitian ini menggunakan sisi 16 pixel dengan rasio kompresi bit 0,025, karena sisi 16 pixel memiliki kualitas gambar yang baik, pemilihan rasio kompresi bit 0,025 dipilih karena menghasilkan rasio kompresi 75%, jika rasio kompresi bit yang dipilih adalah 0,03 maka akan menghasilkan rasio kompresi 100%, sehingga parameter kinerja CS tidak akan mempengaruhi penelitian ini dan juga dengan rasio kompresi bit 0,025, sisi 16 pixel hanya membutuhkan waktu lebih sedikit dari pada sisi 32 pixel.



#### 4.2 Analisis Parameter Awal

Pada bagian ini dijelaskan kinerja *audio watermarking* dengan melakukan beberapa perubahan parameter, untuk mendapatkan parameter terbaik dapat dilihat dari hasil keluaran ODG, SNR, BER, dan kapasitas (C). Parameter yang akan dianalisis adalah n (tingkat dekomposisi), N Frame (jumlah sampel per frame), thr (ambang batas untuk wavelet), nbit (bilangan kuantisasi), skema *hybrid* (skema penggabungan antara dua metode), *typew* (jenis filter wavelet yang digunakan), dan alpha (parameter *cepstrum*). Dalam pengujian ini dijelaskan pengaruh perubahan parameter pada beberapa parameter penting dalam proses *audio watermarking*. Tes pada subbab ini mencari nilai ekstraksi *watermark* yang tidak diserang dan memiliki nilai SNR di atas 20 dB dan nilai BER mendekati 0.

Tabel 4. 1 Parameter Terbaik Sebelum Diserang

N	N Frame	nbit	thr	typew	alfa	Skema hybrid	ODG	SNR	BER	C
2	512	4	0,09	1	0,001	3	-3,8365	25,2762	0	11,2152

#### 4.3 Analisis Ketahanan Watermark Terhadap Serangan

Bagian ini akan diuji parameter terbaik sebelum serangan menggunakan 5 jenis audio dengan serangan pemrosesan sinyal yang diberikan seperti *Low Pass Filter (LPF)*, *Band Pass Filter (BPF)*, *noise*, *resampling*, *Time Scale Modification (TSM)*, *Linear Speed Change*, *Pitch Shifting*, *Equalizer*, *echo*, dan *mp3 compression*. Hasil setelah pengujian parameter terbaik sebelum diserang menunjukkan bahwa ada nilai BER yang diatas 0,4. Sehingga optimasi diperlukan untuk mendapatkan parameter yang menghasilkan nilai BER lebih kecil. Optimasi akan dilakukan pada 5 sampel acak dari masing-masing audio.

#### 4.4 Analisis Optimasi Terhadap Serangan







Optimasi dilakukan dengan mengubah parameter seperti subbab 4.2 terhadap serangan spesifik. Setelah mendapatkan parameter optimasi terbaik, masing-masing parameter akan diuji terhadap semua serangan pada 5 audio yang berbeda untuk menentukan ketahanan dan *imperceptibility watermark*. Hasilnya setelah tes penulis memilih parameter terbaik setelah optimasi yang bisa dilihat pada tabel 4.2.
















Tabel 4. 2 Parameter Terbaik Setelah Diserang dan Dioptimasi

N	N Frame	nbit	thr	typew	alfa	Skema hybrid
2	512	7	0,00001	4	0,001	3

Setelah memilih parameter terbaik, parameter optimasi akan diuji terhadap semua serangan dan menghasilkan ketahanan terbaik dari host.wav, hasil data *watermark* dapat dilihat pada tabel 4.3.

