



**TUGAS AKHIR - KI1502**

**IMPLEMENTASI METODE *CHAOTIC ENCRYPTION* DAN  
*DISCRETE FOURIER TRANSFORM* UNTUK  
MENYEMBUNYIKAN DATA BIOMETRIK SIDIK JARI KE  
DALAM SINYAL AUDIO**

**EBENHAEZER WIRAMARTA YOGASWARA  
NRP 5111100074**

**Dosen Pembimbing I  
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.**

**Dosen Pembimbing II  
Tohari Ahmad, S.Kom., MIT., PhD**

**JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**



UNDERGRADUATE THESES - KI1502

# IMPLEMENTATION OF CHAOTIC ENCRYPTION AND DISCRETE FOURIER TRANSFORM METHOD FOR CONCEALING FINGERPRINT-BIOMETRIC DATA INTO AUDIO SIGNALS

EBENHAEZER WIRAMARTA YOGASWARA  
NRP 5111100074

Supervisor I  
Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.

Supervisor II  
Tohari Ahmad, S.Kom., MIT., PhD

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2015

## LEMBAR PENGESAHAN

### IMPLEMENTASI METODE *CHAOTIC ENCRYPTION* DAN *DISCRETE FOURIER TRANSFORM* UNTUK MENYEMBUNYIKAN DATA BIOMETRIK SIDIK JARI KE DALAM SINYAL AUDIO

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Cerdas Visual  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh  
**EBENHAEZER WIRAMARTA YOGASWARA**  
NRP : 5111 100 074

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Dr.Eng. Chastine Fatchah, S.Kom., M.Kom. ....  
NIP: 197512202001122002 (Pembimbing 1)
2. Tohari Ahmad, S.Kom., MIT, PhD .....  
NIP: 197505252003121002 (Pembimbing 2)

**SURABAYA**  
**JULI, 2015**

# **IMPLEMENTASI METODE *CHAOTIC ENCRYPTION* DAN *DISCRETE FOURIER TRANSFORM* UNTUK MENYEMBUNYIKAN DATA BIOMETRIK SIDIK JARI KE DALAM SINYAL AUDIO**

**Nama Mahasiswa** : EBENHAEZER WIRAMARTA  
YOGASWARA  
**NRP** : 5111100074  
**Jurusan** : Teknik Informatika FTIF-ITS  
**Dosen Pembimbing 1** : Dr.Eng. Chastine Faticah, S.Kom.,  
M.Kom.  
**Dosen Pembimbing 2** : Tohari Ahmad, S.Kom., MIT.,PhD.

## ***Abstrak***

*Keamanan adalah salah satu isu terpenting pada era ini, terutama untuk keamanan data biometrik sidik jari. Pada umumnya, sistem yang menggunakan data biometrik sidik jari menyimpan atau mengirim data tersebut dalam bentuk gambar. Sistem penyimpanan atau pengiriman data sidik jari seperti ini rentan terhadap serangan dari pihak-pihak seperti administrator basis data yang memiliki akses ke dalam basis data, serta peretas yang dapat menginterupsi pengiriman data. Maka dari itu, dibutuhkan suatu metode penyembunyian data biometrik sidik jari ke dalam bentuk yang lebih aman agar data sidik jari dapat dikirim melalui jaringan yang tidak aman.*

*Pada Tugas Akhir ini dibentuk solusi menggunakan metode Chaotic Encryption dan Discrete Transform Fourier (DFT) untuk menyembunyikan data biometrik sidik jari ke dalam sinyal audio. Metode penyembunyian sidik jari ke dalam sinyal audio ini memiliki 4 tahapan, yakni enkripsi data sidik jari dengan Chaotic Encryption, transformasi sinyal audio ke dalam domain frekuensi dengan DFT, penyisipan*

*data terenkripsi ke dalam domain frekuensi hasil DFT dengan kuantisasi, dan pengembalian sampel frekuensi hasil kuantisasi ke dalam sinyal audio dalam domain waktu dengan Inverse Discrete Fourier Transform (IDFT). Metode ekstraksi gambar dari sidik jari terdiri dari 3 tahapan yang merupakan komplemen dari proses-proses penyembunyian. Metode Chaotic Encryption yang digunakan adalah Skew Tent Map Encryption yang terbukti lebih unggul dibandingkan metode Chaotic Encryption yang lain. Tugas akhir ini juga mencari perbandingan metode Discrete Fourier Transform antara Walsh-Hadamard Transform (WHT) dan Discrete Cosine Transform (DCT). Proses kuantisasi adalah perubahan sampel frekuensi ke bentuk normalisasinya sesuai data yang ingin disisipkan. Di dalamnya terdapat variabel penting yang mempengaruhi kinerja aplikasi ini yang disebut quantization step ( $\Delta$ ), dimana variabel ini akan menjadi parameter yang diuji cobakan pada Tugas Akhir ini.*

*Hasil uji coba menunjukkan bahwa kedua metode transformasi fourier dapat menghasilkan sinyal audio yang tidak terdengar perbedaannya bila dibandingkan dengan audio yang asli, dapat merekonstruksi gambar sidik jari dengan akurasi sebesar 100%, dan memiliki ketahanan yang sama terhadap Additive White Gaussian Noise dengan varians sebesar 484, apabila menggunakan nilai quantization step yang optimal yang bernilai  $61 \times 10^{-4}$  untuk metode Discrete Cosine Transform dan  $31 \times 10^{-6}$  untuk metode Walsh-Hadamard Transform. Perbedaan yang signifikan terletak pada running time, dimana running time DCT adalah 0,04 detik, sedangkan running time WHT adalah 1,77 detik.*

***Kata kunci: Chaotic Encryption, Discrete Cosine Transform, Discrete Fourier Transform, Kuantisasi, Skew Tent Map Encryption, Walsh-Hadamard Transform.***

# IMPLEMENTATION OF CHAOTIC ENCRYPTION AND DISCRETE FOURIER TRANSFORM METHOD FOR CONCEALING FINGERPRINT-BIOMETRIC DATA INTO AUDIO SIGNALS

**Student's Name** : EBENHAEZER WIRAMARTA  
YOGASWARA  
**Student's ID** : 5111100074  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Dr.Eng. Chastine Fatichah, S.Kom.,  
M.Kom.  
**Second Advisor** : Tohari Ahmad, S.Kom., MIT.,PhD.

## *Abstract*

*Nowadays, security is one of the most important issues, especially the security of fingerprint biometric data. In general, a system which uses fingerprint data stores and transmits in the form of images. This data storage or transmission system is vulnerable to attack by any party such as database administrator who has an access to the database, as well as hackers who is able to interrupt the transmission of fingerprint data. Therefore, there is a need to hide fingerprint data to a safer form so that it can be sent over nonsecure networks.*

*This final project develops solution by using the Chaotic Encryption and the Discrete Fourier Transform (DFT) method to hide a fingerprint data into audio signals. This method consists of 4 steps. First, the fingerprint data is encrypted by Chaotic Encryption. Second, the transformation of the audio signal into a frequency domain by DFT. Third, the insertion of data into the frequency domain by quantization. And fourth, the reconstruction of audio signal by Inverse*

*Discrete Fourier Transform (IDFT). Fingerprint extraction method consists of 3 steps which are the complement of the data hiding method. The Chaotic Encryption method which is used in this final project is Skew Tent Map Encryption which shows superiority against other Chaotic Encryption method. This final project also tries to compare a better Discrete Fourier Transform method between Walsh-Hadamard Transform (WHT) and Discrete Cosine Transform (DCT). The quantization process modifies a frequency sample to a normalized form frequency sample using the corresponding data that needs hiding. This process also uses a certain variable which affects the performance of this application, called the quantization step ( $\Delta$ ).*

*The experimental results shows that an optimal quantization value for WHT is  $61 \times 10^{-6}$  and for DCT is  $31 \times 10^{-4}$ . The experimental results also shows a comparison between those fourier transform method using four different parameters. First, the difference between the audio-stego signal and the original audio signal must not be recognized by human auditory system. Second, the extraction method must be able to construct the fingerprint data with an accuracy of 100%. Third, the extraction method must be able to construct the fingerprint data with a minimum accuracy of 99,41% after the audio-stego signal is added by an Additive White Gaussian Noise with a variance of 484. And fourth, the running time of both fourier transform methods. Both fourier transform methods are able to satisfy the first three parameters with no significant result. However on the fourth parameter, the DFT shows a better and significant performance compared to the WHT, whereupon the running time of DFT is 0,04 seconds and the running time of WHT is 1,77 seconds.*

***Keywords: Chaotic Encryption, Discrete Cosine Transform, Discrete Fourier Transform, Quantization, Skew Tent Map Encryption, Walsh-Hadamard Transform.***

## KATA PENGANTAR

Segala puji dan syukur kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul **“IMPLEMENTASI METODE *CHAOTIC ENCRYPTION* DAN *DISCRETE FOURIER TRANSFORM* UNTUK MENYEMBUNYIKAN DATA BIOMETRIK SIDIK JARI KE DALAM SINYAL AUDIO”**.

Tugas Akhir ini diajukan untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember.

Dengan selesainya Tugas Akhir ini, penulis berharap apa yang penulis telah kerjakan dapat memberikan manfaat bagi perkembangan ilmu pengetahuan serta bagi penulis.

Keberhasilan dalam penyelesaian Tugas Akhir ini tidak lepas dari bantuan berbagai pihak. Maka dari itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang tulus dan sebesar-besarnya kepada semua pihak, terutama kepada:

1. Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik.
2. Keluarga penulis (Brotosari Rahayu, SpB, FINACS) yang senantiasa memberikan doa dan dukungan baik secara moral dan finansial selama penulis menyelesaikan Tugas Akhir.
3. Bapak Dwi Sunaryono, S.Kom., M.Kom. selaku dosen wali penulis, yang selalu memberikan bimbingan, dukungan, serta motivasi selama masa perkuliahan di Teknik Informatika ITS.
4. Ibu Dr. Chastine Fatichah, S.Kom., M.Kom. selaku dosen pembimbing I atas bimbingan, ilmu, dan kesabaran selama proses pengerjaan tugas akhir.



5. Bapak Tohari Ahmad, S.Kom., MIT., PhD. selaku dosen pembimbing II atas bimbingan, ilmu, dan kesabaran selama proses pengerjaan tugas akhir.
6. Bapak dan Ibu dosen dan karyawan jurusan Teknik Informatika ITS yang telah banyak mengajarkan ilmu pada penulis selama menempuh studi di Teknik Informatika ITS.
7. Teman-teman penulis, khususnya teman-teman angkatan 2011 dalam menempuh perkuliahan di Jurusan Teknik Informatika Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember.
8. Dan seluruh pihak yang turut membantu dalam pengerjaan tugas akhir ini, baik secara moril maupun materiil, yang tidak dapat penulis sebutkan satu per satu.

Penulis berharap bahwa tugas akhir ini dapat memberikan manfaat pada semua pihak khususnya penulis sendiri dan *civitas academica* Teknik Informatika ITS. Penulis mohon maaf bila terdapat kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan ke depan.

Surabaya, Juni 2015  
Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak</i> .....	vii
<i>Abstract</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL.....	xx
DAFTAR KODE SUMBER .....	xxiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Metodologi .....	4
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Biometrik.....	7
2.1.1 Biometrik Sidik Jari.....	8
2.2 Sinyal Audio.....	8
2.3 <i>Chaotic Encryption</i> .....	10
2.3.1 <i>Skew Tent Map</i> .....	11
2.4 <i>Discrete Fourier Transform</i> .....	12
2.4.1 <i>Walsh-Hadamard Transform</i> .....	14
2.4.2 <i>Discrete Cosine Transform</i> .....	16
2.5 Kuantisasi .....	18
2.5.1 Re-kuantisasi .....	20
2.6 <i>Signal-To-Noise Ratio</i> .....	21
2.7 <i>Additive White Gaussian Noise</i> .....	22
2.8 <i>Matching</i> .....	23
BAB III DESAIN PERANGKAT LUNAK.....	25

3.1	Desain Metode Secara Umum .....	25
3.2	Penentuan <i>Key</i> .....	26
3.3	Penyembunyian Sidik Jari .....	26
3.3.1	<i>Skew Tent Map Encryption</i> .....	28
3.3.2	<i>Walsh-Hadamard Transform</i> .....	28
3.3.3	<i>Discrete Cosine Transform</i> .....	29
3.3.4	Kuantisasi .....	30
3.3.5	<i>Inverse WHT</i> .....	31
3.3.6	<i>Inverse DCT</i> .....	32
3.3.7	Normalisasi Audio .....	33
3.3.8	SNR .....	33
3.4	Ekstraksi Sidik Jari .....	34
3.4.1	<i>Additive White Gaussian Noise</i> .....	34
3.4.2	Re-kuantisasi .....	35
3.4.3	<i>Skew Tent Map Decryption</i> .....	36
3.4.4	<i>Matching</i> .....	36
BAB 4 BAB IV IMPLEMENTASI .....		37
4.1	Lingkungan Implementasi .....	37
4.2	Implementasi .....	37
4.2.1	Implementasi Penentuan <i>Key</i> .....	37
4.2.2	Implementasi Fungsi <i>Skew Tent Map Encryption</i> .....	38
4.2.3	Implementasi Fungsi <i>Walsh-Hadamard Transform</i> .....	39
4.2.4	Implementasi Fungsi <i>Discrete Cosine Transform</i> .....	40
4.2.5	Implementasi Fungsi Kuantisasi .....	40
4.2.6	Implementasi Fungsi <i>Inverse WHT</i> .....	42
4.2.7	Implementasi Fungsi <i>Inverse DCT</i> .....	42
4.2.8	Implementasi Fungsi Normalisasi dan Keluaran Audio .....	43
4.2.9	Implementasi Pengujian SNR .....	43
4.2.10	Implementasi <i>Additive White Gaussian Noise</i> .....	43
4.2.11	Implementasi Fungsi Re-kuantisasi .....	44
4.2.12	Implementasi Fungsi <i>Skew Tent Map Decryption</i> .....	45
4.2.13	Implementasi Fungsi <i>Matching</i> .....	45
BAB V UJI COBA DAN EVALUASI .....		47
5.1	Lingkungan Uji Coba .....	47
5.2	Data Uji Coba .....	47

5.3 Uji Coba Tiap Proses.....	49
5.3.1 Uji Coba Enkripsi dan Dekripsi .....	49
5.3.2 Uji Coba Transformasi Fourier .....	50
5.3.3 Uji Coba Kuantisasi, Re-kuantisasi, dan <i>Inverse</i> Transformasi Fourier .....	51
5.4 Uji Coba Secara Keseluruhan.....	56
5.4.1 Uji Coba dengan WHT dan IWHT.....	57
5.4.2 Uji Coba dengan DCT dan IDCT .....	60
5.5 Skenario dan Evaluasi Pengujian .....	63
5.5.1 Skenario Uji Coba 1 .....	64
5.5.2 Skenario Uji Coba 2 .....	67
5.5.3 Skenario Uji Coba 3 .....	69
5.6 Analisis Hasil Uji Coba.....	73
BAB 6 BAB VI KESIMPULAN DAN SARAN .....	77
6.1 Kesimpulan.....	77
6.2 Saran.....	77
DAFTAR PUSTAKA .....	79
LAMPIRAN.....	81
BIODATA PENULIS .....	105

## DAFTAR TABEL

Tabel 5. 1 Hasil Skenario Uji Coba 3 untuk Metode DCT dengan $\Delta=61 \times 10^{-4}$ ( $\Delta$ terbesar).....	71
Tabel 5. 2 Hasil Skenario Uji Coba 3 untuk Metode WHT dengan $\Delta=31 \times 10^{-6}$ ( $\Delta$ terbesar).....	71
Tabel 5. 3 Hasil Skenario Uji Coba 3 untuk Metode DCT dengan $\Delta=1 \times 10^{-4}$ ( $\Delta$ terkecil) .....	72
Tabel 5. 4 Hasil Skenario Uji Coba 3 untuk Metode WHT dengan $\Delta=1 \times 10^{-6}$ ( $\Delta$ terkecil) .....	72
Tabel 5. 5 Tabel Perbandingan WHT dan DCT .....	75
Tabel A. 1 Tabel Detil Hasil Uji Coba 1(1) .....	81
Tabel A. 2 Tabel Detil Hasil Uji Coba 1(2) .....	82
Tabel A. 3 Tabel Detil Hasil Uji Coba 1(3) .....	83
Tabel A. 4 Tabel Detil Hasil Uji Coba 1(4) .....	84
Tabel A. 5 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-6}$ (1).....	86
Tabel A. 6 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-6}$ (2).....	87
Tabel A. 7 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-6}$ (3).....	88
Tabel A. 8 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-6}$ (4).....	89
Tabel A. 9 Tabel Detil Hasil Uji Coba 2(1) .....	90
Tabel A. 10 Tabel Detil Hasil Uji Coba 2(2) .....	91
Tabel A. 11 Tabel Detil Hasil Uji Coba 2(3) .....	92
Tabel A. 12 Tabel Detil Hasil Uji Coba 2(4) .....	93
Tabel A. 13 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-4}$ (1) .....	95
Tabel A. 14 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-4}$ (2) .....	96
Tabel A. 15 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-4}$ (3) .....	97

Tabel A. 16 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-4}$ (4).....	98
Tabel A. 17 Tabel Hasil Uji Coba Metode WHT pada Nilai $\Delta$ di sekitar $31 \times 10^{-6}$ (1).....	99
Tabel A. 18 Tabel Hasil Uji Coba Metode WHT pada Nilai $\Delta$ di sekitar $31 \times 10^{-6}$ (2).....	100
Tabel A. 19 Tabel Hasil Uji Coba Metode WHT pada Nilai $\Delta$ di sekitar $31 \times 10^{-6}$ (3).....	101
Tabel A. 20 Rekap Hasil Uji Coba Metode WHT pada Tabel A. 17 hingga Tabel A. 19 .....	101
Tabel A. 21 Tabel Hasil Uji Coba Metode DCT pada Nilai $\Delta$ di sekitar $61 \times 10^{-4}$ (1).....	102
Tabel A. 22 Tabel Hasil Uji Coba Metode DCT pada Nilai $\Delta$ di sekitar $61 \times 10^{-4}$ (2).....	103
Tabel A. 23 Tabel Hasil Uji Coba Metode DCT pada Nilai $\Delta$ di sekitar $61 \times 10^{-4}$ (3).....	104
Tabel A. 24 Rekap Hasil Uji Coba Metode DCT pada Tabel A. 21 hingga Tabel A. 23 .....	104

## DAFTAR GAMBAR

Gambar 2. 1 Contoh Biometrik Sidik Jari.....	9
Gambar 2. 2 Contoh visualisasi gelombang suara .....	9
Gambar 2. 3 Contoh Binary Skew Tent Map.....	12
Gambar 2. 4 Matriks Hadamard 2x2 ( $H_2$ ) .....	15
Gambar 2. 5 Matriks Hadamard 4x4 ( $H_4$ ) .....	15
Gambar 2. 6 Matriks Hadamard 8x8 ( $H_8$ ) .....	16
Gambar 2. 7 Ilustrasi Pembagian Sampel Frekuensi dengan Quantization Step ( $\Delta$ ).....	18
Gambar 2. 8 Ilustrasi sinyal yang diberi AWGN .....	23
Gambar 3. 1 Diagram Alur Aplikasi .....	27
Gambar 3. 2 Pseudocode Skew Tent Map Encryption.....	28
Gambar 3. 3 Pseudocode WHT .....	29
Gambar 3. 4 Pseudocode DCT .....	29
Gambar 3. 5 Pseudocode Kuantisasi .....	31
Gambar 3. 6 Pseudocode IWHT .....	32
Gambar 3. 7 Pseudocode IDCT.....	33
Gambar 3. 8 Pseudocode Pengujian SNR .....	34
Gambar 3. 9 Pseudocode Additive White Gaussian Noise .....	35
Gambar 3. 10 Pseudocode Re-kuantisasi .....	35
Gambar 3. 11 Pseudocode Dekripsi .....	36
Gambar 3. 12 Pseudocode Matching.....	36
Gambar 5. 1 Gambar Sidik Jari Individu ke-10 .....	48
Gambar 5. 2 Data Audio .....	49
Gambar 5. 3 Gambar asli(kiri), gambar hasil enkripsi(tengah), dan gambar hasil dekripsi(kanan) .....	50
Gambar 5. 4 Gambar sinyal audio(atas) dan gambar sinyal frekuensi hasil WHT(bawah) .....	51
Gambar 5. 5 Gambar sinyal audio(atas) dan gambar sampel frekuensi hasil DCT(bawah) .....	52
Gambar 5. 6 Gambar frekuensi hasil WHT(atas), hasil kuantisasinya(tengah), dan selisihnya (bawah) .....	53

Gambar 5. 7 Gambar frekuensi hasil DCT(atas), hasil kuantisasinya(tengah), dan selisihnya (bawah) .....	53
Gambar 5. 8 Gambar sampel frekuensi WHT(atas), hasil re-kuantisasinya(kiri bawah), dan hasil dekripsinya(kanan bawah) .....	54
Gambar 5. 9 Gambar sampel frekuensi DCT(atas), hasil re-kuantisasinya(kiri bawah), dan hasil dekripsinya(kanan bawah) .....	54
Gambar 5. 10 Gambar sampel frekuensi WHT yang telah dikuantisasi(atas) dan gambar sinyal audio hasil IWHT(bawah).....	55
Gambar 5. 11 Gambar sampel frekuensi DCT yang telah dikuantisasi(atas) dan gambar sinyal audio hasil IDCT(bawah) .....	56
Gambar 5. 12 Gambar sidik jari(kiri) dan enkripsinya(kanan) .....	57
Gambar 5. 13 Gambar hasil kuantisasi sampel frekuensi WHT .....	58
Gambar 5. 14 Gambar sinyal audio hasil IWHT dari sampel frekuensi yang telah dikuantisasi.....	58
Gambar 5. 15 Gambar frekuensi WHT dari sinyal audio-stego .....	59
Gambar 5. 16 Gambar hasil Re-kuantisasi dari sampel frekuensi pada Gambar 5. 15 .....	59
Gambar 5. 17 Gambar hasil dekripsi dari Gambar 5. 16 .....	60
Gambar 5. 18 Gambar hasil kuantisasi sampel frekuensi DCT .....	61
Gambar 5. 19 Gambar sinyal audio hasil IDCT dari sampel frekuensi yang telah dikuantisasi.....	61
Gambar 5. 20 Gambar frekuensi DCT dari sinyal audio-stego .....	62
Gambar 5. 21 Gambar hasil Re-kuantisasi dari sampel frekuensi pada Gambar 5. 20 .....	62
Gambar 5. 22 Gambar hasil dekripsi dari Gambar 5. 21 .....	63



Gambar 5. 23 Grafik Rata-Rata Akurasi Gambar pada Uji Coba 1 (detil pada Tabel A. 1 hingga Tabel A. 4) .....	65
Gambar 5. 24 Grafik Rata-Rata SNR Audio pada Uji Coba 1 (detil pada Tabel A. 1 hingga Tabel A. 4).....	65
Gambar 5. 25 Grafik Rata-Rata Akurasi Gambar pada Uji Coba 2 (detil pada Tabel A. 9 hingga Tabel A. 12) .....	67
Gambar 5. 26 Grafik Rata-Rata SNR Audio pada Uji Coba 2 (detil pada Tabel A. 9 hingga Tabel A. 12).....	68

Gambar A. 1 Grafik Rata-Rata Akurasi Gambar pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-6}$ pada Uji Coba 1 (detil pada Tabel A. 5 hingga Tabel A. 8).....	85
Gambar A. 2 Grafik Rata-Rata SNR pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-6}$ pada Uji Coba 1 (detil pada Tabel A. 5 hingga Tabel A. 8) .....	85
Gambar A. 3 Grafik Rata-Rata Akurasi Gambar pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-4}$ pada Uji Coba 2 (detil pada Tabel A. 13 hingga Tabel A. 16).....	94
Gambar A. 4 Grafik Rata-Rata SNR pada Rentang Nilai $\Delta = 0$ hingga $1 \times 10^{-4}$ pada Uji Coba 2 (detil pada Tabel A. 13 hingga Tabel A. 16) .....	94

## DAFTAR KODE SUMBER

Kode Sumber 4. 1 Implementasi Penentuan Key.....	37
Kode Sumber 4. 2 Implementasi Pengambilan Sampel Gambar dan Fungsi Skew Tent Map.....	38
Kode Sumber 4. 3 Implementasi Detil Fungsi Skew Tent Map .....	39
Kode Sumber 4. 4 Implementasi Pengambilan Sampel Suara dan Fungsi WHT .....	39
Kode Sumber 4. 5 Implementasi Pengambilan Sampel Suara dan Fungsi DCT .....	40
Kode Sumber 4. 6 Implementasi Fungsi Kuantisasi(1).....	41
Kode Sumber 4. 7 Implementasi Fungsi Kuantisasi(2).....	42
Kode Sumber 4. 8 Implementasi Fungsi IWHT .....	42
Kode Sumber 4. 9 Implementasi Fungsi IDCT .....	42
Kode Sumber 4. 10 Implementasi Fungsi Keluaran Audio....	43
Kode Sumber 4. 11 Implementasi Pengujian SNR .....	43
Kode Sumber 4. 12 Implementasi Additive White Gaussian Noise .....	44
Kode Sumber 4. 13 Implementasi Fungsi Re-kuantisasi .....	45
Kode Sumber 4. 14 Implementasi Fungsi Dekripsi .....	45
Kode Sumber 4. 15 Implementasi Fungsi Matching.....	46

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

[1]Pesatnya perkembangan teknologi pada zaman ini sering kali mengangkat isu keamanan dan privasi dalam sistem autentikasi dan identifikasi. E-banking, e-commerce, m-commerce, dan ATM adalah beberapa contoh aplikasi yang krusial dalam dunia bisnis online, serta membutuhkan keamanan data yang tinggi, baik data tersebut disimpan dalam *database* atau jaringan. Salah satu teknik untuk mengidentifikasi seseorang yang paling teruji dan digunakan secara global adalah identifikasi menggunakan biometrik sidik jari.

