DASAR TEORI METODE QUANTUM LEAST SIGNIFICAN BITS

A. Dasar Teori

- ***Watermarking***

  Watermarking adalah proses menyembunyikan informasi untuk perlindungan konten atau verifikasi kepemilikan. Watermark ini berfungsi sebagai tanda pengenal atau pelacak yang tersembunyi dalam sinyal audio yang dapat digunakan untuk tujuan identifikasi keaslian, atau perlindungan hak cipta.

- **Audio *Watermarking***

  Audio *watermarking* mengacu pada teknik yang digunakan untuk menyisipkan informasi rahasia atau watermark dalam *file* audio. Tujuan utama audio *watermarking* adalah untuk memberikan keamanan dan perlindungan terhadap pelanggaran hak cipta, pemalsuan, atau penyebaran ilegal konten audio.

- ***Quantum Representation of Digital Audio Phase Coding***

  Quantum Representation of Digital Audio (QRDA) merupakan suatu proses penggunaan dua urutan qubit pada audio untuk menyimpan nilai fasa audio dan informasi waktu, sehingga seluruh audio digital disimpan dalam keadaan superposisi pada dua urutan qubit tersebut. Kedua urutan qubit ini awalnya berada dalam keadaan dasar $|0\rangle$ dan $|1\rangle$. Persamaan (3.3) menunjukkan proses QRDA phase encoding.

$$|I(\theta)\rangle = \frac{1}{2^n}\sum_{i=0}^{2^{2n-1}} |C_I\rangle|i\rangle \frac{1}{2^n}\sum_{i=0}^{2^{2n-1}} (cos\theta_i|0\rangle + sin\theta_i|1\rangle)|i\rangle$$

  $|0\rangle$ dan $|1\rangle$ adalah keadaan dasar komputasi kuantum dua dimensi, $|i\rangle$, i $= 1, 2, \cdots, 2^{2n}$ dan $\theta = [\theta_0, \theta_1, \cdots, \theta_{2^{2n-2}}, \theta_{2^{2n-1}}]$ adalah vektor sudut audio [14]. QRDA phase encoding akan menghasilkan sebanyak 1 qubit phase. Tahap awal dalam menggunakan phase encoding adalah mencari sudut teta pada audio. Nilai sudut teta kemudian akan dikalikan dengan cosinus dan sinus untuk membentuk fasa audio, dan dilakukan kronisasi untuk perkalian pada domain kuantum. Proses ini memungkinkan audio dalam representasi kuantum dengan memanfaatkan sifat-sifat cosinus dan sinus dalam menggambarkan informasi fasa secara efisien. Proses pengambilan kembali sinyal audio ke bentuk klasik dapat dilakukan dengan menggunakan Persamaan (3.4).

$$M_0 = |0\rangle \otimes \langle 0|$$
$$M_1 = |1\rangle \otimes \langle 1|$$
$$P(i) = \sqrt{\langle \hat{x}|M_j \otimes M_j \otimes M_j|\hat{x}\rangle}$$

$$at(i,1) = \frac{arcos(P(i) \text{ x } 2 \text{ x } (2 \text{ x } (2^{n-1}(i) \text{ x } 2 \text{ x } (2 \text{ x } (2^{n-1})}{\pi - 2^{n-1}}$$

Pengambilan sinyal audio klasik dilakukan dengan menghitung akar kuadrat dari perkalian $\langle \hat{x}|M_j \otimes M_j \otimes M_j|\hat{x}\rangle$. $\hat{x}$ adalah hasil nilai setelah dilakukan proses encoding dan M_j adalah posisi biner qubit yang akan dikronisasi. Setelah diperoleh posisi biner qubit, maka dilakukan perhitungan arcos untuk memperoleh sinyal audio klasik.