Tabel 4. 3 Ekstraksi Watermark pada Host Audio

Tipe Serangan	Parameter	BER	Ekstraksi Watermark	Tipe Serangan	Parameter	BER	Ekstraksi Watermark
LPF	3k	0,38281		Noise	0 dB	0,46484	
	6k	0,30078			10 dB	0,45313	
	9k	0,25781			20 dB	0,45313	

TSM	1%	0,35156		Linear Speed Change	1%	0	
	2%	0,41016			5%	0	
	4%	0,48047			10%	0	
Resampling	22.05k	0,375		MP3 Compression	Echo	0,50391	
	11.025k	0,41797			32k	0,45313	
	16k	0,35938			64k	0,09375	
Equalizer		0,050781		128k	0,01171 9		
				192k	0		

Selain BER, kualitas audio merupakan bagian terpenting dari sistem *audio watermarking*, karena jika kualitas *audio watermarking* buruk, maka harus diperbaiki lagi pada sistem *audio watermarking*. *Audio watermarking* yang berkualitas harus memiliki nilai SNR di atas 20 dB, dan *watermark* yang disisipkan tidak boleh terdengar oleh pendengaran manusia. Berikut adalah hasil pengujian kualitas audio:

Tabel 4. 4 Kualitas Audio

Host	SNR	ODG
	dengan CS,	dengan CS
host.wav	27,6183	-3,7819
piano.wav	35,2858	-3,6936
gitar.wav	39,149	-1,4817
drums.wav	21,599	-3,9112
bass.wav	20,699	-1,4732

Pada tabel 4.4 dapat dilihat bahwa audio memiliki kualitas yang baik, karena host.wav, piano.wav, gitar.wav, drums.wav, dan bass.wav memiliki nilai SNR di atas 20 dB. Nilai ODG yang tidak mendekati 0 menunjukkan bahwa metode yang digunakan tidak baik pada sisi *Objective Different Grade* setelah diserang. Namun ia memiliki nilai BER yang cukup bagus terhadap beberapa serangan. Untuk memastikan bahwa sistem menghasilkan kualitas *audio watermarking* yang baik, survei Mean Opinion Score (MOS) diperlukan. Survei MOS dilakukan pada 30 responden setelah mendengarkan 5 host audio asli dan 5 host audio yang diberi *watermark*, dengan skala nilai MOS 1 sampai 5. Jika nilai MOS lebih rendah maka audio yang diberi *watermark* sangat buruk dan jika nilai MOS dekat dengan 5 maka audio ter-*watermark* sangat mirip dengan audio aslinya. Hasil rata-rata nilai MOS yang telah diamati oleh 30 responden dapat dilihat pada Tabel 4.5:

Tabel 4. 5 Rata-rata Nilai MOS

Tipe Audio	Rata-rata MOS
host.wav	4,13
piano.wav	3,83
gitar.wav	4,7
drums.wav	3,867
bass.wav	3,9
Average	4,08

Dapat dilihat dari tabel di atas bahwa nilai MOS piano.wav, drums.wav, dan bass.wav rata-rata lebih besar dari 3. Sistem ini menunjukkan bahwa *watermark* yang disisipkan terganggu dan dapat terdengar oleh pendengaran manusia. Namun tipe audio host.wav dan gitar.wav mencapai nilai di atas 4 yang berarti *watermark* sedikit terdengar tapi tidak mengganggu.

## V. KESIMPULAN

Metode *audio watermarking* berdasarkan LWT, *cepstrum*, DST, dan CS yang ada pada makalah ini, dapat ditarik kesimpulan sebagai berikut :

1. Sistem *audio watermarking* yang diimplementasikan pada *file* audio \*.wav menggunakan metode LWT-*cepstrum*-DST-CS pada berbagai serangan seperti *Low Pass Filter* (LPF), *Band Pass Filter* (BPF), *noise*, *resampling*, *Time Scale Modification* (TSM), *Linear Speed Change*, *Pitch Shifting*, *equalizer*, *echo*, dan kompresi mp3, memiliki BER 0.36651, SNR 28.87 dB, ODG -2.86832, dan MOS 4.08.
2. Selain nilai BER, nilai ODG pun menjadi pertimbangan dalam pemilihan parameter yang tepat. Meskipun nilai rata-rata BER kecil pada suatu parameter namun jika ODGnya bernilai kecil maka hasil kualitas audio pada sistem *audio watermarking* tersebut dapat membuat audio ter-*watermark* terdengar oleh pendengaran manusia, dan membuat penilaian *Mean Opinion Score* menjadi tidak maksimal.