Pada umumnya, peredaran data biometrik sidik jari dalam suatu sistem masih berupa data gambar yang “mentah”. Beberapa contoh diantaranya yaitu, untuk kepentingan autentikasi dan identifikasi, data sidik jari seseorang harus disimpan ke dalam *database* sidik jari; kemudian untuk kepentingan bank penyimpanan data individu (seperti KTP, SIM, dll) membutuhkan suatu sistem untuk menyimpan semua data individu dan tidak jarang data sidik jari termasuk diantaranya; atau apabila ada kepentingan lain yang mengharuskan pengiriman sebuah data sidik jari melalui jaringan (*internet*, LAN, dll). Sistem penyimpanan atau pengiriman data sidik jari seperti pada contoh di atas rawan akan serangan dari profesi-profesi seperti administrator dan peretas, karena profesi administrator dapat mengakses *database* secara langsung atau peretas yang dapat menginterupsi pengiriman suatu data dalam jaringan. Sehingga reliabilitas suatu sistem biometrik hanya bergantung kepada efektivitas keamanan sistem tersebut. Oleh karena itu, dibutuhkan suatu program yang dapat menyembunyikan data biometrik sidik

jari, untuk menciptakan suatu solusi keamanan data sidik jari selain keamanan dari sistem biometrik tersebut.

Untuk mengatasi masalah keamanan data biometrik sidik jari ini, akan dibuat suatu program yang dapat menyembunyikan data sidik jari ke dalam sinyal audio. Aplikasi ini akan menyediakan fasilitas penyembunyian dan ekstraksi dari data sidik jari. Secara garis besar, proses penyembunyian data sidik jari akan melalui empat tahap. Tahap pertama adalah tahap enkripsi data sidik jari menjadi data gambar yang teracak. Tahap kedua adalah mengubah suatu sinyal audio ke dalam domain frekuensi. Tahap ketiga adalah tahap penyisipan data gambar hasil enkripsi tersebut ke dalam sinyal audio hasil transformasi. Tahap terakhir adalah mengembalikan sinyal audio dalam domain frekuensi yang telah tersisipkan oleh data gambar ke dalam sinyal audio dalam domain waktu, yang akan menjadi sinyal audio keluaran dari proses penyembunyian ini. Sebaliknya, untuk proses ekstraksi data sidik jari, sinyal audio akan ditransformasi ke dalam domain frekuensi, data gambar akan diekstraksi dari domain frekuensi tersebut, kemudian didekripsi untuk menghasilkan data sidik jari yang asli. Sinyal audio (*stego-signal*) keluaran dari aplikasi ini diharapkan dapat menjadi bentuk yang lebih aman daripada data gambar sidik jari yang biasa.

Pada tugas akhir ini, metode enkripsi yang digunakan adalah [1]*Chaotic Encryption* yang dapat mengenkripsi suatu gambar menjadi suatu map permutasi yang dinamis. Metode transformasi *Discrete Fourier Transform* yang digunakan pada tugas akhir ini adalah *Walsh-Hadamard Transform* [2] dan *Discrete Cosine Transform* [3]. Dengan adanya program ini, diharapkan dapat meningkatkan keamanan dari data biometrik sidik jari, baik data tersebut disimpan dalam *database* atau dikirim melalui suatu jaringan.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengimplementasikan metode *Chaotic Encryption* untuk mengenkripsi gambar beserta dekripsinya?
2. Bagaimana melakukan transformasi fourier pada sinyal audio beserta *inverse*-nya?
3. Bagaimana menyisipkan data gambar pada domain frekuensi audio?
4. Bagaimana membandingkan dan menguji metode transformasi fourier yang digunakan serta kinerja aplikasi yang dibangun?

### 1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Data citra yang digunakan didapat dari *database* FVC2004-DB3, sampel yang digunakan sampel pertama dari masing-masing individu yang berjumlah 10, dipotong menjadi gambar berdimensi 256x256, dan diubah menjadi gambar biner.
2. Data audio yang digunakan berupa audio bertipe *mono* yang memiliki resolusi 16-bit, *sampling frequency* sebesar 44100 Hz, dan jumlah gelombang sebesar 65536 buah.
3. Perlakuan sistem penyimpanan *key* diasumsikan ditentukan oleh pengguna, sehingga aplikasi tidak menyediakan fasilitas untuk menyimpan atau mengirim *key* yang digunakan pada aplikasi ini.
4. Metode konstruksi map yang dibutuhkan pada tahap *Chaotic Encryption* adalah *Skew Tent Map*.
5. Metode *Discrete Fourier Transform* yang digunakan adalah *Walsh-Hadamard Transform* dan *Discrete Cosine Transform*.
6. Tidak dilakukan analisa terhadap metode enkripsi dan dekripsi yang digunakan pada aplikasi ini.
7. Aplikasi tidak memberikan fasilitas autentikasi (*matching* dilakukan dengan sampel sidik jari yang asli).

## 1.4 Tujuan

Tujuan Tugas Akhir ini adalah mengimplementasikan metode *Chaotic Encryption* dan *Discrete Fourier Transform* untuk menyembunyikan data biometrik sidik jari ke dalam sinyal audio beserta ekstraksinya.

## 1.5 Manfaat

Dengan dibuatnya Tugas Akhir ini maka penyimpanan data biometrik sidik jari akan lebih aman.

## 1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan yang sama dengan Tugas Akhir ini, yaitu implementasi metode *Chaotic Encryption* dan *Discrete Fourier Transform* untuk menyembunyikan data biometrik sidik jari ke dalam sinyal audio.

2. Studi literatur

Pada tahap ini dilakukan pencarian, pengumpulan, pembelajaran dan pemahaman informasi dan literatur yang diperlukan untuk pembuatan implementasi metode *Chaotic Encryption* dan *Discrete Fourier Transform* untuk menyembunyikan data biometrik sidik jari ke dalam sinyal audio. Informasi dan literatur didapatkan dari literatur buku dan sumber-sumber informasi lain yang berhubungan.

### 3. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

### 4. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

### 5. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba pada data yang telah dikumpulkan. Pengujian dan evaluasi akan dilakukan dengan menggunakan bahasa Matlab. Tahapan ini dimaksudkan untuk mengevaluasi kesesuaian data dan program serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

### 6. Penyusunan buku Tugas Akhir

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

## 1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan

pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

**Bab I    Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

**Bab II   Dasar Teori**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

**Bab III  Perancangan Perangkat Lunak**

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

**Bab IV  Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

**Bab V   Uji Coba Dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

**Bab VI  Kesimpulan Dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.



## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi penjelasan teori-teori yang berkaitan dengan algoritma yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 Biometrik**

[4]Biometrik berasal dari bahasa Yunani yaitu *bios* yang artinya hidup dan *metron* yang artinya mengukur, maka dapat diartikan sebagai studi tentang metode atau kelakuan dari manusia itu sendiri yang memiliki keunikan.

Dalam dunia teknologi informasi, biometrik relevan dengan teknologi yang digunakan untuk menganalisa fisik dan kelakuan manusia sebagai fungsi autentikasi. Contohnya dalam pengenalan fisik manusia yaitu pengenalan sidik jari, retina, iris, pola dari wajah (*facial patterns*), tanda tangan (*signature*), dan cara mengetik (*keystroke*). Ada dua faktor yang mempengaruhi timbulnya kebutuhan akan verifikasi menggunakan biometrik, yaitu :

1. **Verifikasi berdasarkan benda**  
Semua data-data yang dibutuhkan berada pada suatu benda (seperti dokumen atau kartu kredit). Apabila hilang maka orang lain dapat memalsukannya.
2. **Verifikasi berdasarkan pengetahuan**  
Umumnya verifikasi menggunakan *password*. Bahkan jika kita menggunakan algoritma enkripsi terbaik pun, tetap ada kunci yang dapat membukanya.

Keunggulan biometrik :

1. Tidak dapat hilang.
2. Sulit diduplikasi atau dipindah-tangankan.
3. Keaslian lebih terjamin

### 2.1.1 Biometrik Sidik Jari

[5]Data biometrik yang digunakan secara global sebagai fungsi autentikasi pada era ini adalah data biometrik sidik jari. Identifikasi sidik jari memiliki banyak keunggulan mulai dari pihak kepolisian hingga keamanan fasilitas. Metode identifikasi ini dilakukan dengan membandingkan antara sidik jari seseorang dan sidik jari yang diambil dari *database*. Jika sampel sidik jari cocok dengan sidik jari dari *database*, maka individu tersebut dianggap pemilik sidik jari dari *database* tersebut.

Metode identifikasi sidik jari diterima secara legal dalam semua komunitas baik pada pihak penegak hukum maupun publik, ini adalah salah satu keunggulan terbesar yang dimiliki identifikasi sidik jari. Selain itu, identifikasi sidik jari juga memiliki sejarah yang panjang dalam bidang ilmu forensik, lengkap dengan studi-studi yang mendukung kegunaan sidik jari untuk identifikasi.

Salah satu faktor penting yang dimiliki oleh sidik jari adalah akurasi. Dengan studi dan implementasi identifikasi sidik jari yang baik, kemungkinan terjadinya kesalahan identifikasi sangat kecil.

Sidik jari juga cenderung konstan pada setiap individu tidak peduli umur individu tersebut, kecuali individu yang bersangkutan mengalami kecelakaan atau bekerja dalam industri yang dapat merusak sidik jari individu tersebut. Contoh biometriks sidik jari dapat dilihat pada Gambar 2. 1.

## 2.2 Sinyal Audio

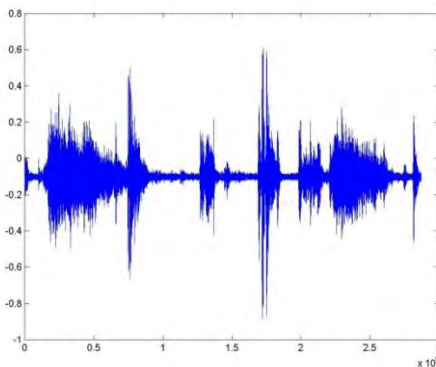
[6]Audio diartikan sebagai suara atau reproduksi suara. Sinyal audio atau gelombang suara adalah gelombang yang

dihasilkan dari sebuah benda yang bergetar pada rentang frekuensi audio. Telinga manusia dapat mendengar bunyi antara 20 Hz hingga 20.000 Hz sesuai batasan sinyal audio.



***Gambar 2. 1 Contoh Biometrik Sidik Jari***

Gelombang suara bervariasi sesuai tekanan media perantara seperti getaran dari suatu obyek, yang menyebabkan udara disekitarnya bergetar. Getaran udara ini kemudian menyebabkan gendang telinga manusia bergetar, yang kemudian oleh otak diinterpretasikan sebagai suara. Contoh gelombang suara dapat dilihat pada Gambar 2. 2.



***Gambar 2. 2 Contoh visualisasi gelombang suara***

### 2.3 *Chaotic Encryption*

*Chaotic Encryption* adalah metode enkripsi dan dekripsi berdasarkan suatu map yang bersifat kacau atau biasa disebut dengan *Chaotic Map*, dimana pembuatan *chaotic map* ini berlandaskan dari teori kekacauan atau [7]*Chaos Theory*. *Chaos Theory* adalah ilmu mengenai sifat yang tidak linier dan tidak dapat ditebak. Dimana ilmu tradisional menangani fenomena yang dapat ditebak atau dihitung seperti gravitasi, listrik, atau reaksi kimia, *chaos theory* menangani benda-benda nonlinier yang tidak dapat ditebak seperti cuaca, pasar saham, otak manusia, dan lain-lain. Beberapa prinsip *chaos theory* yang digunakan dalam sistem dinamis :

- *The Butterfly Effect* : berarti perubahan kecil dalam nilai awal akan berpengaruh secara drastis pada hasil.
- *Unpredictability* : tanpa mengetahui nilai inisial, akan sangat sulit untuk memprediksi dan memahami pola hasil.
- *Fractal* : fraktal adalah suatu pola yang tidak memiliki akhir. Fraktal dibuat dari mengulangi proses yang sama dalam fungsi rekursi.

Sifat *chaotic map* yang kacau ini digunakan sebagai alternatif untuk membangun serangkaian angka atau matriks yang *random* atau acak. *Chaotic map* dapat memiliki parameter diskrit atau kontinu, namun karena sampel pada sinyal audio yang digunakan pada tugas akhir ini adalah sampel diskrit, maka penulis menggunakan *chaotic map* berparameter diskrit-waktu yang memiliki fungsi iteratif dan sangat dipengaruhi oleh nilai inisial. *Chaotic map* memiliki banyak macam dan salah satunya adalah *skew tent map*.

[1]Proses dasar *Chaotic Encryption* adalah membangun suatu *chaotic map* dengan beberapa nilai inisial tertentu yang berfungsi sebagai *key*, kemudian mengenkripsi data dengan melakukan operasi XOR antara *chaotic map* yang dibangun

dengan data tersebut. Apabila data yang dienkripsi adalah data biner, maka *chaotic map* harus dinormalisasi dahulu menjadi data biner. Hasilnya adalah data yang teracak yang sulit untuk dipecahkan tanpa mengetahui nilai inisial yang dibutuhkan untuk membangun *chaotic map* tersebut. Proses dekripsi sama dengan proses enkripsinya, dimana *chaotic map* dibangun menggunakan *key* yang sama, kemudian melakukan operasi XOR antara data terenkripsi dengan *chaotic map* yang dibangun.

### 2.3.1 *Skew Tent Map*

[1]*Skew Tent Map* atau *Asymmetric Tent Map* merupakan salah satu jenis *chaotic map* yang *piecewise-linear* dengan *Markov chaotic map* yang terbukti memiliki keunggulan dibandingkan rangkaian *pseudorandom* yang lain, serta memiliki atribut korelasi yang terkontrol. Formula *skew tent map* dapat dilihat pada Persamaan 2.1.

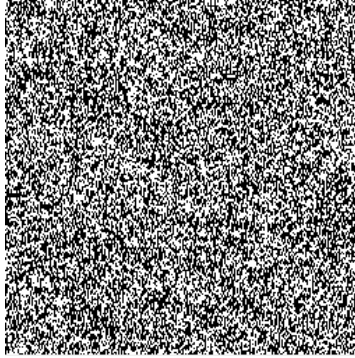
$$x(k+1) = \begin{cases} \frac{x(k)}{a} & \text{jika } 0 \leq x(k) \leq a \\ \frac{x(k)}{a} + \frac{1}{1-a} & \text{jika } a \leq x(k) \leq 1 \end{cases} \quad (2.1)$$

$a = \text{key}$

$x = \text{map hasil}$

$k = \text{indeks map } (0,1,\dots,N-1)$

Dimana  $a$  dan  $x(0)$  adalah *key* untuk enkripsi/dekripsi dengan syarat nilai  $x(0)$  berada antara 0 dan 1, kemudian syarat pemilihan  $a$  adalah  $0 < a < 0,5$  atau  $0,5 < a < 1$ , karena dengan  $a = 0,5$ , hasil akan mendekati atau sama dengan 0 dan dengan  $f(0) = 0$ , rangkaian angka tidak akan berubah. Proses pembangunan map ini adalah proses rekursi mulai dari nilai awal  $x(0)$  hingga  $x(N-1)$  dengan  $N$  adalah jumlah map yang ingin dibangun. Contoh map biner yang dihasilkan dengan metode ini dapat dilihat pada Gambar 2. 3.



**Gambar 2. 3 Contoh Binary Skew Tent Map**

#### **2.4 Discrete Fourier Transform**

[8] Transformasi fourier adalah suatu metode dasar untuk mendekomposisi serangkaian bilangan menjadi komponen-komponen frekuensi yang memiliki amplitudo dan fase. *Discrete Fourier Transform* (DFT) adalah transformasi fourier yang menggunakan sampel diskrit. DFT adalah salah satu konsep yang paling penting dalam pemrosesan sinyal digital dan tidak hanya digunakan untuk memperkirakan distribusi spektral sinyal dalam domain frekuensi, sehingga DFT banyak diimplementasikan pada banyak *Digital Signal Processing* (DSP). Formula dasar DFT dan *inverse*-nya (IDFT) dapat dilihat pada Persamaan 2.2a dan Persamaan 2.2b.

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) e^{-2\pi i k n / N} \quad (2.2a)$$

$x$  = sampel sinyal  
 $X$  = sampel frekuensi  
 $n$  = indeks sampel sinyal  
 $k$  = indeks sampel frekuensi  
 $N$  = jumlah sampel

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{2\pi i kn/N} \quad (2.2b)$$

$x$  = sampel sinyal  
 $X$  = sampel frekuensi  
 $n$  = indeks sampel sinyal  
 $k$  = indeks sampel frekuensi  
 $N$  = jumlah sampel

Fungsi  $e$  pada Persamaan 2.2a dan Persamaan 2.2b merupakan *Euler's Formula* yang memetakan suatu bilangan radian ke dalam fungsi trigonometri kosinus dan sinus seperti pada Persamaan 2.4.

$$e^{-ix} = \cos x - i \sin x \quad (2.4)$$

$x$  = bilangan dalam satuan radian  
 $i$  = bilangan imajiner

Ini berarti fungsi DFT akan memetakan domain sinyal ke dalam domain frekuensi dengan fungsi trigonometri yang terdiri dari bilangan real (fungsi kosinus) dan imajiner (fungsi sinus) atau biasa disebut dengan bilangan kompleks.

Pada umumnya perhitungan fungsi DFT dipermudah menggunakan representasi matriks  $z$ -plane menggunakan Persamaan 2.3a dan Persamaan 2.3b.

$$D(k, n) = z(k)^{-n} \quad (2.3a)$$

$n$  = indeks sampel sinyal  
 $k$  = indeks sampel frekuensi  
 $z$  = representasi fungsi DFT  
 $D$  = matriks  $z$ -plane

$$D^{-1}(n, k) = z(k)^n \quad (2.3b)$$

$n$  = indeks sampel sinyal  
 $k$  = indeks sampel frekuensi  
 $N$  = jumlah sampel  
 $z$  = representasi fungsi IDFT  
 $D^{-1}$  = *inverse* matriks *z-plane*

Fungsi  $z(k)$  pada persamaan di atas akan menghasilkan suatu matriks ( $D$ ), dimana operasi DFT dengan mudah dilakukan menggunakan formula  $X = \frac{1}{\sqrt{N}} D x$  dan *inverse*-nya dengan  $x = \frac{1}{\sqrt{N}} D^{-1} X$ .

Namun karena keterbatasan domain frekuensi hasil DFT yang hanya berupa bilangan kompleks, terdapat berbagai macam studi pengembangan metode DFT yang menghasilkan beberapa metode DFT baru, beberapa diantaranya adalah :

1. *Walsh-Hadamard Transform*
2. *Discrete Cosine Transform*

#### 2.4.1 *Walsh-Hadamard Transform*

[9] *Walsh-Hadamart Transform* (WHT) adalah salah satu variasi dari metode transformasi fourier yang sering digunakan dalam pengukuran sinyal ECG, enkripsi data, pemrosesan sinyal, dan algoritma kompresi data seperti JPEG XR dan MPEG-4 AVC. WHT adalah salah satu contoh generalisasi dari transformasi fourier diskrit dasar yang mendekomposisi suatu sinyal ke dalam set frekuensi yang bernilai real dan hanya memiliki 2 titik amplitudo, yaitu +1 dan -1.

Pada dasarnya, WHT adalah transformasi ortogonal yang mendekomposisi suatu sinyal ke dalam suatu set yang ortogonal dan gelombang persegi yang disebut dengan fungsi Walsh.