- **Postulat Pengukuran**

Postulat merupakan dasar untuk memahami bagaimana pengukuran dilakukan dalam mekanika kuantum. Pengukuran pada domain kuantum diinisialisasikan dengan bilangan M_{sn} yang merujuk pada *measurement*, *s* merujuk pada posisi *state* yang akan diukur, dan *n* sebagai nilai dari *qubit* yang akan diukur. Pengukuran pada domain kuantum merujuk pada probabilitas dari nilai yang diukur. Formula dari pengukuran postulat dapat dinyatakan dengan

$$P_{20} = \langle \psi | M_{20}^{\dagger} M_{20} | \psi \rangle$$

$$|\psi_{20}\rangle = \frac{M_{20}|\psi\rangle}{\sqrt{P_{20}}}$$

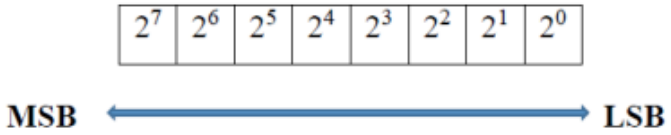
perhitungan pada bilangan P_{20} yang memiliki arti probabilitas *qubit* ke-2 bernilai 0. Setelah itu, hasil dari perhitungan probabilitas digunakan untuk menghitung nilai *qubit* ke-2. Apabila $P_{20} = 1$, maka $|\psi_{20}\rangle = |0\rangle$. Jika $P_{21} = 1$, maka $|\psi_{21}\rangle = |1\rangle$. Berikut adalah contoh perhitungan dari pengukuran postulat menggunakan 2 *qubit.*

Mengukur *qubit* ke-1		Mengukur *qubit* ke-2	
$M_{10} = M_0 \otimes I$		$M_{20} = I \otimes M_0$	
$M_{11} = M_1 \otimes I$		$M_{21} = I \otimes M_1$	
$P_{r10} = \langle \psi \| M_{10}^{\dagger} M_{10} \| \psi \rangle$		$P_{r20} = \langle SF_1 \| M_{20}^{\dagger} M_{20} \| SF_1 \rangle$	
$P_{r11} = \langle \psi \| M_{11}^{\dagger} M_{11} \| \psi \rangle$		$P_{r21} = \langle SF_1 \| M_{21}^{\dagger} M_{21} \| SF_1 \rangle$	
$If\ P_{r10} > P_{r11}$	$else\ P_rF_1 = P_{r11}$	$If\ P_{r20} > P_{r21}$	$else\ P_rF_2 = P_{r21}$
$P_rF_1 = P_{r10}$	$MF_1 = P_{r11}$	$P_rF_2 = P_{r20}$	$MF_2 = M_{21}$
$MF_1 = P_{r10}$	$m = 1$	$MF_2 = P_{r20}$	$n = 1$
$m = 0$		$MF_2 = M_{20}$	
		$n = 0$	

$$SF_1 = \frac{MF_1 \text{ x } |\psi\rangle}{\sqrt{P_rF_1}} \qquad SF_2 = \frac{MF_2 \text{ x } |SF_1\rangle}{\sqrt{P_rF_2}}$$

- ***Least Significant Bits***

  Metode kuantum LSB adalah proses watermarking yang menyisipkan informasi watermark ke dalam bit informasi terkecil atau paling tidak signifikan. Metode LSB adalah metode yang paling umum dan populer digunakan karena memiliki proses yang sederhana. menunjukkan ilustrasi letak bit LSB.



  Langkah-langkah watermarking menggunakan metode kuantum LSB seperti berikut:
  File audio host diubah menjadi bentuk biner, misalkan bentuk audio biner adalah [11001101 01110101 10110110].
  Bit watermark yang telah dikonversi ke bentuk biner adalah [1 0 1].
  Setiap bit akhir pada host akan digantikan oleh bit watermark akan menghasilkan bit [11001101 01110100 10110111]. Ilustrasi perubahan bit pada tabel ini

Host audio	1100110**1**	0111010**1**	1011011**0**
Watermark	**1**	**0**	**1**
Watermark *audio*	1100110**1**	0111010**0**	1011011**1**