## DAFTAR REFERENSI

- [1] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, no. 7, pp. 1079–1107, 1999.
- [2] J. Bajpai, "A Literature Survey – Various Audio Watermarking Techniques and their challenges," *6th Int. Conf. Cloud Syst. Big Data Eng. (Confluence)*, 2016, pp. 451–457, 2016.
- [3] A. Kaur, M. K. Dutta, K. M. Soni, and N. Taneja, "A Secure and High Payload Digital Audio Watermarking Using Features From Iris Image," *Int. Conf. Contemp. Comput. Informatics (IC3I)*, 2014, pp. 509–512, 2014.
- [4] M. Hemis and B. Boudraa, "Digital watermarking in audio for copyright protection," *Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*, 2014, pp. 189–193, 2014.
- [5] Z. Tao, H. Zhao, J. Wu, and J. Gu, "A Lifting Wavelet Domain Audio Watermarking Algorithm Based on the Statistical Characteristics of Sub-Band Coefficients," *J. Inst. Fundam. Technol. Polish Acad. Sci.*, vol. 491, pp. 481–491, 2010.
- [6] P. K. Dhar, "A Blind Audio Watermarking Method Based on Lifting Wavelet Transform and QR Decomposition," *Int. Conf. Electr. Comput. Eng. (ICECE)*, 2014, pp. 136–139, 2014.
- [7] X. Zhang and Y. Hao, "An adaptive audio watermarking algorithm based on cepstrum transform," *Fifth Int. Jt. Conf. Comput. Sci. Optim. (CSO)*, 2012, vol. 1, pp. 806–809, 2012.
- [8] B. N. Madhukar, "A Duality Theorem for the Discrete Sine Transform ( DST )," *Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, 2015, pp. 156–160, 2015.
- [9] V. B. K, I. Sengupta, and A. Das, "Audio Watermarking Based on Mean Quantization in Cepstrum Domain," *16th Int. Conf. Adv. Comput. Commun. 2008. ADCOM 2008.*, pp. 73–77, 2008.

- [10] G. Valenzise, G. Prandi, M. Tagliasacchi, and A. Sarti, "Identification of Sparse Audio Tampering Using Distributed Source Coding and Compressive Sensing Techniques," *EURASIP J. Image Video Process.*, no. September 2008, pp. 1–23.
- [11] C. Xuesong, C. Haiman, and W. Fenglee, "A Dual Digital Audio Watermarking Algorithm Based on LWT," *Int. Conf. Meas. Inf. Control (MIC)*, 2012, no. Mic, pp. 721–725, 2012.
- [12] X. H. Huang, "A complex cepstrum fragile audio watermarking algorithm based on quantization," *3rd Int. Conf. Genet. Evol. Comput. WGECC 2009*, pp. 231–234, 2009.
- [13] H. Yassine, "A Secure and High Robust Audio Watermarking System for Copyright Protection," *24th Int. Conf. Microelectron. (ICM)*, 2012, vol. 53, no. 17, pp. 33–39, 2012.
- [14] R. Patel and J. D. Amin, "Alpha Channel base Data Hiding using Compressive Sensing," *Int. J. Adv. Res. Innov. Ideas Educ.*, no. 3, pp. 544–549, 2016.
- [15] H. H. I, J. A. Shah, and U. Ali, "Comparative Analysis of Sparse Signal Recovery Algorithms based on Minimization Norms," *World Congr. Sustain. Technol. (WCST)*, 2014, pp. 72–76, 2014.
- [16] C. T. Hsieh and P. Y. Tsou, "Blind cepstrum domain audio watermarking based on time energy features," *Int. Conf. Digit. Signal Process. DSP*, vol. 2, pp. 705–708, 2002.
- [17] Q. Li, J. Cox Ingemar, "Using Perceptual Models to Improve Fidelity and Provide Invariance to Volumetric Scaling For Quantization Index Modulation Watermarking," *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 2, pp. 127–139, 2007.
- [18] Q. Zhang, Y. Huang, J. Deng, and L. Zheng, "A Dither Modulation with Distortion Compensation Audio Watermarking Algorithm Based on DWT," *J. Comput. Inf. Syst.*, vol. 1, pp. 106–113, 2011.