Sama halnya dengan transformasi fourier diskrit, WHT dapat digambarkan dengan sebuah matriks yang disebut matriks Hadamard, dimana matriks tersebut harus memiliki atribut-atribut berikut ini :

1.  $H^T \cdot H = nI$ , dengan I adalah matriks identitas.
2.  $n = 2^p$ , dengan P adalah bilangan integer positif.
3. matriks H dapat dikonfigurasi sehingga semua elemen dari baris pertama dan semua elemen dari kolom pertama dijumlah menjadi n.

Contoh matriks Hadamard dasar yang berukuran 2x2 ( $H_2$ ) dapat dilihat pada Gambar 2. 4. Untuk menghasilkan matriks Hadamard 4x4 ( $H_4$ ), cukup menggunakan formula pada Gambar 2. 5. Dengan formula yang sama, matriks  $H_8$  dapat dilihat pada Gambar 2. 6.

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

**Gambar 2. 4 Matriks Hadamard 2x2 ( $H_2$ )**

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

**Gambar 2. 5 Matriks Hadamard 4x4 ( $H_4$ )**

$$H_8 = \left[ \begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array} \right]$$

**Gambar 2. 6 Matriks Hadamard 8x8 ( $H_8$ )**

Kemudian setelah matriks Hadamard dihasilkan, sampel hasil didapatkan dengan mudah menggunakan formula  $X = (1/N)Hx$  dan *inverse*-nya (IWHT) dengan formula  $x = HX$ . Dapat dilihat bahwa sesuai dengan atribut pertama, matriks  $H$  bila dikalikan dengan matriks  $H$  itu sendiri akan menghasilkan matriks identitas, sehingga formula IWHT juga menggunakan matriks yang sama.

### 2.4.2 *Discrete Cosine Transform*

*Discrete Cosine Transform* (DCT) adalah salah satu metode transformasi yang berasal langsung dari modifikasi *Discrete Fourier Transform* (DFT), dimana DFT dipecah menjadi *Discrete Sine Transform* (DST) dan DCT. Perbedaannya adalah DCT murni menggunakan bilangan real dalam perhitungannya, sedangkan DST murni menggunakan bilangan imajiner. DCT adalah salah satu metode transformasi yang paling sering digunakan, salah satu kegunaan DCT yang sering digunakan adalah sebagai metode kompresi mulai dari kompresi data, gambar, audio, hingga video.

[3]Ada empat tipe DCT, namun DCT yang secara umum digunakan pada aplikasi kehidupan nyata adalah DCT tipe 2. Formula DCT 1-dimensi tipe 2 dapat dilihat pada Persamaan 2.5.

$$X(k) = w(k) \sum_{n=1}^N x(n) \cos \left[ \frac{\pi}{2N} (2n-1)(k-1) \right] \quad (2.5)$$

$$\text{dimana : } w(k) \begin{cases} \frac{1}{\sqrt{N}}, k = 1 \\ \sqrt{\frac{2}{N}}, 2 \leq k \leq N \end{cases}$$

$x$  = sampel sinyal

$X$  = sampel frekuensi

$n$  = indeks sampel sinyal

$k$  = indeks sampel frekuensi (1, 2, ...,  $N$ )

$N$  = jumlah sampel

Sedangkan *Inverse DCT* (IDCT) 1-dimensi dapat dilihat pada Persamaan 2.6.

$$x(n) = \sum_{k=1}^N w(k) X(k) \cos \left[ \frac{\pi}{2N} (2n-1)(k-1) \right] \quad (2.6)$$

$$\text{dimana : } w(k) \begin{cases} \frac{1}{\sqrt{N}}, k = 1 \\ \sqrt{\frac{2}{N}}, 2 \leq k \leq N \end{cases}$$

$x$  = sampel sinyal

$X$  = sampel frekuensi

$n$  = indeks sampel sinyal (1, 2, ...,  $N$ )

$k$  = indeks sampel frekuensi

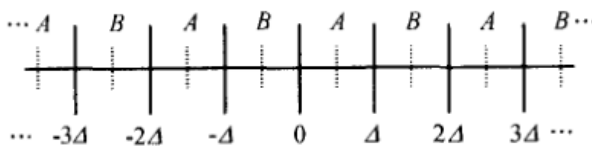
$N$  = jumlah sampel

Dapat dilihat bahwa formula DCT tidak menggunakan fungsi trigonometri sinus lagi, ini dikarenakan DCT tidak perlu memetakan bilangan imajiner, sehingga hasil dari DCT adalah bilangan yang real dan ortogonal. Kemudian pada formula DCT, variabel  $n$  akan membentuk suatu fase bilangan ganjil, ini

dikarenakan pada DFT, yang memetakan bilangan realnya adalah fungsi kosinusnya, sedangkan fungsi sinus akan memetakan bilangan imajinernya. Sehingga formula DCT sering disebut “fungsi real-DFT”. Formula DCT juga tergolong dalam kategori *Fast Fourier Transform* (FFT) karena dalam formulanya hanya ada perhitungan bilangan real, dibandingkan dengan DFT standar yang harus melakukan perhitungan bilangan real dan bilangan imajiner. Contohnya proses perkalian pada DCT hanya perkalian biasa yaitu  $U \times V = W$ , sedangkan perkalian pada DFT perlu melakukan perkalian bilangan kompleks  $(X + Yi)(U + Vi) = XU + YV + XVi + UYi$ . Perbedaan ini menghemat waktu yang cukup signifikan dibandingkan metode DFT yang standar.

## 2.5 Kuantisasi

[1]Ide pokok dari kuantisasi adalah membagi sampel frekuensi ke dalam beberapa set sesuai dengan jarak terdekatnya terhadap set tersebut. Kuantisasi digunakan untuk menyisipkan sampel gambar biner ( $WW$ ) dengan cara mengubah nilai frekuensi ke dalam set modulus ganjil yang mewakili nilai biner 0 dan set modulus genap mewakili nilai biner 1. Ilustrasi kuantisasi dapat dilihat pada Gambar 2. 7.



**Gambar 2. 7 Ilustrasi Pembagian Sampel Frekuensi dengan Quantization Step ( $\Delta$ )**

Gambar di atas ini memiliki arti frekuensi dibagi ke dalam set A dan B sesuai dengan *quantization step* ( $\Delta$ ), dimana set A dan B memiliki arti tersendiri sesuai data yang ingin disisipkan. Dalam hal ini, set A berarti hasil bagi sampel frekuensi dengan  $\Delta$

yang bernilai genap, sedangkan set B berarti hasil bagi sampel frekuensi dengan  $\Delta$  yang bernilai ganjil.

Sebelum dilakukan proses kuantisasi, dibutuhkan beberapa variabel pendukung, yaitu modulus ( $m$ ) dan residual ( $r$ ). Modulus  $m$  dihitung menggunakan Persamaan 2.7.

$$m(k) = \left\lfloor \frac{f(k)}{\Delta} \right\rfloor \quad (2.7)$$

$m$  = modulus  
 $\Delta$  = *quantization step*  
 $f$  = sampel frekuensi  
 $k$  = indeks sampel

Residual  $r$  dihitung menggunakan Persamaan 2.8.

$$r(k) = f(k) - m(k) \times \Delta \quad (2.8)$$

$m$  = modulus  
 $\Delta$  = *quantization step*  
 $f$  = sampel frekuensi  
 $r$  = residual  
 $k$  = indeks sampel

Aturan kuantisasi dibagi menjadi dua, yaitu untuk sampel gambar bernilai 0 dan sampel gambar bernilai 1. Aturan kuantisasi untuk sampel gambar bernilai 0 dapat dilihat pada Persamaan 2.9 dan sampel gambar bernilai 1 pada Persamaan 2.10. Untuk sampel gambar  $WW(k) = 0$ , sampel frekuensi dinormalisasi ke set modulus ganjil :

$$Fq(k) = \begin{cases} m(k) * \Delta + \frac{\Delta}{2} & \text{jika } m(k) \text{ bernilai ganjil} \\ m(k) * \Delta + \frac{3\Delta}{2} & \text{jika } m(k) \text{ bernilai genap dan} \\ & r(k) > \Delta/2 \\ m(k) * \Delta - \frac{\Delta}{2} & \text{jika } m(k) \text{ bernilai genap dan} \\ & r(k) \leq \Delta/2 \end{cases} \quad (2.9)$$

$m$  = modulus

$r$  = residual

$\Delta$  = *quantization step*

$Fq$  = frekuensi hasil

$k$  = indeks sampel

Untuk sampel gambar  $WW(k) = 1$ , sampel frekuensi dinormalisasi ke set modulus genap :

$$Fq(k) = \begin{cases} m(k) * \Delta + \frac{\Delta}{2} & \text{jika } m(k) \text{ bernilai genap} \\ m(k) * \Delta + \frac{3\Delta}{2} & \text{jika } m(k) \text{ bernilai ganjil dan} \\ & r(k) > \Delta/2 \\ m(k) * \Delta - \frac{\Delta}{2} & \text{jika } m(k) \text{ bernilai ganjil dan} \\ & r(k) \leq \Delta/2 \end{cases} \quad (2.10)$$

$m$  = modulus

$r$  = residual

$\Delta$  = *quantization step*

$Fq$  = frekuensi hasil

$k$  = indeks sampel

### 2.5.1 Re-kuantisasi

[1]Proses re-kuantisasi adalah proses mengekstraksi sampel gambar dari sampel frekuensi yang telah dikuantisasi, dimana sampel dengan modulus ganjil diterjemahkan sebagai piksel biner 0 dan modulus genap diterjemahkan sebagai piksel biner 1.

Untuk melakukan proses re-kuantisasi, cukup menghitung modulus  $m$  menggunakan Persamaan 2.7, kemudian menggunakan Persamaan 2.11 untuk membangun data gambar.

$$WW(k) = \begin{cases} 0 & \text{jika } m(k) \text{ ganjil} \\ 1 & \text{jika } m(k) \text{ genap} \end{cases} \quad (2.11)$$

$WW$  = array hasil re-kuantisasi

$m$  = modulus

$k$  = indeks sampel

Data gambar yang dihasilkan dari persamaan di atas adalah gambar biner (0 dan 1). Kesalahan pada proses pengambilan sampel frekuensi, nilai *quantization step* yang salah, atau pengubahan sampel frekuensi yang melebihi nilai *quantization step* akan mengurangi akurasi gambar yang diekstraksi dari proses ini. Maka dari itu perlakuan terhadap sampel frekuensi yang terkuantisasi akan sangat mempengaruhi keberhasilannya proses kuantisasi ini. Idealnya, nilai *quantization step* yang digunakan adalah nilai yang sebesar mungkin, karena semakin besar nilai *quantization step*, maka semakin besar nilai toleransi kesalahan pada tahap re-kuantisasi ini. Namun semakin besar nilai *quantization step*, maka pergeseran nilai pada tahap kuantisasi akan semakin besar.

## 2.6 *Signal-To-Noise Ratio*

[10] Pada komunikasi analog dan digital, *signal-to-noise ratio*, sering disebut S/N atau SNR, adalah metode pengukuran kekuatan suatu sinyal relatif terhadap *noise*-nya. Rasio biasa diukur dalam desibel (dB). Formula SNR dapat dilihat pada Persamaan 2.12.

$$SNR = 10x \log_{10} \left( \frac{A_{sinyal}}{A_{noise}} \right)^2 \quad (2.12)$$

$A_{sinyal}$  = amplitudo sinyal

$A_{noise}$  = amplitudo *noise*

Berdasarkan definisi, semakin besar rasio SNR maka semakin bagus kualitas sinyal tersebut. Batas minimum rasio SNR yang digunakan dalam tugas akhir ini adalah 35 dB, karena ketika rasio SNR mencapai 35 dB, akan ada perbedaan yang signifikan pada sinyal audio dengan *noise*-nya hingga terdengar oleh indra pendengaran manusia [1].

## 2.7 Additive White Gaussian Noise

[11] *Additive White Gaussian Noise* (AWGN) adalah model sederhana dari kecacatan pada saluran komunikasi. Ketika kita mengirim suatu sinyal ke dalam ruang, atmosfer, atau kabel tembaga, akan ada gangguan (*noise*) pada saluran pengiriman karena berbagai alasan. Salah satu alasan tersebut adalah *noise* termal yang dikarenakan pergerakan elektron dalam sirkuit elektronik yang digunakan baik untuk transmisi maupun penerimaan sinyal. Gangguan atau *noise* ini dimodelkan dengan *additive white gaussian noise*. Perilaku *noise* ini pada domain waktu dapat dijabarkan sebagai berikut :

- **Additive**

Karena *noise* akan ditambahkan ke dalam sinyal yang akan dikirimkan. Sehingga sinyal yang diterima  $y(t) = x(t) + n(t)$ , dimana  $x(t)$  adalah sinyal original yang dikirimkan dan  $n(t)$  adalah *noise* pada saluran pengiriman.

- **Gaussian**

Konstruksi keacakan *noise*  $n(t)$  menggunakan distribusi normal atau biasa disebut distribusi *Gaussian*.

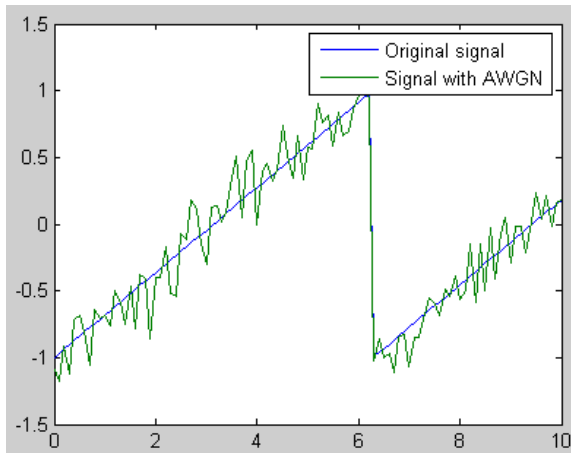
Kemudian perilaku *noise* pada domain frekuensi adalah :

- **White**

Ini berarti *noise* memiliki kuadrat (*power*) yang sama untuk semua frekuensi.



Ilustrasi AWGN dapat dilihat pada Gambar 2. 8.



**Gambar 2. 8 Ilustrasi sinyal yang diberi AWGN**

## 2.8 Matching

[1] *Matching* adalah suatu teknik untuk membandingkan suatu obyek dengan obyek pembanding. Dalam proses *matching* gambar, suatu gambar dibandingkan dengan gambar pembanding (*template*) dan akan menghasilkan suatu nilai kemiripan antara kedua gambar tersebut atau juga bisa disebut sebagai akurasi. Proses *matching* gambar biner yang standar hanya membandingkan kesamaan masing-masing piksel antara kedua gambar, kemudian diukur kemiripannya. Formula pengukuran akurasi antara dua buah gambar dapat dilihat pada Persamaan 2.13.

$$Akurasi = \frac{N - \sum_{n=0}^{N-1} x(n) \oplus y(n)}{N} \quad (2.13)$$

*Akurasi* = nilai akurasi gambar

*x* = gambar pertama

$y$  = gambar kedua  
 $n$  = indeks piksel gambar  
 $N$  = jumlah sampel

Formula ini membandingkan tiap piksel dari kedua gambar biner dan nilai akurasi akan berkurang untuk tiap piksel yang tidak bernilai sama pada kedua gambar.

## **BAB III**

### **DESAIN PERANGKAT LUNAK**

Pada bab ini akan dijelaskan perancangan program yang dibuat. Perancangan akan dibagi menjadi lima proses utama, yaitu:

1. Enkripsi data biometrik sidik jari menggunakan *Chaotic Encryption (Skew Tent Map)*.
2. Pengubahan sinyal digital audio ke dalam domain frekuensi menggunakan transformasi fourier.
3. Penyisipan data yang telah terenkripsi ke dalam frekuensi audio menggunakan metode kuantisasi frekuensi.
4. Pengembalian audio dari domain frekuensi ke sinyal digital menggunakan *inverse* dari transformasi fourier.
5. Ekstraksi data sidik jari dari audio-stego dengan transformasi fourier, re-kuantisasi, dan dekripsi *Skew Tent Map*.

Pada bab ini akan dijelaskan gambaran umum setiap program utama dalam *flowchart* selanjutnya untuk penjelasan lebih detail akan disajikan dalam *pseudocode*.

#### **3.1 Desain Metode Secara Umum**

Metode ini memiliki dua tahap inti, yaitu tahap penyembunyian sidik jari dan tahap ekstraksi sidik jari. Tahap penyembunyian sidik jari terdiri dari empat proses. Dimulai dari tahap enkripsi gambar sidik jari menggunakan *Skew Tent Map* serta mengubah sinyal digital audio ke dalam domain frekuensi menggunakan transformasi fourier (*Walsh-Hadamard Transform* dan *Discrete Cosine Transform*). Kemudian penyisipan data yang telah terenkripsi ke dalam domain frekuensi audio dengan aturan kuantisasi. Kemudian dikembalikan ke sinyal digital audio dengan *inverse* transformasi fourier (*Inverse Walsh-Hadamard Transform* dan *Inverse Discrete Cosine Transform*). Pada tahap ini akan dilakukan perhitungan *Signal-to-noise rasio* (SNR) untuk

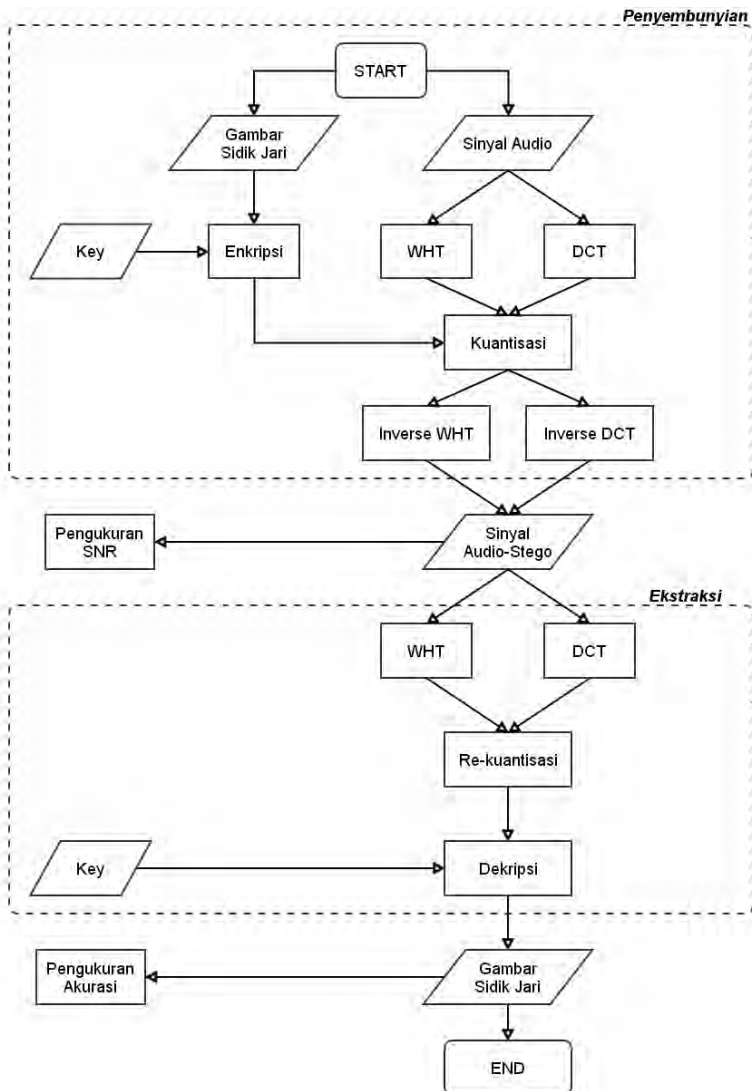
membandingkan perbedaan sinyal audio asli dengan sinyal audio-stego. Tahap ekstraksi sidik jari terdiri dari komplemen proses-proses penyembunyian sidik jari. Dimulai dari mengubah audio-stego ke dalam domain frekuensi dengan transformasi fouriernya, re-kuantisasi untuk mendapatkan gambar yang terenkripsi, dan dekripsi *Skew Tent Map* untuk mendapatkan gambar sidik jari. Terakhir adalah perhitungan akurasi dari data yang telah diekstraksi dengan data sidik jari asli. Penerapan seluruh tahapan pada tugas akhir ini menggunakan program Matlab. Diagram alur dari proses ditunjukkan oleh Gambar 3. 1.

### 3.2 Penentuan Key

Proses penentuan *key* digunakan pada tahap penyembunyian dan tahap ekstraksi. *Key* yang diperlukan ada 3 yaitu  $a$  dan  $x$  yang digunakan pada tahap enkripsi dan dekripsi, dimana pemilihan nilai  $a$  dan  $x$  sesuai ketentuan yang dijelaskan pada subbab 2.3.1. Aplikasi tidak menyediakan fasilitas untuk menyimpan atau mengirim *key*, sehingga diasumsikan *key* dimasukkan oleh pengguna aplikasi.

### 3.3 Penyembunyian Sidik Jari

Pada bagian ini akan dijelaskan ke-empat proses penyembunyian sidik jari ke dalam sinyal audio yang terdiri dari enkripsi menggunakan *Skew Tent Map*, transformasi fourier (*Walsh-Hadamard Transform* dan *Discrete Cosine Transform*), kuantisasi, dan *inverse* dari transformasi fourier (*Inverse Walsh-Hadamard Transform* dan *Inverse Discrete Cosine Transform*). Kemudian hasil sampel audio dinormalisasi ke dalam rentang sinyal audio 16-bit. Juga disertakan proses pengujian *Signal-to-noise ratio* untuk mengukur perbedaan antara sinyal audio yang original dengan sinyal audio yang dihasilkan dari tahap penyembunyian sidik jari ini.



**Gambar 3. 1 Diagram Alur Aplikasi**

### 3.3.1 *Skew Tent Map Encryption*

Proses enkripsi gambar sidik jari diawali dengan membangun sebuah *chaotic map* dengan ukuran sesuai banyaknya piksel pada gambar sidik jari yang akan dienkripsi. Konstruksi map ini dilakukan menggunakan Persamaan 2.1 dimana dalam persamaan tersebut terdapat dua buah *key* yaitu  $a$  dan  $x(0)$ . *Key* ini didapat dari tahap penentuan *key* pada subbab 3.2. Kemudian dilakukan operasi XOR untuk masing-masing piksel gambar sidik jari dengan map yang telah dibangun. Hasil yang didapat adalah sebuah gambar yang telah terenkripsi. *Pseudocode* dari proses enkripsi ini dapat dilihat pada Gambar 3. 2.