- **Serangan Quantum**

  Proses serangan *gate* kuantum dilakukan untuk pengujian keamanan jika terjadinya serangan *noise* terhadap audio yang telah terwatermark, serangan ini dilakukan dengan tiga jenis serangan kuantum. Pada serangan jenis kuantum Pauli-X berfungsi untuk *bitflip* yaitu membalikan nilai *qubit* state |0⟩ dan 1⟩. Representasi ukuran matriks .

$$x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

  Serangan noise Pauli-Z juga dilakukan pengujian pada tugas akhir ini. Pauli-Z berfungsi sebagai *phase-flip operator*. Jika memiliki *quantum state* |0⟩ dan 1⟩, pauli Z akan membalikan *qubit phase* pada saat *state* 1⟩. *Noise* ini berbentuk bit yang direpresentasikan pada ukuran matriks.

$$z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

  Jenis serangan CNOT juga dilakukan pengujian pada tugas akhir ini. Serangan Pauli-CNOT terdiri dari 2 *qubit* yaitu kontrol *qubit* dan target *qubit.*

Gerbang CNOT mengubah keadaan *qubit* target, jika *qubit* kontrol berada dalam keadaan |1⟩. Noise ini berbentuk bit yang direpresentasikan pada bentuk matriks.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- **Perhitungan BER dan SNR**

  Perhitungan SNR dilakukan untuk melihat perbandingan antara sinyal asli dan sinyal yang telah diwatermark, setelah dilakukannya proses *embedding* dan konversi ke klasik dapat dilakukannya perhitungan SNR untuk menilai sistem yang digunakan dengan (3.12).

$$SNR = 10log\left(\frac{P_s}{P_N}\right)$$

  P_s adalah daya efektif sinyal dan P_N adalah daya efektif sinyal dan P_N adalah daya noise yang diterima, adapun parameter yang diuji pada penelitian ini adalah nilai BER seperti pada (3.13).

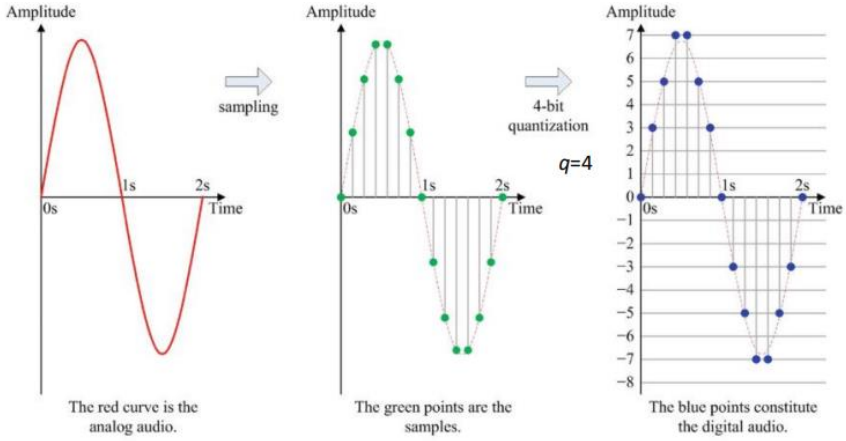
$$BER = \frac{N_e}{T_N}\ x\ 100\%$$

  N_e didefinisikan sebagai jumlah bit yang eror pada watermark, dan T_n didefinisikan sebagai jumlah total bit.

B. Tahapan Proses *Quantum Watermarking*

- **Normalisasi Audio**

  Langkah awal yang dilakukan dalam proses watermarking kuantum adalah normalisasi audio. Dalam tahap ini, audio disesuaikan agar sesuai dengan panjang bit kuantum yang telah ditentukan sebelumnya. Sebagai ilustrasi, jika panjang bit kuantum yang digunakan adalah 8, maka seluruh nilai audio akan dinormalisasi sedemikian rupa sehingga rentang nilai maksimalnya adalah 7 dan nilai minimalnya adalah -7.



- **Normalisasi Citra Watermark**

  Proses watermarking melibatkan penyisipan citra watermark ke dalam audio. Citra watermark akan diubah menjadi skala warna hitam putih dan kemudian diubah ukurannya agar sesuai dengan dimensi audio. Untuk dapat menyisipkan bit-bit citra watermark ke dalam audio, citra tersebut harus mengalami proses konversi menjadi format biner satu dimensi.

- **Embedding LSB**

  Embedding adalah proses penyisipan bit watermark pada bit paling tidak signifikan atau bit terakhir pada bit audio. Contoh Proses embedding sebagai berikut

  1. Untuk memulai proses watermarking, kita perlu menentukan jumlah bit kuantum dan bit waktu yang akan digunakan. Jumlah bit kuantum akan diinisialisasi dalam variabel N, sementara jumlah bit waktu akan diinisialisasi dalam variabel Ns. Adapun panjang segmentasi yang didapatkan dari 2^Ns Pada contoh ini, kita akan menggunakan N=8 dan Ns=2. Selain itu, dalam proses watermarking, dibutuhkan sebuah kunci yang menggunakan matriks Hadamard untuk tahap ekstraksi. Kunci ini akan diinisialisasi dalam variabel key.

$$N = 8$$
$$\mathrm{Ns} = 2$$
$$\mathrm{Ls} = 2^2 = 4$$
$$\mathrm{H} = \mathrm{Hadamard(Ls)}$$
$$\mathrm{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

  Dalam metode LSB, kunci (key) harus berbentuk biner dengan nilai antara 0 dan 1. Proses penanaman (embedding) kunci ke dalam watermark diinisialisasi dengan variabel C, seperti berikut

$$\mathrm{Key} = 2$$
$$\mathrm{C} = \frac{1}{2} * \mathrm{H}(2) + \frac{1}{2}$$

H(2) adalah pengambilan matriks baris 2 pada matriks Hadamard, sehingga nilai matriks C akan menghasilkan

$$C = [1 \quad 0 \quad 1 \quad 0]$$

2. Tahap selanjutnya adalah proses penyisipan watermark pada audio dengan dilakukan proses xor nilai bit terakhir pada watermark dan C

$$W = \text{xor}(\text{watermark}, c)$$

Watermark	C	W
1	1	0
1	0	1
1	1	0
1	0	1

Setelah mendapatkan nilai W, audio akan dipecah menjadi matriks dengan nilai 4 bit pada setiap segmennya. Sebagai contoh, untuk metode LSB, nilai dari segmen terakhir digunakan, kemudian nilai tersebut akan ditambahkan dengan 2^N-1 dan diubah menjadi representasi biner. Nilai segmen kemudian menjadi:

$$Xs = [119 \quad 122 \quad 126 \quad 132] + 255$$

$$Xs = [374 \quad 377 \quad 381 \quad 387]$$

Diubah menjadi bentuk biner

374= 101110110
377= 101111001
381= 101111101
387= 110000011

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Lalu akan dilakukan proses penyisipan pada bit paling tidak signifikan yaitu pada bit paling kanan dengan matriks C.

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Setelah mendapatkan hasil biner yang telah disisipkan akan di ubah menjadi bentuk desimal kembali untuk melakukan proses encoding.

$$Xw = [374 \quad 377 \quad 380 \quad 387] - 255$$

$$Xw = [119 \quad 122 \quad 125 \quad 132]$$

Perbedaan nilai segmentasi setelah dilakukan penyisipan

Xs [119 122 126 132]

Xw [119 122 125 132]

Nilai segmentasi mengalami perubahan pada nilai 126 menjadi 125 dikarenakan proses penyisipan.