Masukan	Sampel sidik jari, key
Keluaran	Sampel sidik jari terenkripsi
<ol style="list-style-type: none"> <li>1. data = sampel sidik jari</li> <li>2. key = key untuk <i>generate</i> map</li> <li>3. map = <i>SkewTentMap</i>(data, key)</li> <li>4. dataEncrypted = <i>xor</i>(data, map)</li> <li>5. return dataEncrypted</li> </ol>	

**Gambar 3. 2 *Pseudocode Skew Tent Map Encryption***

### 3.3.2 *Walsh-Hadamard Transform*

Pada tahap ini, sampel sinyal digital akan diubah menjadi sinyal frekuensi menggunakan metode WHT. Proses dimulai dengan membangun matriks Hadamard sesuai panjang sampel yang ingin ditransformasi. Kemudian menerapkan formula Sampel frekuensi hasil proses WHT adalah murni bilangan real yang kemudian dapat dilakukan proses kuantisasi pada tahap selanjutnya. *Pseudocode* WHT dapat dilihat pada Gambar 3. 3.

Masukan	Sampel sinyal
Keluaran	Sampel frekuensi
<pre> 1. data = sampel sinyal 2. N = panjang data 3. H = membangun matriks Hadamard sebesar N 4. F = (1/N) * H * data //perkalian matriks 5. return F </pre>	

**Gambar 3. 3 Pseudocode WHT**

### 3.3.3 Discrete Cosine Transform

Pada tahap ini, sampel sinyal digital akan diubah menjadi sinyal frekuensi menggunakan metode *Discrete Cosine Transform* (DCT) menggunakan Persamaan 2.5. Sampel frekuensi hasil proses DCT adalah murni bilangan real yang kemudian dapat dilakukan proses kuantisasi pada tahap selanjutnya. *Pseudocode* DCT dapat dilihat pada Gambar 3. 4.

Masukan	Sampel sinyal
Keluaran	Sampel frekuensi
<pre> 1. data = sampel sinyal 2. N = size(data) 3. for k=1:N 4.     for n=1:N 5.         if k=1 6.             W = 1/sqrt(N) 7.         else 8.             W = sqrt(2/N) 9.         end if 10.        F(k)=F(k)+(W * data(n) * cos(pi*(2*n-1)*(k-1)/2*N)) 11.    end for 12. end for 13. return F </pre>	

**Gambar 3. 4 Pseudocode DCT**

### 3.3.4 Kuantisasi

Tahap kuantisasi adalah tahap menormalisasi sampel frekuensi sesuai aturan kuantisasi pada Persamaan 2.9 dan Persamaan 2.10. Formula kuantisasi dibatasi untuk bekerja hanya dengan bilangan real, ini sebabnya implemetasi DFT dasar secara langsung tidak dapat dilakukan, karena sampel frekuensi hasil dari DFT adalah bilangan kompleks. Bila dipaksakan, hilangnya bilangan imajiner setelah proses kuantisasi berlangsung akan menyebabkan proses *inverse*-nya yang tidak sempurna, sehingga pada saat dilakukan ekstraksi, akurasi gambar yang diekstraksi akan berkurang secara signifikan.

Pada dasarnya tahap kuantisasi hanya mengubah sampel frekuensi ke dalam set terdekat, dimana pemilihan set tersebut didasari oleh piksel data yang ingin disisipkan. Dalam hal ini data yang ingin disisipkan adalah sampel sidik jari yang terenkrripsi.

Kemudian beberapa variabel pendukung proses kuantisasi juga perlu diterapkan pada fungsi ini, yaitu menghitung modulus  $m$  dan residual  $r$  menggunakan Persamaan 2.7 dan Persamaan 2.8. Selain itu, variabel yang berperan dalam fungsi kuantisasi adalah *Quantization Step* ( $\Delta$ ) yang didapat dari tahap penentuan *key*, dimana  $\Delta$  adalah angka pembagi yang menentukan jarak antar kedua set pada sampel frekuensi yang baru, serta signifikansi perubahan sampel frekuensi yang lama dengan yang baru.  $m$  merupakan modulus atau hasil bagi sampel frekuensi dengan  $\Delta$  yang dibulatkan ke bawah, sedangkan  $r$  adalah sisa dari hasil bagi sampel frekuensi dengan  $\Delta$ . *Pseudocode* proses kuantisasi dapat dilihat pada Gambar 3. 5.

Masukan	Sampel frekuensi, sampel yang ingin disisipkan, <i>quantization step</i>
Keluaran	Sampel frekuensi yang telah dikuantisasi



```

1. data = sampel frekuensi
2. W = sampel yang ingin disisipkan
3. qs = quantization step
4. m = floor(data/qs)
5. r = data - m*qs
6. even = bilangan genap
7. odd = bilangan ganjil
8. for i=1:length(data)
9.     if W(i)=0
10.        if m(i)=odd
11.            Fq(i)=m(i)*qs + qs/2
12.        elseif m(i)=even and r(i)>qs/2
13.            Fq(i)=m(i)*qs + 3*qs/2
14.        elseif m(i)=even and r(i)<=qs/2
15.            Fq(i)=m(i)*qs - qs/2
16.        end if
17.    elseif W(i)=1
18.        if m(i)=even
19.            Fq(i)=m(i)*qs + qs/2
20.        elseif
21.            (m(i)>=0 and m(i)=odd and r(i)>qs/2) or
22.            (m(i)<0 and m(i)=odd and r(i)<=qs/2)
23.            Fq(i)=m(i)*qs + 3*qs/2
24.        elseif
25.            Fq(i)=m(i)*qs - qs/2
26.        end if
27.    end if
28. end for
29. return Fq

```

**Gambar 3. 5 Pseudocode Kuantisasi**

### 3.3.5 *Inverse* WHT

Proses *Inverse Walsh-Hadamard Transform* (IWHT) adalah proses *inverse* dari metode *Walsh-Hadamard Transform* (WHT) yang berfungsi mengembalikan sampel frekuensi ke sampel sinyal digital. Hasil dari proses ini adalah sinyal audio yang sudah mengandung data gambar yang terenkripsi. Namun dengan adanya perubahan sampel frekuensi pada tahap kuantisasi, akan ada perbedaan sinyal audio hasil IWHT dan sinyal audio asli. Karena itu dibutuhkan pengukuran SNR dengan batas

minimum agar perbedaan sinyal audio hasil IWHT dengan sinyal audio yang asli tidak terdengar. *Pseudocode* proses IWHT dapat dilihat pada Gambar 3. 6.

Masukan	Sampel frekuensi
Keluaran	Sampel sinyal audio
<pre> 1. data = sampel frekuensi 2. N = panjang data 3. H = membangun matriks Hadamard sebesar N 4. F = H * data 5. return x </pre>	

**Gambar 3. 6 Pseudocode IWHT**

### 3.3.6 Inverse DCT

Proses *Inverse Discrete Cosine Transform* (IDCT) adalah proses *inverse* dari metode DCT yang berfungsi mengembalikan sampel frekuensi ke sampel sinyal digital. Hasil dari proses ini adalah sinyal audio yang sudah mengandung data gambar yang terenkripsi. Namun dengan adanya perubahan sampel frekuensi pada tahap kuantisasi, akan ada perbedaan sinyal audio hasil IDCT dan sinyal audio asli. Karena itu dibutuhkan pengukuran SNR dengan batas minimum agar perbedaan sinyal audio hasil IDCT dengan sinyal audio yang asli tidak terdengar. *Pseudocode* proses IWHT dapat dilihat pada Gambar 3. 7.

Masukan	Sampel frekuensi
Keluaran	Sampel sinyal audio
<pre> 1. data = sampel frekuensi 2. for k=1:N 3.     for n=1:N 4.         if k=1 5.             W = 1/sqrt(N) 6.         else </pre>	

```

7.           W = sqrt(2/N)
8.           end if
9.           x(n)=x(n)+(W * data(k) * cos(pi*(2*n-
              1)*(k-1)/2*N))
10.        end for
11. end for
12. return x

```

**Gambar 3. 7 Pseudocode IDCT**

### 3.3.7 Normalisasi Audio

Hasil dari *Inverse Walsh-Hadamard Transform* (IWHT) dan *Inverse Discrete Cosine Transform* (IDCT) ini kemudian dikeluarkan dalam bentuk audio berekstensi .wav. Perlu diperhatikan bahwa sampel sinyal audio yang akan dikeluarkan perlu berada dalam rentang yang dimiliki sinyal audio 16-bit. Sampel sinyal perlu dinormalisasi ke dalam rentang sinyal audio yang berkisar antara -32768 hingga 32767. Hal ini akan menyebabkan sedikit pergeseran nilai pada sampel sinyal yang dapat mempengaruhi domain frekuensi audio secara signifikan dan menyebabkan berkurangnya akurasi pada gambar yang disisipkan.

### 3.3.8 SNR

Pengujian *Signal-to-noise ratio* (SNR) dibutuhkan untuk mengetahui rasio perbedaan sinyal audio yang asli dengan *noise* yang dihasilkan setelah perubahan pada tahap kuantisasi. Proses pengukuran SNR dilakukan dengan membandingkan sinyal audio yang asli dengan *noise*-nya yang didapat dari selisih antara sinyal audio yang asli dengan sinyal audio-stego. *Pseudocode* pengujian SNR dapat dilihat pada Gambar 3. 8.

Masukan	Sinyal audio original, sinyal audio-stego
Keluaran	Nilai SNR
1. data = sinyal audio original 2. data2 = sinyal audio-stego 3. SNR = $10 * \log_{10}(\text{data} / (\text{data} - \text{data2}))$ 4. return SNR	

**Gambar 3. 8 Pseudocode Pengujian SNR**

### 3.4 Ekstraksi Sidik Jari

Pada bagian ini akan dijelaskan tahapan yang harus dilalui untuk mengekstraksi data sidik jari dari audio-stego. Tahapan ekstraksi data terdiri dari transformasi fourier (sesuai transformasi yang digunakan), re-kuantisasi, dan dekripsi *Skew Tent Map*. Namun karena tahap transformasi adalah sama dengan subbab 3.3.2 – 3.3.3, yang dicantumkan pada subbab ini hanya tahap re-kuantisasi dan dekripsi. Kemudian disertakan proses penambahan *White Gaussian Noise* (WGN) yang akan digunakan pada salah satu pengujian kinerja aplikasi dan proses pengukuran akurasi (*matching*) antara gambar hasil ekstraksi dengan gambar yang asli.

#### 3.4.1 *Additive White Gaussian Noise*

Tahap *Additive White Gaussian Noise* (AWGN) digunakan pada tahap pengujian untuk menguji ketahanan sinyal yang telah disisipkan data gambar terhadap serangan AWGN. AWGN akan ditambahkan pada sinyal yang akan melalui tahap ekstraksi, kemudian diuji pengaruhnya terhadap akurasi gambar yang tersisipkan. Rentang keacakan *noise* dari AWGN ditentukan dari varians dari distribusi normalnya. Sehingga masukkan dari tahap ini ada dua, yaitu sinyal audio dan varians. *Pseudocode* penambahan AWGN pada sinyal audio dapat dilihat pada Gambar 3. 9.

Masukan	Sinyal audio, varians
Keluaran	Sinyal audio dengan noise
1. input = sinyal audio 2. v = varians 3. n = <i>random noise</i> berdistribusi normal (0 ~ 1) 4. output = input + (n * sqrt(v)) 5. return output	

**Gambar 3. 9 Pseudocode Additive White Gaussian Noise**

### 3.4.2 Re-kuantisasi

Re-kuantisasi adalah tahap ekstraksi sampel gambar dari sampel frekuensi. Karena pada tahap kuantisasi, sampel frekuensi telah dibagi-bagi ke dalam set tertentu, pada tahap ini proses re-kuantisasi cukup “menerjemahkan” sampel frekuensi tersebut ke dalam suatu sampel sesuai dengan posisi sampel frekuensi pada set yang sudah ditentukan oleh *quantization step*. Variabel-variabel yang digunakan pada tahap kuantisasi kembali digunakan pada proses ini kecuali residual  $r$ . *Quantization step* ( $\Delta$ ) yang digunakan adalah sama dengan  $\Delta$  yang digunakan pada tahap kuantisasi. *Pseudocode* dari tahap ini dapat dilihat pada Gambar 3. 10.

Masukan	Sampel frekuensi, <i>quantization step</i>
Keluaran	Sampel gambar
1. data = sampel frekuensi 2. qs = <i>quantization step</i> 3. dataGambar = requantz(data, qs) 4. return dataGambar	

**Gambar 3. 10 Pseudocode Re-kuantisasi**

### 3.4.3 *Skew Tent Map Decryption*

Proses dekripsi *Skew Tent Map* sama seperti proses enkripsinya, yaitu *generate* map dan operasi XOR. Map dibangun berdasarkan *key* yang sama dengan *key* yang digunakan untuk enkripsi, kemudian dilakukan operasi XOR dengan gambar hasil dari re-kuantisasi untuk mendapatkan gambar sidik jari. *Pseudocode* proses dekripsi dapat dilihat pada Gambar 3. 11.

Masukan	Sampel hasil re-kuantisasi, key
Keluaran	Sampel sidik jari
<ol style="list-style-type: none"> <li>1. data = data hasil re-kuantisasi</li> <li>2. key = key untuk <i>generate</i> map</li> <li>3. map = <i>SkewTentMap</i>(data, key)</li> <li>4. dataGambar = xor(data, map)</li> <li>5. return dataGambar</li> </ol>	

**Gambar 3. 11 Pseudocode Dekripsi**

### 3.4.4 *Matching*

Pada tahap ini akan dilakukan pengukuran akurasi antara gambar hasil dari proses ekstraksi dengan gambar yang asli. *Pseudocode* pengukuran akurasi dapat dilihat pada Gambar 3. 12.

Masukan	Gambar asli, gambar hasil ekstraksi
Keluaran	Nilai Akurasi
<ol style="list-style-type: none"> <li>1. oi = gambar asli</li> <li>2. ei = gambar hasil ekstraksi</li> <li>3. Acc = (sum(oi XNOR ei))/N</li> <li>4. return Acc</li> </ol>	

**Gambar 3. 12 Pseudocode Matching**

## BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

### 4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah *Matlab* yang diinstal pada sistem operasi *Windows 7*.

### 4.2 Implementasi

Pada subbab ini akan dijelaskan implementasi setiap subbab yang terdapat pada bab sebelumnya yaitu bab perancangan program. Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program tugas akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

#### 4.2.1 Implementasi Penentuan *Key*

Tahap ini adalah implementasi dari penentuan *key* yang dimasukkan oleh pengguna yang dapat dilihat pada Kode Sumber 4. 1.

1	<code>% key</code>
2	<code>a = 0.7;      % 0 &lt; a &lt; 0.5    0.5 &lt; a &lt; 1</code>
3	<code>x = 0.4;      % 0 &lt; x &lt; 1</code>

***Kode Sumber 4. 1 Implementasi Penentuan Key***

#### 4.2.2 Implementasi Fungsi *Skew Tent Map Encryption*

Fungsi **SkewTentMap()** adalah fungsi untuk membangun map *skew tent map* yang disertai perubahan hasil map tersebut ke dalam bilangan biner. Dimulai dari membaca masukan sampel gambar sidik jari, memastikan gambar sidik jari adalah gambar biner, menjalankan fungsi **SkewTentMap()** untuk membuat map enkripsi, lalu lakukan operasi XOR dengan gambar sidik jari. Perlu diperhatikan juga bahwa normalisasi *chaotic map* hasil dari *Skew Tent Map* menjadi map biner terdapat di dalam fungsi **SkewTentMap()**. Implementasi fungsi masukan gambar sidik jari dan operasi XOR dapat dilihat pada Kode Sumber 4. 2 dan detail fungsi **SkewTentMap()** pada Kode Sumber 4. 3.

1	<code>% Image Input</code>
2	<code>I = imread(im);</code>
3	<code>I = im2bw(I, 0);</code>
4	<code>I = reshape(I, [1, 65536]);</code>
5	
6	<code>% Enkripsi - Skew Tent Map</code>
7	<code>% fungsi SkewTentMap(a, nilai awal, jumlah map yang ingin dibuat -1)</code>
8	<code>stm = SkewTentMap(a, x, length(I)-1);</code>
9	<code>IXor = xor(I, stm);</code>

**Kode Sumber 4. 2 Implementasi Pengambilan Sampel Gambar dan Fungsi *Skew Tent Map***

1	<code>function x = SkewTentMap(a, x1, N)</code>
2	<code>x = zeros(1, N+1);</code>
3	<code>x(1) = x1;</code>
4	<code>%skew tent map formula</code>
5	<code>for ii = 1:N</code>
6	<code>    if (x(ii) &lt; a)</code>
7	<code>        x(ii+1) = x(ii)/a;</code>
8	<code>    elseif (x(ii) &gt;= a)</code>
9	<code>        x(ii+1) = (1 - x(ii)) / (1 - a);</code>
10	<code>    end</code>



11	<code>%binary stream quantization formula</code>
12	<code>if (x(ii) &gt;= 0.5)</code>
13	<code>    x(ii) = 1;</code>
14	<code>else</code>
15	<code>    x(ii) = 0;</code>
16	<code>end</code>
17	<code>end</code>
18	<code>if (x(N+1) &gt;= 0.5)</code>
19	<code>    x(N+1) = 1;</code>
20	<code>else</code>
21	<code>    x(N+1) = 0;</code>
22	<code>end</code>
23	<code>end</code>

**Kode Sumber 4. 3 Implementasi Detil Fungsi Skew Tent Map**

#### 4.2.3 Implementasi Fungsi Walsh-Hadamard Transform

Fungsi `fwht()` adalah fungsi *Walsh-Hadamard Transform* yang disediakan oleh Matlab dengan masukan sampel sinyal untuk menghasilkan sampel frekuensi dari sinyal tersebut. Dimulai dari pengambilan sampel gelombang audio sebanyak piksel sidik jari yang ingin disisipkan, kemudian memproses sampel tersebut dengan fungsi `fwht()`. Perlu diperhatikan bahwa sampel audio yang memiliki resolusi 16-bit pada Matlab akan dinormalisasi ke dalam rentang -1 hingga 1, jadi semua proses sinyal pada aplikasi ini akan berada pada rentang normalisasi yaitu -1 hingga 1. Implementasi fungsi ini dapat dilihat pada Kode Sumber 4. 4.

1	<code>% Audio Input</code>
2	<code>[song2, fs, nbits] = wavread(au); %ambil</code> <code>sampel suara</code>
3	
4	<code>% Transformasi Fourier</code>
5	<code>F = fwht(song2,length(song2),'hadamard');</code>

**Kode Sumber 4. 4 Implementasi Pengambilan Sampel Suara dan Fungsi WHT**

#### 4.2.4 Implementasi Fungsi *Discrete Cosine Transform*

Fungsi **dct()** adalah fungsi *Discrete Cosine Transform* yang disediakan oleh Matlab dengan masukan sampel sinyal untuk menghasilkan sampel frekuensi dari sinyal tersebut. Dimulai dari pengambilan sampel gelombang audio sebanyak piksel sidik jari yang ingin disisipkan, kemudian memproses sampel tersebut dengan fungsi **dct()**. Sama dengan proses pengambilan sinyal audio pada fungsi **fwht()**, sinyal audio yang diambil berada pada rentang normalisasi -1 hingga 1. Implementasi fungsi ini dapat dilihat pada Kode Sumber 4. 5.

1	<code>% Audio Input</code>
2	<code>[song2, fs, nbits] = wavread(au); %ambil sampel suara</code>
3	
4	<code>% Transformasi Fourier</code>
5	<code>F = dct(song2);</code>

**Kode Sumber 4. 5 Implementasi Pengambilan Sampel Suara dan Fungsi DCT**

#### 4.2.5 Implementasi Fungsi Kuantisasi

Fungsi **quantz()** adalah fungsi kuantisasi untuk menyisipkan data gambar ke dalam sampel frekuensi dengan cara mengubah sampel frekuensi sesuai aturan kuantisasi. Dimulai dari memasukkan parameter sampel frekuensi yang terdiri dari sampel frekuensi, data yang ingin disisipkan, dan *quantization step* ke dalam fungsi **quantz()**, lalu menyimpan hasil kuantisasi ke dalam suatu *array*. Implementasi pemasukkan parameter fungsi **quantz()** dan penyimpanan hasil dalam suatu *array* dapat dilihat pada Kode Sumber 4. 6 dan detil fungsi **quantz()** dapat dilihat pada Kode Sumber 4. 7.

1	<code>% Quantization</code>
2	<code>W = IXor; % data gambar</code>

3	<code>Fq = quantz(F, W, qs);</code>
4	<code>Fq = reshape(Fq, [1,65536]);</code>

***Kode Sumber 4. 6 Implementasi Fungsi Kuantisasi(1)***

1	<code>function Fq = quantz(F, W, qs)</code>
2	<code>    rF = real(F); rF = rF(:);</code>
4	<code>    m = floor(rF(1:length(rF)) / qs);</code>
5	<code>    r = mod(rF(1:length(rF)), qs);</code>
6	<code>    Fq = zeros(length(rF),1);</code>
11	
12	<code>    for a=1:length(rF)</code>
13	<code>        if W(a) == 0</code>
14	<code>            if mod(m(a),2)~=0</code>
15	<code>                Fq(a) = m(a)*qs + qs/2;</code>
16	<code>            elseif (mod(m(a),2)==0) &amp;&amp;</code> <code>                (r(a) &gt; qs/2)</code>
17	<code>                Fq(a) = m(a)*qs + 3*qs/2;</code>
18	<code>            elseif (mod(m(a),2)==0) &amp;&amp;</code> <code>                (r(a) &lt;= qs/2)</code>
19	<code>                Fq(a) = m(a)*qs - qs/2;</code>
20	<code>            end</code>
21	<code>        elseif W(a) == 1</code>
22	<code>            if mod(m(a),2)==0</code>
23	<code>                Fq(a) = m(a)*qs + qs/2;</code>
24	<code>            elseif (sign(rF(a))==1   </code> <code>                sign(rF(a))==0) &amp;&amp; (mod(m(a),2)~=0) &amp;&amp;</code> <code>                (r(a) &gt; qs/2)    ( sign(rF(a))==-1 &amp;&amp;</code> <code>                mod(m(a),2)~=0 &amp;&amp; r(a) &lt;= qs/2 )</code>
25	<code>                Fq(a) = m(a)*qs + 3*qs/2;</code>
26	<code>            elseif ((sign(rF(a))==1   </code> <code>                sign(rF(a))==0) &amp;&amp; (mod(m(a),2)~=0) &amp;&amp;</code> <code>                (r(a) &lt;= qs/2))    ( sign(rF(a))==-1 &amp;&amp;</code> <code>                mod(m(a),2)~=0 &amp;&amp; r(a) &gt; qs/2 )</code>
27	<code>                Fq(a) = m(a)*qs - qs/2;</code>
28	<code>            end</code>
29	<code>        end</code>
30	<code>    end</code>

31	<code>return</code>
32	<code>end</code>

**Kode Sumber 4. 7 Implementasi Fungsi Kuantisasi(2)**

#### 4.2.6 Implementasi Fungsi *Inverse* WHT

Fungsi `ifwht()` adalah fungsi *Inverse Walsh-Hadamard Transform* yang disediakan oleh Matlab dengan masukan sampel frekuensi untuk menghasilkan sampel sinyal dari frekuensi tersebut. Sampel frekuensi yang telah dikuantisasi pada tahap ini akan dikembalikan ke dalam sinyal audio yang kemudian akan diukur SNR-nya terhadap sinyal audio asli. Implementasi fungsi ini dapat dilihat pada Kode Sumber 4. 8.

1	<code>%Inverse FT - Audio Output - iF</code>
2	<code>iF = ifwht(Fq);</code>

**Kode Sumber 4. 8 Implementasi Fungsi IWHT**

#### 4.2.7 Implementasi Fungsi *Inverse* DCT

Fungsi `idct()` adalah fungsi *Inverse Discrete Cosine Transform* yang disediakan oleh Matlab dengan masukan sampel frekuensi untuk menghasilkan sampel sinyal dari frekuensi tersebut. Sampel frekuensi yang telah dikuantisasi pada tahap ini akan dikembalikan ke dalam sinyal audio yang kemudian akan diukur SNR-nya terhadap sinyal audio asli. Implementasi fungsi ini dapat dilihat pada Kode Sumber 4. 9.

1	<code>%Inverse FT - Audio Output - iF</code>
2	<code>iF = idct(Fq);</code>

**Kode Sumber 4. 9 Implementasi Fungsi IDCT**

#### 4.2.8 Implementasi Fungsi Normalisasi dan Keluaran Audio

Fungsi *wavwrite* dari Matlab sudah menyediakan fasilitas untuk menormalisasi keluaran audio ke dalam rentang sinyal audio 16-bit (-32768 hingga 32767). Namun karena Matlab memproses sinyal dalam rentang -1 hingga 1, maka rentang normalisasi ini juga berubah menjadi -1 hingga 0.999969482421875. Fungsi *wavwrite* membutuhkan parameter sampel sinyal audio yang akan dikeluarkan, frekuensi *sampling*, jumlah bit audio, dan lokasi penyimpanan *file* audio. Implementasi keluaran sinyal audio dapat dilihat pada Kode Sumber 4. 10.

1	<code>% Audio Output</code>
2	<code>wavwrite(iF, fs, nbits, ou);</code>

**Kode Sumber 4. 10 Implementasi Fungsi Keluaran Audio**

#### 4.2.9 Implementasi Pengujian SNR

Pengujian *Signal-to-noise ratio* (SNR) diimplementasikan menggunakan fungsi dari Matlab dengan masukan sinyal audio yang original dan *noise*-nya untuk menghasilkan nilai SNR. Implementasi fungsi ini dapat dilihat pada Kode Sumber 4. 11.

1	<code>% Hitung SNR</code>
2	<code>Rsnr = snr(abs(song2), abs(song2- reshape(extF, [65536, 1])));</code>

**Kode Sumber 4. 11 Implementasi Pengujian SNR**

#### 4.2.10 Implementasi Additive White Gaussian Noise

Untuk mengimplementasikan *Additive White Gaussian Noise* (AWGN), diperlukan dua masukan, yaitu sinyal yang ingin ditambahkan AWGN dan varians dari distribusi normal yang digunakan. Fungsi *randn()* dari Matlab digunakan untuk

membentuk bilangan acak yang berdistribusi normal. Kemudian nilai varians diakar untuk mendapatkan standar deviasi, lalu dinormalisasi ke dalam rentang sinyal audio (rentang -1 hingga 1 adalah 2). Lalu nilai standar deviasi yang sudah dinormalisasi dikalikan dengan bilangan acak yang telah dibuat untuk menormalisasi bilangan acak tersebut sesuai standar deviasinya. Bilangan acak ini adalah pemodelan *noise* pada AWGN. Kemudian sinyal audio ditambahkan dengan *noise* ini sebagai model sinyal yang terganggu oleh AWGN. Implementasi fungsi ini dapat dilihat pada Kode Sumber 4. 12.

1	<code>%Gaussian-Noise varians</code>
2	<code>variance = 484;</code>
3	<code>sd = sqrt(variance);</code>
4	
5	<code>% Additive Gaussian White-Noise</code>
6	<code>gaus = ((sd/65536) * 2) .*</code> <code>randn(1, size(extF, 2));</code>
7	<code>extF = extF + gaus;</code>

**Kode Sumber 4. 12 Implementasi Additive White Gaussian Noise**

#### 4.2.11 Implementasi Fungsi Re-kuantisasi

Fungsi **requantz()** adalah fungsi re-kuantisasi dengan masukan sampel frekuensi untuk menghasilkan sampel gambar sesuai aturan re-kuantisasi. Fungsi ini memiliki dua parameter yaitu sampel frekuensi dan *quantization step* yang kemudian akan mengembalikan *array* berisi sampel gambar yang disisipkan pada sampel frekuensi yang diproses. Implementasi fungsi **requantz()** dapat dilihat pada Kode Sumber 4. 13.

1	<code>function iFq = requantz(F, qs)</code>
2	<code>    rF = real(F); rF = rF(:);</code>
3	<code>    m = floor(rF(1:length(rF)) / qs);</code>

4	<code>iFq = zeros (length (rF) , 1) ;</code>
6	<code>for a=1:length (F)</code>
7	<code>    if mod (m (a) , 2) == 0</code>
8	<code>        iFq (a) = 1 ;</code>
9	<code>    end</code>
10	<code>end</code>
11	<code>return</code>
12	<code>end</code>

***Kode Sumber 4. 13 Implementasi Fungsi Re-kuantisasi***

#### **4.2.12 Implementasi Fungsi *Skew Tent Map Decryption***

Implementasi proses dekripsi *Skew Tent Map* sama dengan proses enkripsinya, yaitu membangun map dekripsi dengan fungsi **SkewTentMap()** menggunakan Kode Sumber 4. 3, kemudian dilakukan operasi XOR dengan sampel gambar yang ingin didekripsi untuk menghasilkan sampel gambar sidik jari. Implementasi proses dekripsi dapat dilihat pada Kode Sumber 4. 14.

1	<code>tesrq = reshape (tesrq, [1, 65536]) ;</code>
2	<code>stm = SkewTentMap (0.7, 0.4123, 65535) ;</code>
3	<code>INXor = xor (tesrq, stm) ;</code>

***Kode Sumber 4. 14 Implementasi Fungsi Dekripsi***

#### **4.2.13 Implementasi Fungsi *Matching***

Implementasi proses pengukuran akurasi (*matching*) antara dua buah gambar dilakukan dengan melakukan operasi XNOR antara kedua gambar tersebut, kemudian dibagi dengan jumlah sampel gambar tersebut. Implementasi proses *matching* dapat dilihat pada Kode Sumber 4. 15.

1	<code>% Hitung Akurasi</code>
2	<code>acc = ~xor(I, INXor);</code>
3	<code>acc = sum(acc);</code>
4	<code>acc = sum(acc);</code>
5	<code>acc = acc/65536;</code>

***Kode Sumber 4. 15 Implementasi Fungsi Matching***



## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji kebenaran dan uji kinerja serta analisa setiap pengujian.

#### **5.1 Lingkungan Uji Coba**

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi metode pada Tugas Akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
  - a. Prosesor: Intel® Core™ i7-2620M CPU @ 2.70GHz
  - b. *Memory*(RAM): 4,00 GB
  - c. Tipe sistem: 64-bit sistem operasi
2. Perangkat lunak
  - a. Sistem operasi: *Windows 7 Professional*.
  - b. Perangkat pengembang: *Matlab*.

#### **5.2 Data Uji Coba**

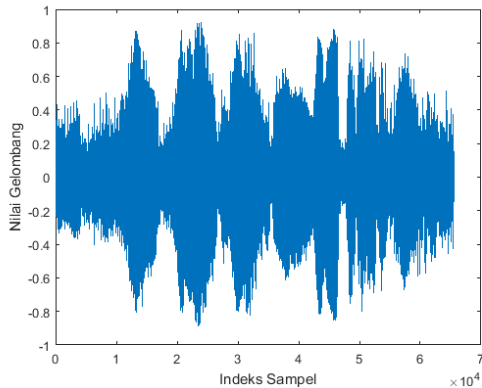
Data gambar sidik jari yang digunakan diambil dari *database* FVC2004-DB3. Terdapat 80 buah gambar sidik jari yang terdiri dari 10 individu, masing-masing memiliki 8 buah sampel sidik jari yang diambil dari sudut dan proses pengambilan yang bervariasi. Namun karena aplikasi ini tidak melakukan identifikasi sidik jari, maka cukup diambil masing-masing 1 buah sampel dari masing-masing individu. Gambar sidik jari pada *database* ini memiliki dimensi sebesar 300x480 piksel yang kemudian dipotong menjadi 256x256 piksel untuk memudahkan

proses uji coba. Kemudian warna pada gambar sidik jari ini berupa *grayscale* (hitam putih) yang bernilai 0 hingga 255, namun pada proses uji coba, warna gambar ini akan diubah menjadi gambar biner yang bernilai 0 atau 1. Contoh data sidik jari yang sudah diubah dapat dilihat pada Gambar 5. 1.



**Gambar 5. 1 Gambar Sidik Jari Individu ke-10**

Data audio yang digunakan adalah audio bertipe *mono* dan memiliki atribut resolusi 16-bit, *sampling frequency* sebesar 44,1 kHz, dan jumlah sampel sebesar 65536. Data audio diambil dari website "<http://download.wavetlan.com/SVV/Media/HTTP/http-wav.htm>" pada tab "Miscellaneous WAV" yang bernama "Test 10". Kemudian audio yang dipakai diambil pada sampel ke 100000 hingga 165535. Data audio bisa berupa sampel berekstensi .wav manapun asalkan memiliki resolusi sebesar 16-bit dan jumlah sampel yang mencukupi. *Sampling frequency* hanya mempengaruhi berapa lama durasi audio, apabila *sampling frequency* semakin besar dan dengan jumlah sampel yang sama, maka durasi audio akan semakin pendek. Sinyal audio yang digunakan pada uji coba ini dapat dilihat pada .



**Gambar 5. 2 Data Audio**

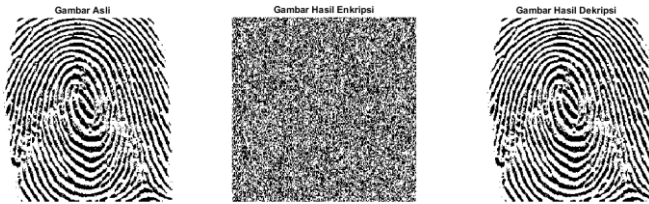
### 5.3 Uji Coba Tiap Proses

Pada subbab ini, dilakukan uji coba pada aplikasi untuk masing-masing tahapan untuk menguji kebenaran implementasi setiap proses pada aplikasi ini.

#### 5.3.1 Uji Coba Enkripsi dan Dekripsi

Uji coba ini digunakan untuk menguji metode enkripsi dan dekripsi yang digunakan pada aplikasi ini. Hasil enkripsi menggunakan metode *Skew Tent Map* adalah gambar yang teracak, dimana dekripsinya menggunakan metode dan proses yang sama. Contoh hasil enkripsi dan dekripsi sidik jari dari gambar sidik jari pertama menggunakan nilai  $a = 0.7$  dan  $x = 0.4$  dapat dilihat pada Gambar 5. 3.

Hasil uji coba menunjukkan bahwa proses enkripsi berhasil menghasilkan gambar yang teracak, sehingga tanpa mengetahui nilai  $a$  dan  $x$  akan sulit untuk mendekripsi gambar ini. Kemudian hasil uji coba dekripsi menggunakan nilai  $a$  dan  $x$  yang sama juga menghasilkan gambar sidik jari dengan akurasi sebesar 100% apabila tidak ada perubahan dari gambar hasil enkripsi.



**Gambar 5. 3 Gambar asli(kiri), gambar hasil enkripsi(tengah), dan gambar hasil dekripsi(kanan)**

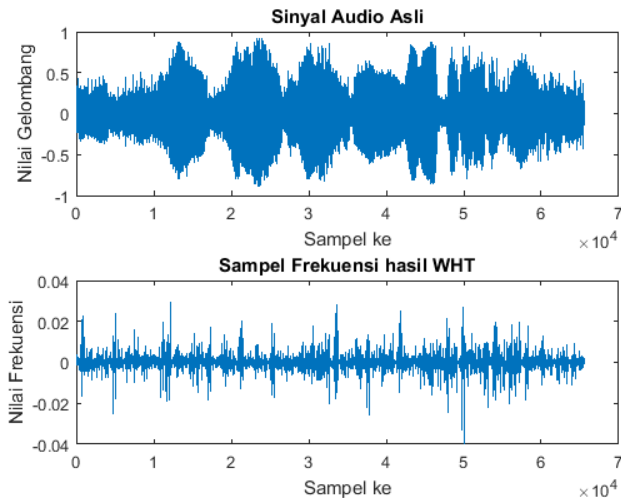
Bila memang ada perubahan pada gambar hasil enkripsi, akurasi gambar akan berkurang hanya pada piksel gambar yang berubah. [1]Metode *skew tent map* terbukti memiliki keunggulan dalam hal kehandalan dan keamanan dibanding metode *pseudorandom* yang lain, sehingga penulis tidak melakukan uji coba kriptanalisis pada tahap ini.

### 5.3.2 Uji Coba Transformasi Fourier

Pada proses ini dua metode transformasi yang diuji cobakan adalah *Walsh-Hadamard Transform* dan *Discrete Cosine Transform*. Hasil dari kedua transformasi ini adalah domain frekuensi yang real dan ortogonal. Dapat dilihat pada Gambar 5. 4 bahwa sampel frekuensi hasil WHT adalah real dan ortogonal, sehingga sampel frekuensi ini dapat digunakan pada tahap kuantisasi. Gambar 5. 5 juga menunjukkan bahwa sampel frekuensi hasil DCT adalah real dan ortogonal, namun dengan perbedaan pada jarak minimum dan maksimum sampel frekuensinya. Rentang ini akan terlihat pengaruhnya pada sampel frekuensi setelah tahap kuantisasi yang mengakibatkan perbedaan pada sinyal audio hasil transformasi balik.

Kemudian juga dibutuhkan uji coba untuk proses *inverse* dari masing-masing metode transformasi untuk sampel yang telah dikuantisasi. Uji coba ini akan dijelaskan pada subbab

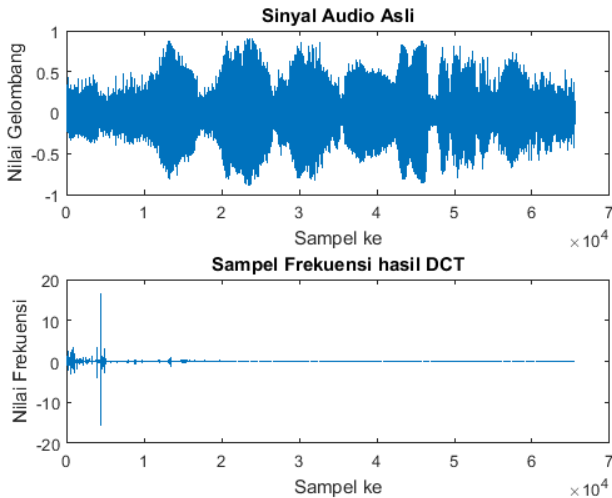
selanjutnya, karena untuk melakukan uji coba ini, diperlukan uji coba kuantisasi terlebih dahulu.



**Gambar 5. 4 Gambar sinyal audio(atas) dan gambar sinyal frekuensi hasil WHT(bawah)**

### **5.3.3 Uji Coba Kuantisasi, Re-kuantisasi, dan *Inverse* Transformasi Fourier**

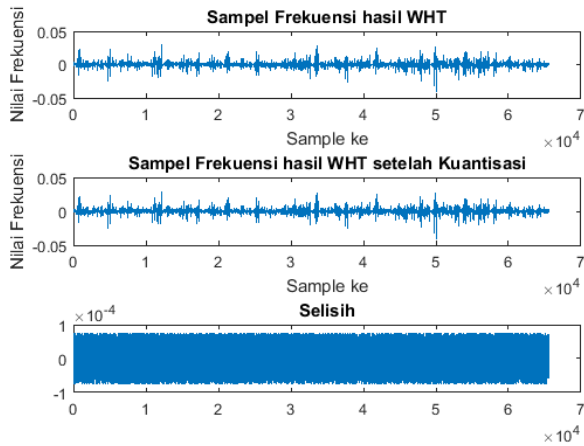
Uji coba ini dilakukan untuk menguji akurasi proses kuantisasi dan re-kuantisasi, yang disertai dengan transformasi balik sebagai kelanjutan dari uji coba transformasi fourier pada subbab 5.3. Proses kuantisasi adalah proses pengubahan sampel frekuensi sesuai data yang akan disisipkan, sedangkan re-kuantisasi adalah proses rekonstruksi sampel data yang disisipkan dalam sampel frekuensi. *Quantization step* ( $\Delta$ ) yang digunakan pada tahap ini (untuk kedua metode transformasi fourier) adalah  $5 \times 10^{-5}$ .



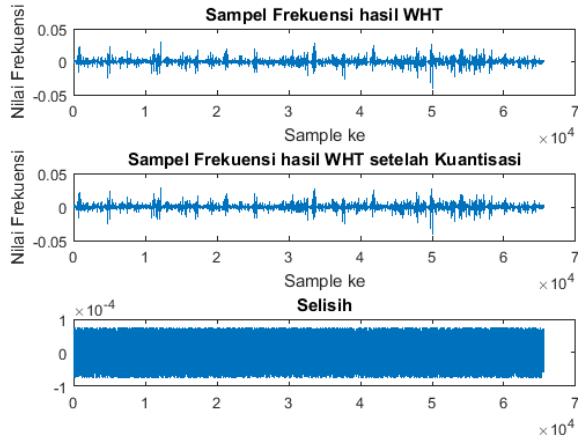
**Gambar 5. 5 Gambar sinyal audio(atas) dan gambar sampel frekuensi hasil DCT(bawah)**

Dapat dilihat pada Gambar 5. 6 dan Gambar 5. 7 bahwa hasil kuantisasi dari sampel frekuensi hasil *Walsh-Hadamard Transform* (WHT) dan *Discrete Cosine Transform* (DCT) dengan  $\Delta = 5 \times 10^{-5}$  tidak jauh berbeda dengan sampel frekuensi yang asli. Namun karena rentang sampel frekuensi hasil DCT yang lebih besar daripada rentang sampel frekuensi hasil WHT, ini akan menyebabkan perubahan pada sampel frekuensi WHT menghasilkan perubahan yang lebih signifikan pada sinyal audio hasil *inverse*-nya dibandingkan dengan perubahan pada sampel frekuensi hasil DCT.

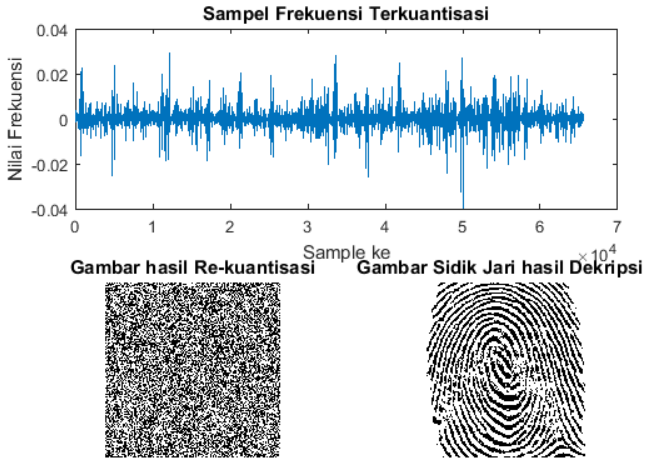
Untuk menguji akurasi proses kuantisasi, dibutuhkan uji coba dari proses re-kuantisasi, dimana pada tahap ini akan dilakukan rekonstruksi sampel gambar dari sampel frekuensi yang telah melalui tahap kuantisasi.



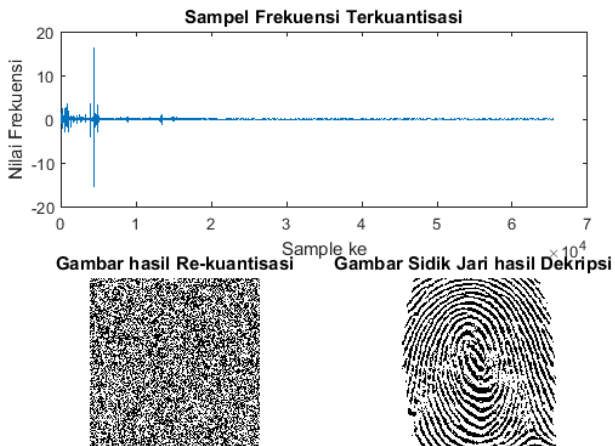
**Gambar 5. 6** Gambar frekuensi hasil WHT(atas), hasil kuantisasinya(tengah), dan selisihnya (bawah)



**Gambar 5. 7** Gambar frekuensi hasil DCT(atas), hasil kuantisasinya(tengah), dan selisihnya (bawah)



***Gambar 5. 8 Gambar sampel frekuensi WHT(atas), hasil re-kuantisasinya(kiri bawah), dan hasil dekripsinya(kanan bawah)***

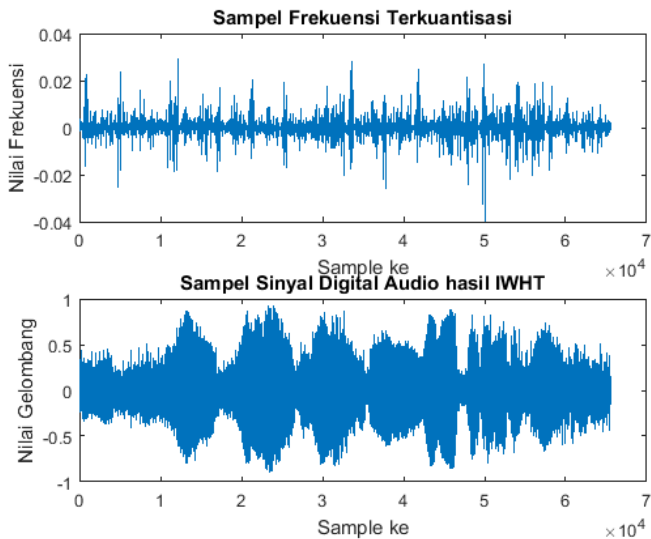


***Gambar 5. 9 Gambar sampel frekuensi DCT(atas), hasil re-kuantisasinya(kiri bawah), dan hasil dekripsinya(kanan bawah)***



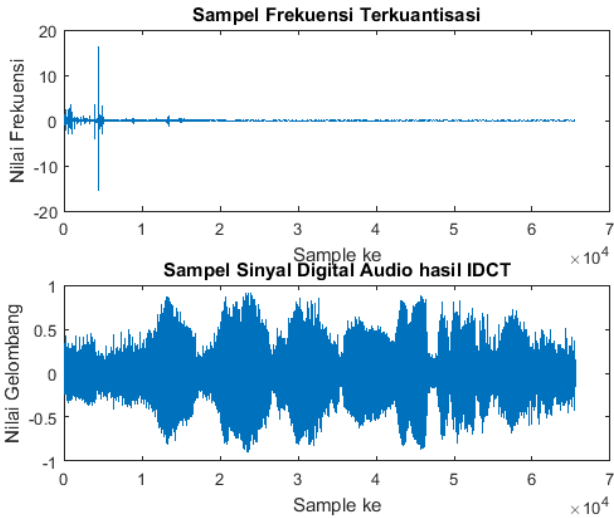
Gambar 5. 8 dan Gambar 5. 9 menunjukkan bahwa proses re-kuantisasi berhasil merekonstruksi sampel gambar yang terenkripsi dengan akurasi 100% untuk kedua metode transformasi, dibuktikan dengan hasil dekripsinya.