- **Phase encoding**

A= [119 122 125 132]

00 01 10 11

$$|x\rangle = \left(\cos\frac{119}{132}\frac{\pi}{2}|0\rangle + \sin\frac{119}{132}\frac{\pi}{2}|1\rangle|00\rangle\right) +$$
$$\left(\cos\frac{122}{132}\frac{\pi}{2}|0\rangle + \sin\frac{122}{132}\frac{\pi}{2}|1\rangle|01\rangle\right) +$$
$$\left(\cos\frac{125}{132}\frac{\pi}{2}|0\rangle + \sin\frac{125}{132}\frac{\pi}{2}|1\rangle|10\rangle\right) +$$
$$\left(\cos\frac{132}{132}\frac{\pi}{2}|0\rangle + \sin\frac{132}{132}\frac{\pi}{2}|1\rangle|11\rangle\right) +$$

Disederhanakan:

$$\left(\cos\frac{119}{132}\frac{\pi}{2}|000\rangle + \sin\frac{119}{132}\frac{\pi}{2}|100\rangle\right) +$$
$$\left(\cos\frac{122}{132}\frac{\pi}{2}|001\rangle + \sin\frac{122}{132}\frac{\pi}{2}|101\rangle\right) +$$
$$\left(\cos\frac{125}{132}\frac{\pi}{2}|010\rangle + \sin\frac{125}{132}\frac{\pi}{2}|110\rangle\right) +$$
$$\left(\cos\frac{132}{132}\frac{\pi}{2}|011\rangle + \sin\frac{119}{132}\frac{\pi}{2}|111\rangle\right) +$$

Hasil akan diurutkan berdasarkan posisi dari 000 sampai 111.

$$|x\rangle = \begin{bmatrix} \cos\frac{119}{132}\frac{\pi}{2} \\ \cos\frac{122}{132}\frac{\pi}{2} \\ \cos\frac{125}{132}\frac{\pi}{2} \\ \cos\frac{132}{132}\frac{\pi}{2} \\ \sin\frac{119}{132}\frac{\pi}{2} \\ \sin\frac{122}{132}\frac{\pi}{2} \\ \sin\frac{125}{132}\frac{\pi}{2} \\ \sin\frac{132}{132}\frac{\pi}{2} \end{bmatrix} \qquad |x\rangle = \begin{bmatrix} 0{,}4067 \\ 0{,}3983 \\ 0{,}3898 \\ 0{,}3698 \\ 0{,}9135 \\ 0{,}9173 \\ 0{,}9209 \\ 0{,}9291 \end{bmatrix}$$

- **Phase Decoding**

$$|x\rangle = \begin{bmatrix} 0{,}4067 \\ 0{,}3983 \\ 0{,}3898 \\ 0{,}3698 \\ 0{,}9135 \\ 0{,}9173 \\ 0{,}9209 \\ 0{,}9291 \end{bmatrix}$$

X= ?

X= [a b c d]

A b c d adalah nilai $|x\rangle$ dari perhitungan cos, untuk menentukan mengambalikan keadaan semula maka dilakukan arcos,

$$a = \text{acos } 0{,}4067 \frac{132*2}{\pi} \qquad c = \text{acos } 0{,}3898 \frac{132*2}{\pi}$$

$$b = \text{acos } 0{,}3983 \frac{132*2}{\pi} \qquad d = \text{acos } 0{,}3698 \frac{132*2}{\pi}$$

Perhitungan menghasilkan nilai yang sama sebelum dan sesudah proses phase coding.

A = 119 , B= 122, C= 125, D=132

- **Ekstraksi**

Audio berbentuk segmentasi akan dipecah kembali untuk melakukan proses ekstraksi. Sebagai contoh

Nilai hasil decoding = $[119 \quad 122 \quad 125 \quad 132]$ (Masih tersegmentasi)

Disatukan = 123 …. 119 122 125 132 (rentang panjang audio)

119 122 125 132 akan dilakukan pertambahan dengan 2^n+1 (255) hasil nilai akan menjadi 374 377 380 387 dan akan diproses perubahan kedalam bentuk biner untuk proses ekstraksi

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Diambil pada urutan bit yang paling kanan untuk dilakukan proses penyamaan dengan C dan dilakukan xor.

Hostw (end)	C	Xor
0	1	1
1	0	1
0	1	1
1	0	1

Setelah dilakukan mendapatkan hasil xor akan dilakukan proses rata rata xor bernilai 1. Rata-rata 1 dapat menunjukkan bahwa jumlah 1 dalam data hampir sama banyaknya dengan jumlah 0.