Kemudian lanjutan pembahasan uji coba transformasi fourier pada subbab 5.3, dilakukan uji coba untuk mengembalikan sampel frekuensi yang telah dikuantisasi ke sampel sinyal digital audio menggunakan *Inverse Walsh-Hadamard Transform* (IWHT) dan *Inverse Discrete Cosine Transform* (IDCT).



**Gambar 5. 10** Gambar sampel frekuensi WHT yang telah dikuantisasi(atas) dan gambar sinyal audio hasil IWHT(bawah)

Gambar 5. 10 dan Gambar 5. 11 menunjukkan bahwa IWHT dan IDCT berhasil merekonstruksi sinyal digital audio dengan *quantization step* =  $5 \times 10^{-5}$  dengan perbedaan yang dapat diukur dengan metode SNR antara sinyal audio hasil rekonstruksi dari proses ini dengan sinyal audio yang asli.



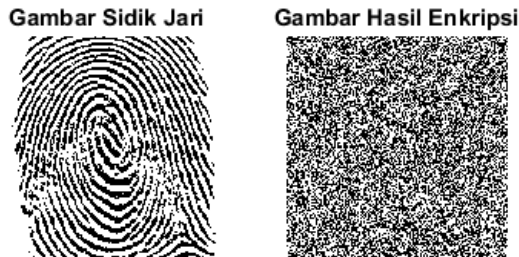
**Gambar 5. 11** Gambar sampel frekuensi DCT yang telah dikuantisasi(atas) dan gambar sinyal audio hasil IDCT(bawah)

#### 5.4 Uji Coba Secara Keseluruhan

Pada subbab ini, dilakukan uji coba pada aplikasi secara keseluruhan dengan tahap-tahap sebagai berikut: enkripsi, transformasi fourier, kuantisasi, transformasi fourier balik (*inverse*), transformasi fourier, re-kuantisasi, dan dekripsi. Uji coba ini dilakukan dengan tujuan menguji apakah aplikasi telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya. *Key* yang digunakan pada uji coba ini adalah sama yaitu  $a = 0.7$ ,  $x = 0.4$ , dan *quantization step* =  $5 \times 10^{-5}$ . Uji coba ini kemudian dilanjutkan pada subbab 5.6 untuk menguji aplikasi secara keseluruhan pada semua data untuk mencari *quantization step* yang optimal.

#### 5.4.1 Uji Coba dengan WHT dan IWHT

Uji coba pertama menggunakan *Walsh-Hadamard Transform* (WHT) dan *Inverse Walsh-Hadamard Transform* (IWHT) sebagai metode transformasi fourier dan transformasi baliknya.



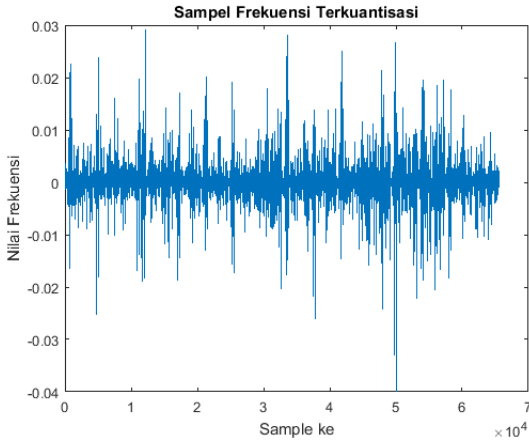
**Gambar 5. 12 Gambar sidik jari(kiri) dan enkripsinya(kanan)**

Gambar 5. 12 menunjukkan hasil enkripsi gambar sidik jari, kemudian sampel frekuensi hasil WHT dapat dilihat pada subbab sebelumnya yaitu pada Gambar 5. 4. Kedua sampel ini kemudian digunakan untuk proses selanjutnya yaitu kuantisasi. Sampel frekuensi hasil kuantisasi pada Gambar 5. 13 kemudian ditransformasi balik menggunakan IWHT.

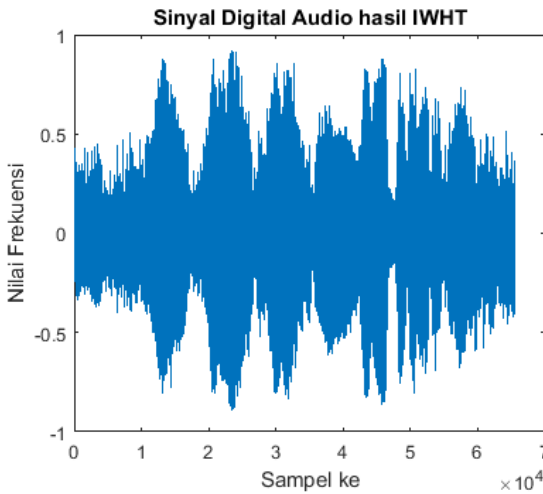
Pada tahap ini sinyal audio akan dinormalisasi ke dalam resolusi audio 16-bit sekaligus dikeluarkan dalam bentuk *file* berekstensi wav, yang kemudian *file* tersebut akan dibaca kembali dan dilakukan pengujian SNR dengan sinyal audio yang asli. Kemudian dilanjutkan dengan proses ekstraksi yang dimulai dengan proses WHT kembali.

Sampel frekuensi hasil WHT pada Gambar 5. 15 ini kemudian di-rekuantisasi untuk mendapatkan gambar yang terenkripsi. Gambar terenkripsi yang ditunjukkan pada Gambar 5. 16 didekripsi menggunakan teknik enkripsi yang sama.

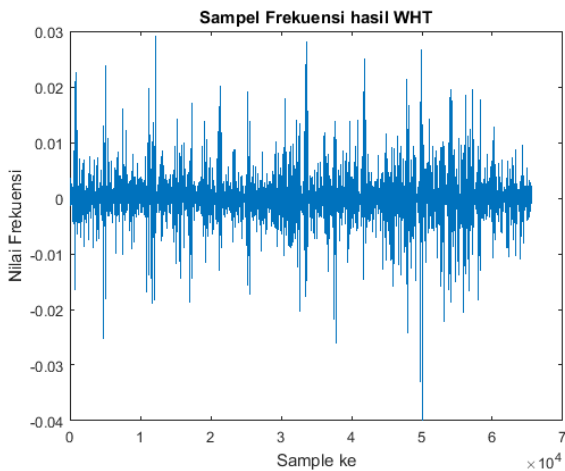
Gambar 5. 17 menunjukkan bahwa akurasi gambar hasil ekstraksi adalah 100% dengan nilai SNR di atas 35 dB untuk  $quantization\ step = 5 \times 10^{-5}$ .



**Gambar 5. 13** Gambar hasil kuantisasi sampel frekuensi WHT

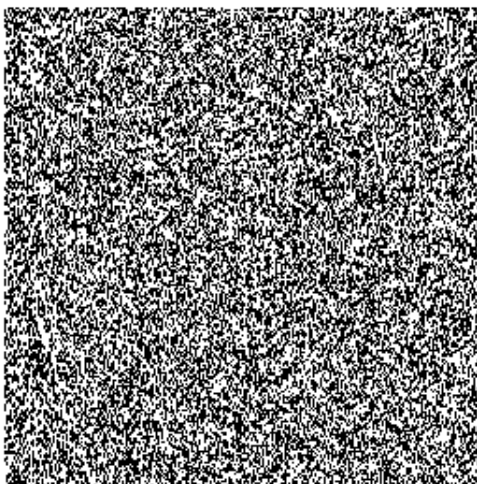


**Gambar 5. 14** Gambar sinyal audio hasil IWHT dari sampel frekuensi yang telah dikuantisasi



***Gambar 5. 15 Gambar frekuensi WHT dari sinyal audio-stego***

**Gambar hasil Re-kuantisasi**



***Gambar 5. 16 Gambar hasil Re-kuantisasi dari sampel frekuensi pada Gambar 5. 15***



***Gambar 5. 17 Gambar hasil dekripsi dari Gambar 5. 16***

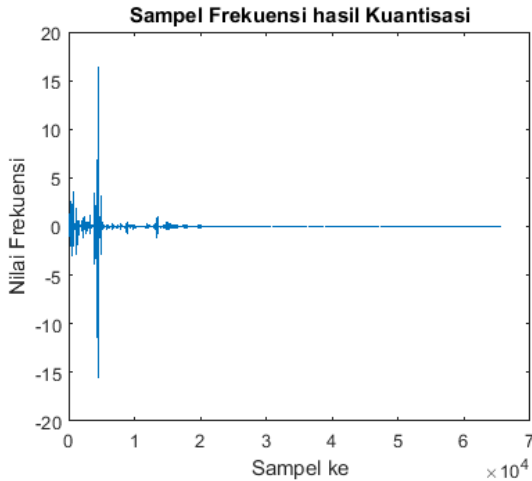
#### **5.4.2 Uji Coba dengan DCT dan IDCT**

Pada uji coba ini metode yang digunakan untuk proses transformasi dan transformasi balik adalah *Discrete Cosine Transform* (DCT) dan *Inverse Discrete Cosine Transform* (IDCT).

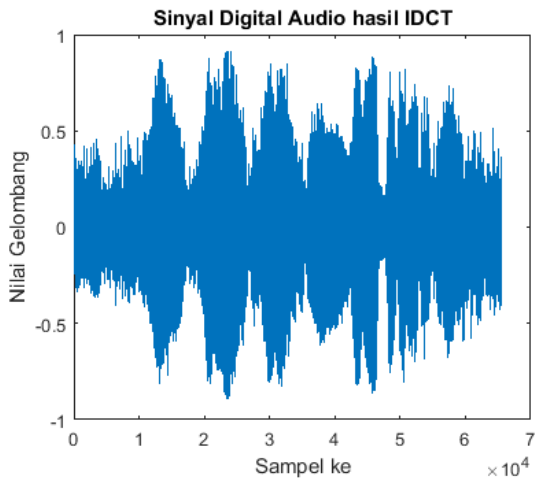
Gambar sidik jari yang digunakan sama dengan uji coba pada subbab 5.5.1 sehingga sampel gambar yang digunakan juga sama dengan Gambar 5. 12, sedangkan sampel frekuensi hasil DCT yang digunakan sama dengan sampel yang ditunjukkan pada Gambar 5. 5. Kedua sampel ini kemudian digunakan pada proses kuantisasi. Gambar 5. 18 adalah sampel frekuensi hasil kuantisasi yang kemudian ditransformasi balik menggunakan IDCT, dinormalisasi sekaligus dikeluarkan dalam bentuk *file* berekstensi *wav*.

Setelah *file* keluaran dibaca kembali, akan dilakukan pengujian SNR dengan sinyal audio yang asli, kemudian dilanjutkan ke proses ekstraksi yang dimulai dengan proses DCT kembali.

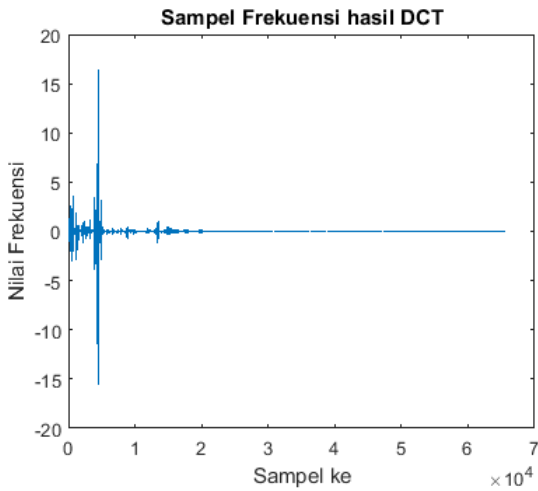
Sampel frekuensi hasil DCT pada Gambar 5. 20 kemudian dire-kuantisasi untuk mendapatkan gambar yang terenkripsi.



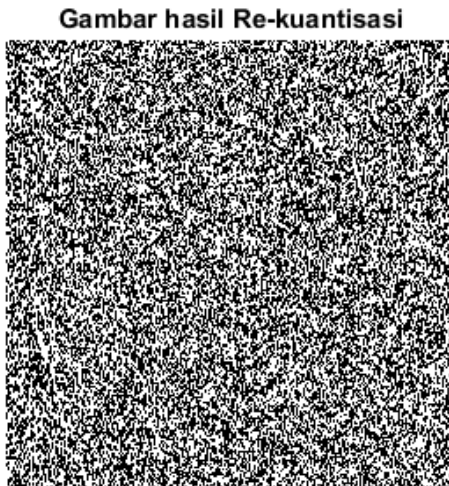
**Gambar 5. 18** Gambar hasil kuantisasi sampel frekuensi DCT



**Gambar 5. 19** Gambar sinyal audio hasil IDCT dari sampel frekuensi yang telah dikuantisasi



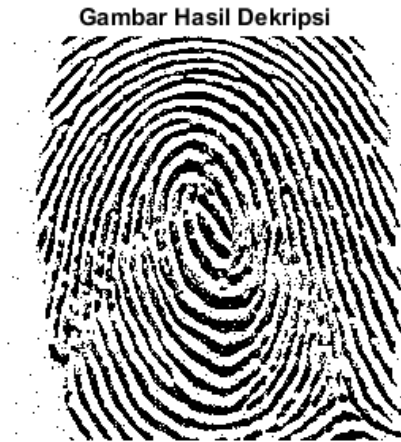
**Gambar 5. 20** Gambar frekuensi DCT dari sinyal audio-stego



**Gambar 5. 21** Gambar hasil Re-kuantisasi dari sampel frekuensi pada Gambar 5. 20



Gambar terenkripsi yang ditunjukkan pada Gambar 5. 21 didekripsi menggunakan teknik enkripsi yang sama.



**Gambar 5. 22** Gambar hasil dekripsi dari Gambar 5. 21

Gambar 5. 22 menunjukkan bahwa akurasi gambar hasil ekstraksi adalah 99,54% dengan nilai SNR di atas 35 dB untuk  $quantization\ step = 5 \times 10^{-5}$ . Berkurangnya akurasi pada pengujian inilah yang disebabkan karena normalisasi sinyal audio yang disebutkan pada subbab 3.3.7. Karena itu nilai  $quantization\ step = 5 \times 10^{-5}$  pada metode DCT tidak cocok.

## **5.5 Skenario dan Evaluasi Pengujian**

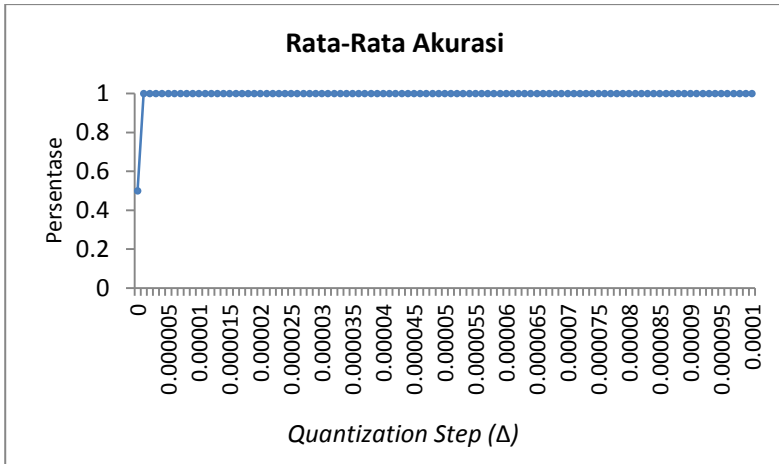
Uji coba ini dilakukan untuk mencari nilai  $quantization\ step$  yang optimal sesuai uji coba secara keseluruhan pada subbab 5.5 yang diimplementasikan pada semua data gambar sidik jari agar aplikasi dapat berjalan dengan lancar. Uji coba ini juga bertujuan mencari perbandingan kedua metode transformasi fourier yang digunakan.

Skenario pengujian terdiri dari 3, yaitu:

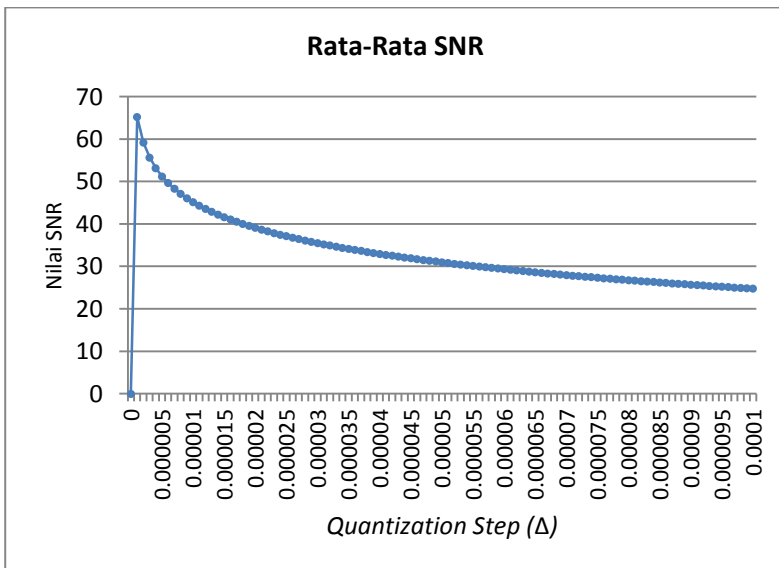
1. Skenario perhitungan SNR, akurasi, dan *running time* semua gambar sidik jari menggunakan metode *Walsh-Hadamard Transform* untuk mencari *quantization step* yang optimal.
2. Skenario perhitungan SNR, akurasi, dan *running time* semua gambar sidik jari menggunakan metode *Discrete Cosine Transform* untuk mencari *quantization step* yang optimal.
3. Skenario perhitungan ketahanan sinyal terhadap *Additive White Gaussian Noise* menggunakan *quantization step* yang optimal pada kedua metode transformasi fourier, serta mendapatkan perbandingan kedua metode transformasi fourier.

### 5.5.1 Skenario Uji Coba 1

Skenario uji coba 1 adalah perhitungan SNR, akurasi, dan *running time* menggunakan metode *Walsh-Hadamard Transform* (WHT) untuk mencari *quantization step* ( $\Delta$ ) yang optimal. Batas SNR yang diperbolehkan pada uji coba ini adalah 35 dB ke atas, SNR dengan nilai di bawah 35 dB tidak diterima sebagai kandidat  $\Delta$  yang optimal. Secara teoritis, semakin besar nilai  $\Delta$ , maka pengaruhnya adalah rasio SNR semakin kecil (kualitas sinyal audio semakin buruk), distorsi sinyal audio akibat normalisasi audio semakin tidak signifikan (akurasi gambar semakin bagus), dan ketahanan sinyal terhadap *noise* semakin bagus. Sehingga nilai  $\Delta$  yang optimal harus dicari dengan acuan ketiga parameter ini. Namun pada uji coba ini, sinyal audio diuji tanpa *noise* terlebih dahulu, sehingga parameter yang menjadi acuan pada uji coba ini adalah rasio SNR dan akurasi gambar hasil ekstraksi.. *Key* yang digunakan untuk proses enkripsi dan dekripsi adalah  $a = 0.7$  dan  $x = 0.4$ .



**Gambar 5. 23 Grafik Rata-Rata Akurasi Gambar pada Uji Coba 1 (detil pada Tabel A. 1 hingga Tabel A. 4)**



**Gambar 5. 24 Grafik Rata-Rata SNR Audio pada Uji Coba 1 (detil pada Tabel A. 1 hingga Tabel A. 4)**

Hasil percobaan menunjukkan bahwa rentang  $\Delta$  yang bagus untuk melihat pola rata-rata akurasi gambar dan rata-rata SNR audio adalah antara 0 hingga  $1 \times 10^{-4}$  dengan tahap peningkatan  $1 \times 10^{-6}$ . Detil dari Gambar 5. 23 dan Gambar 5. 24 dapat dilihat pada Tabel A. 1 hingga Tabel A. 4. Dari grafik rata-rata akurasi di atas dan detilnya dapat dilihat bahwa dengan nilai  $\Delta$  lebih besar sama dengan  $1 \times 10^{-6}$  (lebih tepatnya  $0,35 \times 10^{-6}$  dapat dilihat pada Tabel A. 6) dapat mengekstraksi gambar dengan akurasi sebesar 100%. Ini berarti normalisasi sinyal audio ke dalam resolusi 16-bit tidak berpengaruh secara signifikan pada nilai  $\Delta$  tersebut. Sedangkan dapat dilihat juga bahwa antara nilai  $\Delta = 0$  hingga  $1 \times 10^{-6}$  terdapat sinyal audio yang terpengaruh secara signifikan oleh proses normalisasi terlihat dari nilai rata-rata akurasi gambar sebesar 49,96% pada  $\Delta = 0$  dan terus meningkat hingga 100% pada  $\Delta = 1 \times 10^{-6}$ . Detil sinyal audio yang terpengaruh oleh proses normalisasi dengan nilai  $\Delta$  pada rentang ini dapat dilihat pada Gambar A. 1 dan Gambar A. 2.

Kemudian dari detil grafik rata-rata SNR pada Gambar 5. 24, terlihat bahwa pada  $\Delta = 0$ , nilai SNR adalah 0, ini disebabkan karena pada tahap kuantisasi, nilai frekuensi tidak dapat dibagi dengan 0, akibatnya hasil sinyal audio keluaran dari penyembunyian tidak dapat dibandingkan SNRnya. Untuk nilai selain 0, kita dapat melihat pola SNR yang semakin lama semakin menurun dengan membesarnya nilai  $\Delta$ . Nilai  $\Delta$  paling kecil yang dapat menghasilkan rasio SNR lebih besar dari 35dB adalah  $1 \times 10^{-6}$  yang menghasilkan sinyal audio dengan rata-rata SNR sebesar 65,21 dB dan yang paling besar adalah  $31 \times 10^{-6}$  yang menghasilkan sinyal audio dengan rata-rata SNR sebesar 35,23 dB.

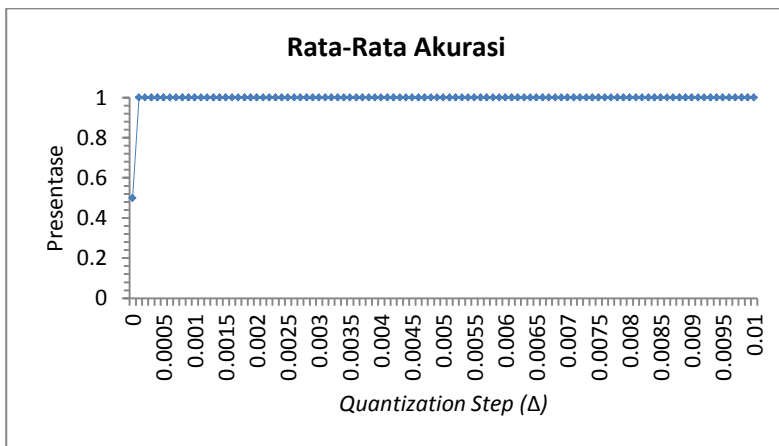
Kemudian *running time* dari aplikasi ini tidak terpengaruh oleh semua variabel yang digunakan pada aplikasi ini, sehingga *running time* hanya bergantung pada metode WHT yang memiliki kompleksitas waktu sebesar  $N \log N$ , dimana aplikasi pada uji coba ini melakukan satu proses secara

keseluruhan baik proses penyembunyian dan ekstraksi dengan rata-rata *running time* sebesar 1,77 detik.

Untuk uji coba ketiga, penulis akan membuktikan parameter terakhir, dimana semakin besar nilai  $\Delta$  maka sinyal audio yang dihasilkan akan semakin tahan terhadap *noise*. Dan karena dapat kita lihat bahwa pola SNR akan terus menurun tanpa anomali dengan meningkatnya nilai  $\Delta$ , maka nilai yang diuji cukup dua yaitu nilai  $\Delta$  terkecil dan terbesar ( $1 \times 10^{-6}$  dan  $31 \times 10^{-6}$ ) akan diambil sebagai kandidat nilai  $\Delta$  yang optimal untuk diuji ketahanannya terhadap *noise* pada uji coba 3.

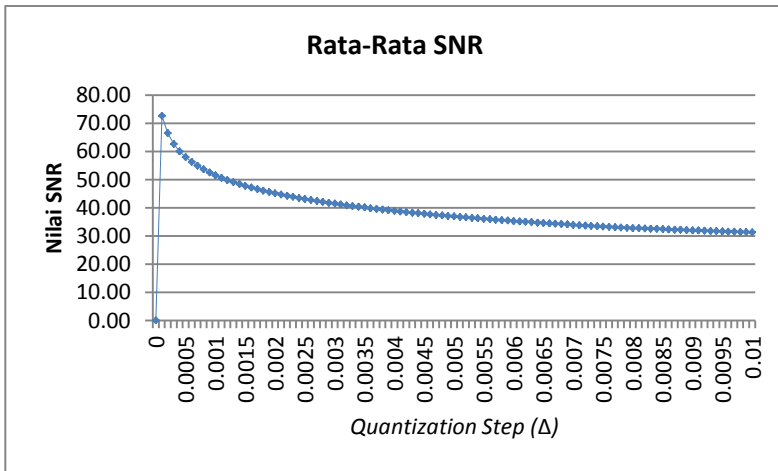
### 5.5.2 Skenario Uji Coba 2

Skenario uji coba 2 adalah perhitungan SNR, akurasi, dan *running time* menggunakan metode *Discrete Cosine Transform* (DCT) untuk mencari *quantization step* ( $\Delta$ ) yang optimal. *Key* dan parameter pengujian pada uji coba ini sama dengan yang digunakan pada skenario uji coba 1.



**Gambar 5. 25 Grafik Rata-Rata Akurasi Gambar pada Uji Coba 2 (detil pada Tabel A. 9 hingga Tabel A. 12)**

Hasil percobaan menunjukkan bahwa rentang  $\Delta$  yang bagus untuk melihat pola rata-rata akurasi gambar dan rata-rata SNR audio adalah antara 0 hingga  $1 \times 10^{-2}$  dengan tahap peningkatan  $1 \times 10^{-4}$ . Detil dari Gambar 5. 25 dan Gambar 5. 26 dapat dilihat pada Tabel A. 9 hingga Tabel A. 12. Grafik rata-rata akurasi gambar hasil ekstraksi menunjukkan bahwa dengan nilai  $\Delta$  lebih besar sama dengan  $1 \times 10^{-4}$  (lebih tepatnya  $0,86 \times 10^{-4}$  dapat dilihat pada Tabel A. 16) gambar dapat diekstraksi dengan akurasi sebesar 100%. Sama halnya dengan uji coba 1, ini berarti dengan nilai  $\Delta$  tersebut, proses normalisasi sinyal audio ke dalam resolusi 16-bit tidak berpengaruh secara signifikan pada domain frekuensi audio. Untuk melihat nilai  $\Delta$  yang menghasilkan sinyal audio yang terdistorsi secara signifikan akibat dari proses normalisasi yang mengakibatkan berkurangnya akurasi gambar hasil ekstraksi dapat dilihat pada Gambar A. 3 dan Gambar A. 4.



**Gambar 5. 26 Grafik Rata-Rata SNR Audio pada Uji Coba 2 (detil pada Tabel A. 9 hingga Tabel A. 12)**

Gambar 5. 26 menunjukkan pola rata-rata SNR pada sinyal audio yang dihasilkan pada uji coba 2. Dapat dilihat bahwa pola rata-rata SNR adalah semakin menurun (kualitas audio

memburuk) dengan nilai  $\Delta$  yang semakin besar dan perkecualian untuk nilai  $\Delta = 0$  karena domain frekuensi tidak dapat dibagi dengan 0 pada proses kuantisasi, sehingga sinyal audio keluaran tidak dapat diukur SNRnya. Nilai  $\Delta$  paling kecil yang dapat menghasilkan sinyal audio dengan SNR di atas 35 dB adalah  $1 \times 10^{-4}$  dengan rata-rata SNR 72,67 dB, sedangkan nilai  $\Delta$  paling besar yang dapat menghasilkan sinyal audio dengan SNR di atas 35 dB adalah  $62 \times 10^{-4}$  dengan rata-rata SNR 35,01 dB, namun pada Tabel A. 22 dan Tabel A. 23 terlihat bahwa terdapat 2 buah gambar yang menghasilkan sinyal audio dengan SNR di bawah 35 dB, sehingga nilai  $\Delta$  paling besar yang lebih aman digunakan agar memenuhi syarat SNR adalah  $61 \times 10^{-4}$  dengan rata-rata SNR 35,15 dB.

Pada uji coba ini, *running time* tidak terpengaruh oleh variabel manapun, sehingga *running time* dari uji coba ini sepenuhnya bergantung pada metode DCT yang memiliki kompleksitas waktu sebesar  $O(N \log N)$ , dimana pada uji coba ini rata-rata *running time*-nya adalah 0,04 detik.

Sama halnya dengan uji coba 1, karena pola SNR tidak menunjukkan anomali pada rentang nilai  $\Delta$  yang menghasilkan akurasi gambar 100%, maka nilai  $\Delta$  terkecil dan terbesar ( $1 \times 10^{-4}$  dan  $61 \times 10^{-4}$ ) akan diambil sebagai kandidat nilai  $\Delta$  optimal sesuai dengan parameter pemilihan nilai  $\Delta$  yang terakhir yaitu ketahanan sinyal audio terhadap *noise*.

### 5.5.3 Skenario Uji Coba 3

Pada skenario uji coba 3, akan dilakukan uji ketahanan sinyal audio-stego yang dihasilkan dari Uji Coba 1 (*Walsh-Hadamard Transform - WHT*) dan Uji Coba 2 (*Discrete Cosine Transform - DCT*) terhadap serangan *Additive White Gaussian Noise* (AWGN). Hasil dari uji coba 1 dan 2 menunjukkan bahwa proses kuantisasi berhasil mengekstraksi gambar dengan akurasi 100% pada rentang nilai  $\Delta$  tertentu, ini berarti kedua transformasi fourier berhasil merekonstruksi sinyal audio tanpa distorsi yang

signifikan pada domain frekuensinya akibat proses normalisasinya, sehingga penulis hanya perlu memilih *quantization step* ( $\Delta$ ) yang memenuhi syarat SNR dan memiliki ketahanan lebih tinggi terhadap serangan AWGN (akurasi gambar memiliki prioritas lebih tinggi daripada kualitas suara). Idealnya, semakin besar nilai  $\Delta$  maka semakin besar ruang toleransi *error* pada domain frekuensi, sehingga ketahanan sinyal tersebut terhadap *noise* semakin besar. Namun untuk membuktikan hal ini, akan diuji coba nilai  $\Delta$  yang terbesar dan yang terkecil (karena nilai  $\Delta$  kecil dapat menghasilkan SNR yang tinggi atau kualitas suara yang bagus) yang didapat dari hasil uji coba 1 dan 2. Varians AWGN dan parameter ketahanan sinyal audio yang diterima pada uji coba ini mengacu pada penelitian sebelumnya [1], yaitu sinyal audio yang diberi AWGN dengan varians sebesar 484 (0,74% dari rentang sinyal audio) harus bisa menghasilkan gambar dengan akurasi lebih besar sama dengan 99,41% (toleransi kesalahan piksel gambar adalah 387 buah).

Hasil uji coba pada Tabel 5. 1 dan Tabel 5. 2 menunjukkan bahwa kedua nilai  $\Delta$  terbesar dapat menghasilkan sinyal audio tahan terhadap *noise* karena gambar hasil ekstraksi dari sinyal audio yang telah diberi AWGN tersebut memenuhi syarat akurasi gambar (kesalahan piksel di bawah 387 buah). Sedangkan pada Tabel 5. 3 dan Tabel 5. 4 menunjukkan bahwa kedua nilai  $\Delta$  terkecil menghasilkan sinyal audio yang tidak tahan terhadap *noise* karena gambar hasil ekstraksi dari sinyal audio yang telah diberi AWGN tidak memenuhi syarat akurasi gambar (rata-rata akurasi gambar hasil ekstraksi adalah 50,04% untuk metode DCT dan 50,01% untuk metode WHT). Ini berarti parameter pemilihan nilai  $\Delta$  yang baik adalah benar bahwa semakin besar nilai  $\Delta$ , maka semakin tinggi ketahanan sinyal tersebut terhadap *noise*.

Maka dari hasil uji coba ini, didapatkan bahwa nilai  $\Delta$  yang optimal untuk aplikasi ini berdasarkan hasil uji coba 1, 2, dan 3 adalah  $61 \times 10^{-4}$  untuk metode DCT dan  $31 \times 10^{-6}$  untuk metode WHT.



**Tabel 5. 1 Hasil Skenario Uji Coba 3 untuk Metode DCT dengan  $\Delta=61 \times 10^{-4}$  ( $\Delta$  terbesar)**

<b>Gambar</b>	<b>Akurasi</b>	<b>Kesalahan piksel</b>
1	0,9999695	2
2	0,9999847	1
3	0,9999847	1
4	0,9999847	1
5	0,9999847	1
6	0,9999847	1
7	0,9999847	1
8	0,9999847	1
9	0,9999695	2
10	0,9999847	1
<b>Mean :</b>	<b>0,9999817</b>	<b>1.2</b>

**Tabel 5. 2 Hasil Skenario Uji Coba 3 untuk Metode WHT dengan  $\Delta=31 \times 10^{-6}$  ( $\Delta$  terbesar)**

<b>Gambar</b>	<b>Akurasi</b>	<b>Kesalahan piksel</b>
1	1	0
2	1	0
3	1	0
4	1	0
5	1	0
6	1	0
7	1	0
8	1	0
9	1	0
10	1	0
<b>Mean :</b>	<b>1</b>	<b>0</b>

**Tabel 5. 3 Hasil Skenario Uji Coba 3 untuk Metode DCT dengan  $\Delta=1 \times 10^{-4}$  ( $\Delta$  terkecil)**

<b>Gambar</b>	<b>Akurasi</b>	<b>Kesalahan piksel</b>
<b>1</b>	0,5001068	32761
<b>2</b>	0,4968719	32973
<b>3</b>	0,5001221	32760
<b>4</b>	0,5009003	32709
<b>5</b>	0,4982605	32882
<b>6</b>	0,5020752	32632
<b>7</b>	0,5011597	32692
<b>8</b>	0,5021667	32626
<b>9</b>	0,4977112	32918
<b>10</b>	0,5047760	32455
<b>Mean :</b>	<b>0,5004150</b>	<b>32740.8</b>

**Tabel 5. 4 Hasil Skenario Uji Coba 3 untuk Metode WHT dengan  $\Delta=1 \times 10^{-6}$  ( $\Delta$  terkecil)**

<b>Gambar</b>	<b>Akurasi</b>	<b>Kesalahan piksel</b>
<b>1</b>	0,5008240	32714
<b>2</b>	0,5021362	32628
<b>3</b>	0,5031586	32561
<b>4</b>	0,4998474	32778
<b>5</b>	0,5018005	32650
<b>6</b>	0,5001984	32755
<b>7</b>	0,4987793	32848
<b>8</b>	0,4998932	32775
<b>9</b>	0,4971313	32956
<b>10</b>	0,4980621	32895
<b>Mean :</b>	<b>0,5001831</b>	<b>32756</b>

## 5.6 Analisis Hasil Uji Coba

Parameter pemilihan nilai *quantization step* ( $\Delta$ ) ada tiga, yaitu sinyal audio yang dihasilkan harus memiliki rasio *signal-to-noise ratio* (SNR) di atas 35 dB bila dibandingkan dengan sinyal audio yang asli, tahan terhadap proses normalisasi ke dalam rentang audio dengan resolusi 16-bit (akurasi gambar hasil ekstraksi tetap 100% meskipun sinyal audio telah bergeser akibat proses normalisasi), dan yang terakhir adalah sinyal audio yang dihasilkan harus tahan terhadap *noise* (akurasi gambar hasil ekstraksi di atas 99,41% setelah sinyal audio diberi *Additive White Gaussian Noise* – AWGN dengan varians sebesar 484).

Berdasarkan hasil uji coba 1 dan 2, didapatkan suatu rentang nilai  $\Delta$  yang dapat menghasilkan sinyal audio yang memiliki nilai SNR lebih besar dari 35 dB dan akurasi hasil ekstraksi gambar dari sinyal audio tersebut adalah 100% pada kedua transformasi fourier ( $1 \times 10^{-6}$  hingga  $31 \times 10^{-6}$  untuk metode *Walsh Hadamard Transform* – WHT dan  $1 \times 10^{-4}$  hingga  $61 \times 10^{-4}$  untuk metode *Discrete Cosine Transform* - DCT). Penulis dapat berasumsi bahwa tidak ada anomali pada nilai  $\Delta$  di luar rentang uji coba karena pola yang didapat pada rata-rata akurasi gambar dan rata-rata SNR tidak menunjukkan hal tersebut. Maka pemilihan nilai  $\Delta$  bergantung pada parameter terakhir yaitu ketahanan sinyal audio terhadap *noise*. Nilai  $\Delta$  yang diuji cobakan pada uji coba 3 adalah nilai  $\Delta$  terkecil dan terbesar dari rentang nilai  $\Delta$  yang didapat dari hasil uji coba 1 dan 2. Hasil dari uji coba 3 menunjukkan bahwa akurasi gambar hasil ekstraksi dari sinyal audio dengan nilai  $\Delta$  tekecil menurun secara drastis untuk kedua metode transformasi fourier dibandingkan dengan sinyal audio dengan nilai  $\Delta$  terbesar. Dari hasil uji coba ini, didapatkan bahwa nilai  $\Delta$  yang optimal adalah yang terbesar, yaitu  $61 \times 10^{-4}$  untuk metode DCT dan  $31 \times 10^{-6}$  untuk metode WHT. Hasil selengkapnya mengenai SNR sinyal audio hasil penyembunyian, akurasi gambar hasil ekstraksi, dan *running time* aplikasi dengan nilai  $\Delta$  yang terpilih dapat dilihat pada Tabel A. 17 hingga Tabel

A. 20 untuk metode WHT dan Tabel A. 21 hingga Tabel A. 24 untuk metode DCT.

Kemudian diperlukan analisis untuk membandingkan kedua metode transformasi fourier (WHT dan DCT). Ada empat parameter perbandingan kedua metode transformasi fourier, yaitu nilai SNR sinyal audio yang dihasilkan, akurasi gambar yang diekstraksi, ketahanan terhadap *noise*, dan *running time*.

Menggunakan nilai  $\Delta$  yang optimal dari hasil uji coba, kedua transformasi fourier menghasilkan sinyal audio dengan nilai SNR yang berbeda. Pada metode WHT dengan nilai  $\Delta = 31 \times 10^{-6}$ , rata-rata SNR yang dihasilkan adalah 35,24 dB. Sedangkan metode DCT dengan nilai  $\Delta = 61 \times 10^{-4}$ , rata-rata SNR yang dihasilkan adalah 35,15 dB. Bila dilihat dari perbandingan kedua nilai SNR ini, sinyal audio yang dihasilkan menggunakan metode WHT memiliki kualitas lebih bagus dibandingkan dengan metode DCT. Namun hal ini dapat diabaikan karena penelitian lebih lanjut mengenai pemilihan nilai  $\Delta$  dapat menghasilkan sinyal audio dengan nilai SNR yang kurang lebih sama dengan kedua metode transformasi fourier, misalnya memilih nilai  $\Delta$  beberapa angka dibelakang koma lebih dalam dibandingkan nilai  $\Delta$  yang optimal namun tetap memenuhi seluruh parameter pemilihan nilai  $\Delta$ . Hal ini dapat dimungkinkan karena nilai  $\Delta$  yang menghasilkan sinyal audio dengan SNR mendekati 35 dB tetap memenuhi syarat akurasi gambar dengan melihat pola rata-rata akurasi gambar hasil ekstraksi tersebut pada Gambar 5. 23 dan Gambar 5. 25. Jadi untuk parameter perbandingan pertama, meskipun sinyal audio hasil penyembunyian dengan metode WHT memiliki kualitas lebih bagus dibandingkan metode DCT, keduanya memenuhi syarat SNR dengan sinyal audio yang dihasilkan memiliki rata-rata SNR 35,24 dB untuk metode WHT dan 35,15 dB untuk metode DCT.

Kemudian akurasi gambar yang diekstraksi dari sinyal audio hasil penyembunyian menggunakan metode WHT maupun DCT dengan nilai  $\Delta$  yang optimal adalah 100%. Sehingga dengan

parameter perbandingan kedua, kedua metode transformasi fourier tidak dapat dibandingkan.

Untuk parameter perbandingan ketiga, yaitu ketahanan sinyal audio terhadap *noise* (AWGN) pada nilai  $\Delta$  yang optimal, hasil rata-rata akurasi gambar hasil ekstraksi menggunakan metode DCT sedikit lebih rendah dibandingkan dengan metode WHT (lihat Tabel 5. 1 dan Tabel 5. 2), dimana dengan menggunakan metode DCT rata-rata akurasi gambarnya adalah 99,99% (1,2 buah kesalahan piksel), sedangkan metode WHT adalah 100% (0 buah kesalahan piksel). Sama halnya dengan parameter perbandingan pertama, hal ini juga dapat diabaikan karena faktor nilai  $\Delta$  lebih besar pada metode DCT dapat menghasilkan gambar tanpa kesalahan piksel sama sekali sama dengan hasil ketahanan sinyal menggunakan metode WHT, serta kedua nilai akurasi ini memenuhi syarat akurasi gambar yang ditentukan pada uji coba 3 [1], yaitu toleransi kesalahan piksel adalah 387 buah. Karena kedua transformasi fourier lolos ketiga parameter perbandingan pertama, maka penulis beralih ke parameter perbandingan yang terakhir yaitu *running time*.

Selisih *running time* kedua metode transformasi fourier cukup signifikan, dimana DCT memiliki *running time* sebesar 0,04 detik dan WHT memiliki *running time* sebesar 1,77 detik. Ini disebabkan karena proses WHT memiliki kompleksitas waktu lebih tinggi daripada proses DCT. Rangkuman dari hasil analisis perbandingan kedua transformasi fourier dapat dilihat pada Tabel 5. 5.

**Tabel 5. 5 Tabel Perbandingan WHT dan DCT**

	<b>WHT</b>	<b>DCT</b>
<b>SNR (&gt; 35 dB)</b>	35,24 dB	35,15 dB
<b>Akurasi Gambar (=100%)</b>	100%	100%
<b>Ketahanan Terhadap Noise (Akurasi Gambar &gt; 99,41%)</b>	99,99%	100%
<b>Running Time</b>	1,77	0,04

Karena perbedaan *running time* kedua metode transformasi fourier cukup signifikan, parameter *running time* dapat digunakan sebagai parameter penentu dari kedua metode transformasi fourier. Dan dari hasil perbandingan ini dapat disimpulkan bahwa metode DCT memiliki performa yang lebih baik pada aplikasi ini dengan *quantization step* =  $61 \times 10^{-4}$ .

## BAB VI KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

### 6.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Metode *Chaotic Encryption* dan *Discrete Fourier Transform* dapat digunakan untuk menyembunyikan gambar sidik jari ke dalam sinyal audio beserta ekstraksinya.
2. Dengan *quantization step* bernilai  $31 \times 10^{-6}$  untuk metode *Walsh-Hadamard Transform* (WHT) dan  $61 \times 10^{-4}$  untuk metode *Discrete Cosine Transform* (DCT), proses penyembunyian sidik jari dapat menghasilkan suatu sinyal audio yang perbedaannya terhadap sinyal audio yang asli berada di bawah batas sensitivitas pendengaran manusia, akurasi gambar yang tinggi pada saat diekstraksi, dan tahan terhadap *noise*.
3. Metode DCT memiliki performa lebih bagus dengan selisih *running time* 1,73 detik lebih cepat daripada metode WHT, dimana rata-rata *running time* metode DCT adalah 0,04 detik dan rata-rata *running time* metode WHT adalah 1,77 detik.

### 6.2 Saran

Saran yang diberikan untuk pengembangan aplikasi ini adalah:

1. Bila aplikasi ini digunakan untuk tujuan autentikasi, lebih baik menggunakan metode *hash* daripada metode enkripsi.

2. Dibutuhkan penelitian lebih lanjut mengenai *quantization step* untuk semua kemungkinan permutasi gambar biner.



## DAFTAR PUSTAKA

- [1] Ling Xie, Jiashu Zhang Muhammad Khurram Khan, "Chaos and NDFT-based spread spectrum concealing of fingerprint-biometric data into audio signals," 2009.
- [2] Prof. M.A. Artemi, R.M. Suliman, M.M. AL Rjebi Prof. W.S. Besbas, "Content Based Face Image Retrieval in Walsh Hadamard Transform (WHT) Domain," 2012.
- [3] Member, IEEE, Karen Panetta, Fellow, IEEE, and Sos Agaian, Senior Member, IEEE Yicong Zhou, "Image Encryption Using Discrete Parametric Cosine Transform".
- [4] Ambara. (2012) [Online]. <https://am8ara.wordpress.com/2012/08/03/apa-itu-biometrik/>
- [5] wiseGEEK Team. [Online]. <http://www.wisegeek.com/what-are-the-advantages-of-fingerprint-identification.htm>
- [6] Agus Purnama. [Online]. <http://elektronika-dasar.web.id/teori-elektronika/sinyal-audio-gelombang-suara/>
- [7] The Fractal Foundation. [Online]. <http://fractalfoundation.org/resources/what-is-chaos-theory/>
- [8] Engineering Productivity Tools Ltd. [Online]. [http://www.engineeringproductivitytools.com/stuff/T0001/P\\_T01.HTM](http://www.engineeringproductivitytools.com/stuff/T0001/P_T01.HTM)
- [9] A.A.I. Perrera, M.A.P. Chamikara A.A.C.A. Jayathilake. (2013, January) [Online]. [http://www.academia.edu/3322481/Discrete\\_Walsh-Hadamard\\_transform\\_in\\_signal\\_processing](http://www.academia.edu/3322481/Discrete_Walsh-Hadamard_transform_in_signal_processing)
- [10] Margaret Rouse. [Online]. <http://searchnetworking.techtarget.com/definition/signal-to-noise-ratio>
- [11] Siddhant Raman. (2014, November) [Online]. <http://www.quora.com/What-is-an-Additive-White->

[Gaussian-Noise-channel-in-terms-of-digital-communication](#)

- [12] NASA Ames Research Center Andrew B. Watson, "Image Compression Using the Discrete Cosine Transform," 1994.
- [13] Athanasios Nikolaidis, Nikos Nikolaidis, Vassilios Solachidis, Sofia Tsekeridou and Ioannis Pitas Anastasios Tefas, "Performance Analysis of Correlation-based Watermarking Schemes employing Markov Chaotic Sequences".
- [14] PhD Benjamin Jacoby, "Walsh Function : A Digital Fourier Series".
- [15] Ruye Wang. (2013, Oct.) [Online]. <http://fourier.eng.hmc.edu/e161/lectures/wht/node2.html>
- [16] VU University Amsterdam. [Online]. [https://www.nbtwiki.net/doku.php?id=tutorial:the\\_discrete\\_fourier\\_transformation\\_dft#.VYDJkZUzPw](https://www.nbtwiki.net/doku.php?id=tutorial:the_discrete_fourier_transformation_dft#.VYDJkZUzPw)
- [17] National Programme on Technology Learning. nptel.ac.in/. [Online]. [http://nptel.ac.in/courses/117104069/chapter\\_7/7\\_26.html](http://nptel.ac.in/courses/117104069/chapter_7/7_26.html)
- [18] Dave Marshall. (2001, Apr.) [Online]. <http://www.cs.cf.ac.uk/Dave/Multimedia/node231.html>

## LAMPIRAN

*Tabel A. 1 Tabel Detil Hasil Uji Coba 1(1)*

$\Delta$ (e-6)	Mean ACC	Kesalahan Pixel	Mean SNR
0	0.4996399	32791.6	0
1	1	0	65.21
2	1	0	59.21
3	1	0	55.68
4	1	0	53.20
5	1	0	51.24
6	1	0	49.66
7	1	0	48.31
8	1	0	47.14
9	1	0	46.11
10	1	0	45.19
11	1	0	44.37
12	1	0	43.59
13	1	0	42.87
14	1	0	42.22
15	1	0	41.63
16	1	0	41.06
17	1	0	40.53
18	1	0	40.02
19	1	0	39.55
20	1	0	39.12
21	1	0	38.67
22	1	0	38.26
23	1	0	37.87
24	1	0	37.50
25	1	0	37.14

*Tabel A. 2 Tabel Detil Hasil Uji Coba 1(2)*

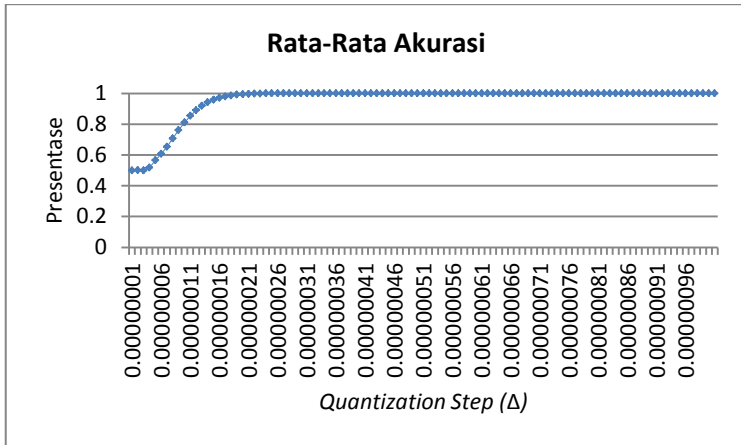
$\Delta$ (e-6)	Mean ACC	Kesalahan Piksel	Mean SNR
26	1	0	36.80
27	1	0	36.47
28	1	0	36.15
29	1	0	35.82
30	1	0	35.52
31	1	0	35.23
32	1	0	34.95
33	1	0	34.68
34	1	0	34.41
35	1	0	34.16
36	1	0	33.91
37	1	0	33.67
38	1	0	33.43
39	1	0	33.20
40	1	0	32.97
41	1	0	32.74
42	1	0	32.55
43	1	0	32.34
44	1	0	32.13
45	1	0	31.93
46	1	0	31.74
47	1	0	31.53
48	1	0	31.35
49	1	0	31.19
50	1	0	31.00

**Tabel A. 3 Tabel Detil Hasil Uji Coba 1(3)**

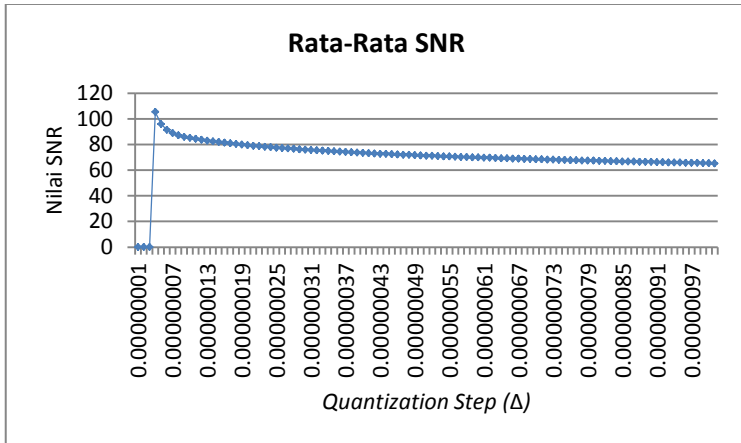
$\Delta$ (e-6)	Mean ACC	Kesalahan Piksel	Mean SNR
51	1	0	30.82
52	1	0	30.63
53	1	0	30.46
54	1	0	30.29
55	1	0	30.14
56	1	0	29.98
57	1	0	29.82
58	1	0	29.67
59	1	0	29.51
60	1	0	29.37
61	1	0	29.22
62	1	0	29.07
63	1	0	28.92
64	1	0	28.77
65	1	0	28.63
66	1	0	28.50
67	1	0	28.36
68	1	0	28.24
69	1	0	28.10
70	1	0	27.97
71	1	0	27.84
72	1	0	27.73
73	1	0	27.60
74	1	0	27.48
75	1	0	27.36

*Tabel A. 4 Tabel Detil Hasil Uji Coba 1(4)*

$\Delta$ (e-6)	Mean ACC	Kesalahan Pikel	Mean SNR
76	1	0	27.23
77	1	0	27.12
78	1	0	27.01
79	1	0	26.90
80	1	0	26.79
81	1	0	26.67
82	1	0	26.56
83	1	0	26.45
84	1	0	26.34
85	1	0	26.24
86	1	0	26.13
87	1	0	26.02
88	1	0	25.92
89	1	0	25.82
90	1	0	25.72
91	1	0	25.62
92	1	0	25.53
93	1	0	25.43
94	1	0	25.33
95	1	0	25.23
96	1	0	25.14
97	1	0	25.04
98	1	0	24.95
99	1	0	24.86
100	1	0	24.77



**Gambar A. 1** Grafik Rata-Rata Akurasi Gambar pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-6}$  pada Uji Coba 1 (detil pada Tabel A. 5 hingga Tabel A. 8)



**Gambar A. 2** Grafik Rata-Rata SNR pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-6}$  pada Uji Coba 1 (detil pada Tabel A. 5 hingga Tabel A. 8)

**Tabel A. 5 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-6}$  (1)**

$\Delta$ (e-6)	Mean ACC	Kesalahan Piksel	Mean SNR
<b>0</b>	0.4996399	32791.6	0
<b>0.01</b>	0.5016846	32657.6	Inf
<b>0.02</b>	0.4994583	32803.5	Inf
<b>0.03</b>	0.5179291	31593	105.18
<b>0.04</b>	0.5658295	28453.8	95.94
<b>0.05</b>	0.6076828	25710.9	91.38
<b>0.06</b>	0.6520844	22801	88.82
<b>0.07</b>	0.7061111	19260.3	87.15
<b>0.08</b>	0.7613541	15639.9	85.94
<b>0.09</b>	0.8111084	12379.2	85.05
<b>0.1</b>	0.8543945	9542.4	84.28
<b>0.11</b>	0.8891891	7262.1	83.61
<b>0.12</b>	0.918779	5322.9	82.96
<b>0.13</b>	0.9409927	3867.1	82.35
<b>0.14</b>	0.9581345	2743.7	81.81
<b>0.15</b>	0.970784	1914.7	81.26
<b>0.16</b>	0.9799026	1317.1	80.77
<b>0.17</b>	0.9865997	878.2	80.25
<b>0.18</b>	0.9910629	585.7	79.82
<b>0.19</b>	0.9941269	384.9	79.38
<b>0.2</b>	0.9963684	238	78.97
<b>0.21</b>	0.9977264	149	78.56
<b>0.22</b>	0.9986115	91	78.16
<b>0.23</b>	0.9991943	52.8	77.81
<b>0.24</b>	0.9995285	30.9	77.45
<b>0.25</b>	0.9997086	19.1	77.12



**Tabel A. 6 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-6}$  (2)**

$\Delta$ (e-6)	Mean ACC	Kesalahan Pixel	Mean SNR
<b>0.26</b>	0.999855	9.5	76.77
<b>0.27</b>	0.9999161	5.5	76.48
<b>0.28</b>	0.9999588	2.7	76.16
<b>0.29</b>	0.9999756	1.6	75.86
<b>0.3</b>	0.9999924	0.5	75.60
<b>0.31</b>	0.9999939	0.4	75.31
<b>0.32</b>	0.9999939	0.4	75.04
<b>0.33</b>	0.9999969	0.2	74.79
<b>0.34</b>	0.9999985	0.1	74.51
<b>0.35</b>	1	0	74.28
<b>0.36</b>	1	0	74.03
<b>0.37</b>	1	0	73.81
<b>0.38</b>	1	0	73.57
<b>0.39</b>	1	0	73.36
<b>0.4</b>	1	0	73.12
<b>0.41</b>	1	0	72.91
<b>0.42</b>	1	0	72.72
<b>0.43</b>	1	0	72.49
<b>0.44</b>	1	0	72.31
<b>0.45</b>	1	0	72.12
<b>0.46</b>	1	0	71.93
<b>0.47</b>	1	0	71.75
<b>0.48</b>	1	0	71.59
<b>0.49</b>	1	0	71.40
<b>0.5</b>	1	0	71.22

**Tabel A. 7 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-6}$  (3)**

$\Delta$ (e-6)	Mean ACC	Kesalahan Piksel	Mean SNR
<b>0.51</b>	1	0	71.05
<b>0.52</b>	1	0	70.88
<b>0.53</b>	1	0	70.73
<b>0.54</b>	1	0	70.58
<b>0.55</b>	1	0	70.40
<b>0.56</b>	1	0	70.23
<b>0.57</b>	1	0	70.09
<b>0.58</b>	1	0	69.94
<b>0.59</b>	1	0	69.80
<b>0.6</b>	1	0	69.67
<b>0.61</b>	1	0	69.52
<b>0.62</b>	1	0	69.37
<b>0.63</b>	1	0	69.23
<b>0.64</b>	1	0	69.11
<b>0.65</b>	1	0	68.95
<b>0.66</b>	1	0	68.82
<b>0.67</b>	1	0	68.71
<b>0.68</b>	1	0	68.58
<b>0.69</b>	1	0	68.45
<b>0.7</b>	1	0	68.33
<b>0.71</b>	1	0	68.19
<b>0.72</b>	1	0	68.09
<b>0.73</b>	1	0	67.94
<b>0.74</b>	1	0	67.83
<b>0.75</b>	1	0	67.73

**Tabel A. 8 Tabel Detil Hasil Uji Coba 1 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-6}$  (4)**

$\Delta$ (e-6)	Mean ACC	Kesalahan Pixel	Mean SNR
<b>0.76</b>	1	0	67.60
<b>0.77</b>	1	0	67.49
<b>0.78</b>	1	0	67.36
<b>0.79</b>	1	0	67.28
<b>0.8</b>	1	0	67.18
<b>0.81</b>	1	0	67.07
<b>0.82</b>	1	0	66.97
<b>0.83</b>	1	0	66.82
<b>0.84</b>	1	0	66.73
<b>0.85</b>	1	0	66.63
<b>0.86</b>	1	0	66.55
<b>0.87</b>	1	0	66.44
<b>0.88</b>	1	0	66.31
<b>0.89</b>	1	0	66.26
<b>0.9</b>	1	0	66.16
<b>0.91</b>	1	0	66.05
<b>0.92</b>	1	0	65.94
<b>0.93</b>	1	0	65.87
<b>0.94</b>	1	0	65.76
<b>0.95</b>	1	0	65.69
<b>0.96</b>	1	0	65.61
<b>0.97</b>	1	0	65.51
<b>0.98</b>	1	0	65.42
<b>0.99</b>	1	0	65.31
<b>1</b>	1	0	65.21

**Tabel A. 9 Tabel Detil Hasil Uji Coba 2(1)**

$\Delta$ (e-4)	Mean ACC	Kesalahan Piksel	Mean SNR
0	0.4996399	32791.6	0.00
1	1	0	72.67
2	1	0	66.44
3	1	0	62.68
4	1	0	60.02
5	1	0	57.97
6	1	0	56.29
7	1	0	54.85
8	1	0	53.60
9	1	0	52.50
10	1	0	51.50
11	1	0	50.61
12	1	0	49.79
13	1	0	49.06
14	1	0	48.37
15	1	0	47.73
16	1	0	47.15
17	1	0	46.59
18	1	0	46.07
19	1	0	45.57
20	1	0	45.12
21	1	0	44.67
22	1	0	44.25
23	1	0	43.85
24	1	0	43.47
25	1	0	43.09

*Tabel A. 10 Tabel Detil Hasil Uji Coba 2(2)*

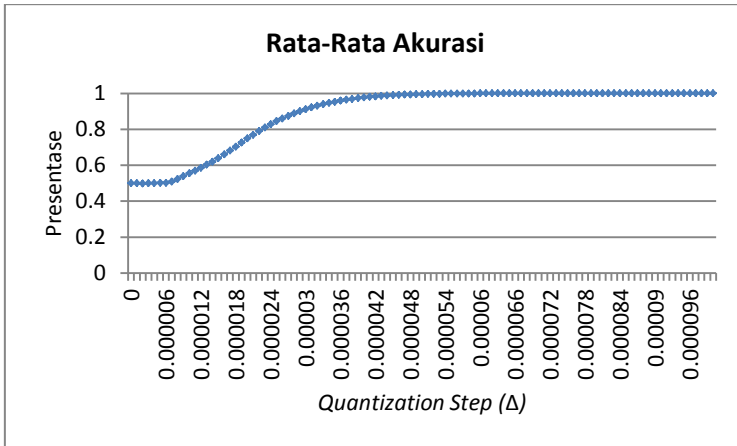
$\Delta$ (e-4)	Mean ACC	Kesalahan Piksel	Mean SNR
26	1	0	42.75
27	1	0	42.41
28	1	0	42.08
29	1	0	41.77
30	1	0	41.46
31	1	0	41.16
32	1	0	40.87
33	1	0	40.60
34	1	0	40.34
35	1	0	40.08
36	1	0	39.83
37	1	0	39.58
38	1	0	39.35
39	1	0	39.12
40	1	0	38.89
41	1	0	38.67
42	1	0	38.47
43	1	0	38.25
44	1	0	38.05
45	1	0	37.85
46	1	0	37.64
47	1	0	37.46
48	1	0	37.27
49	1	0	37.09
50	1	0	36.91

*Tabel A. 11 Tabel Detil Hasil Uji Coba 2(3)*

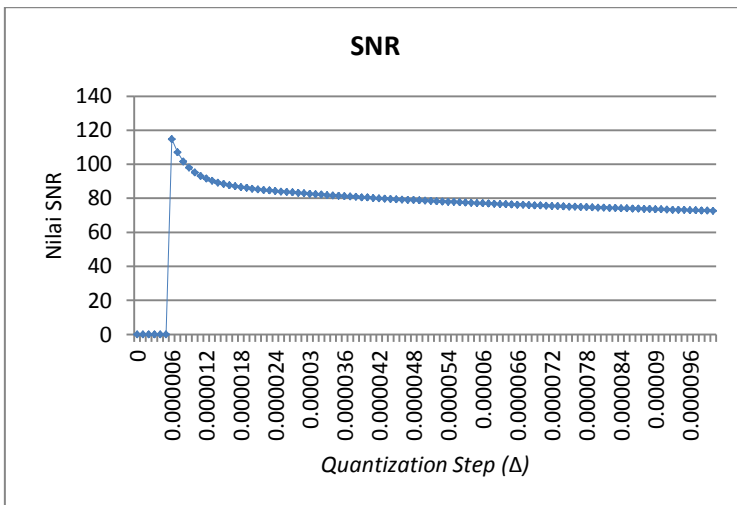
$\Delta$ (e-4)	Mean ACC	Kesalahan Pikel	Mean SNR
51	1	0	36.73
52	1	0	36.56
53	1	0	36.40
54	1	0	36.23
55	1	0	36.06
56	1	0	35.91
57	1	0	35.75
58	1	0	35.59
59	1	0	35.44
60	1	0	35.29
61	1	0	35.15
62	1	0	35.01
63	1	0	34.87
64	1	0	34.73
65	1	0	34.59
66	1	0	34.45
67	1	0	34.32
68	1	0	34.19
69	1	0	34.06
70	1	0	33.93
71	1	0	33.81
72	1	0	33.69
73	1	0	33.56
74	1	0	33.45
75	1	0	33.33

*Tabel A. 12 Tabel Detil Hasil Uji Coba 2(4)*

$\Delta$ (e-4)	Mean ACC	Kesalahan Piksel	Mean SNR
76	1	0	33.22
77	1	0	33.10
78	1	0	32.98
79	1	0	32.89
80	1	0	32.80
81	1	0	32.72
82	1	0	32.64
83	1	0	32.55
84	1	0	32.48
85	1	0	32.39
86	1	0	32.31
87	1	0	32.24
88	1	0	32.15
89	1	0	32.08
90	1	0	31.99
91	1	0	31.91
92	1	0	31.83
93	1	0	31.75
94	1	0	31.68
95	1	0	31.61
96	1	0	31.53
97	1	0	31.46
98	1	0	31.40
99	1	0	31.34
100	1	0	31.29



**Gambar A. 3 Grafik Rata-Rata Akurasi Gambar pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-4}$  pada Uji Coba 2 (detil pada Tabel A. 13 hingga Tabel A. 16)**



**Gambar A. 4 Grafik Rata-Rata SNR pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-4}$  pada Uji Coba 2 (detil pada Tabel A. 13 hingga Tabel A. 16)**



**Tabel A. 13 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-4}$  (1)**

$\Delta$ (e-4)	Mean ACC	Kesalahan Pixel	Mean SNR
<b>0.00</b>	0.499639893	32791.6	0
<b>0.01</b>	0.499343872	32811	Inf
<b>0.02</b>	0.498632813	32857.6	Inf
<b>0.03</b>	0.499807739	32780.6	Inf
<b>0.04</b>	0.499066162	32829.2	Inf
<b>0.05</b>	0.50110321	32695.7	Inf
<b>0.06</b>	0.501295471	32683.1	114.87
<b>0.07</b>	0.509326172	32156.8	107.00
<b>0.08</b>	0.522285461	31307.5	101.62
<b>0.09</b>	0.538362122	30253.9	97.98
<b>0.10</b>	0.555781555	29112.3	95.26
<b>0.11</b>	0.569929504	28185.1	93.18
<b>0.12</b>	0.585995483	27132.2	91.62
<b>0.13</b>	0.602511597	26049.8	90.37
<b>0.14</b>	0.619799805	24916.8	89.27
<b>0.15</b>	0.63936615	23634.5	88.42
<b>0.16</b>	0.660066223	22277.9	87.73
<b>0.17</b>	0.681428528	20877.9	87.10
<b>0.18</b>	0.703352356	19441.1	86.61
<b>0.19</b>	0.725836182	17967.6	86.13
<b>0.20</b>	0.749095154	16443.3	85.70
<b>0.21</b>	0.768833923	15149.7	85.33
<b>0.22</b>	0.79009552	13756.3	84.96
<b>0.23</b>	0.808905029	12523.6	84.64
<b>0.24</b>	0.827882385	11279.9	84.33
<b>0.25</b>	0.845001221	10158	84.03

**Tabel A. 14 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-4}$  (2)**

$\Delta$ (e-4)	Mean ACC	Kesalahan Pixel	Mean SNR
<b>0.26</b>	0.859811401	9187.4	83.73
<b>0.27</b>	0.874559021	8220.9	83.49
<b>0.28</b>	0.887844849	7350.2	83.22
<b>0.29</b>	0.900317383	6532.8	82.94
<b>0.30</b>	0.911132813	5824	82.70
<b>0.31</b>	0.921401978	5151	82.44
<b>0.32</b>	0.930708313	4541.1	82.21
<b>0.33</b>	0.938716125	4016.3	81.96
<b>0.34</b>	0.946156311	3528.7	81.72
<b>0.35</b>	0.952867126	3088.9	81.50
<b>0.36</b>	0.959100342	2680.4	81.31
<b>0.37</b>	0.964260864	2342.2	81.08
<b>0.38</b>	0.968946838	2035.1	80.87
<b>0.39</b>	0.973216248	1755.3	80.64
<b>0.40</b>	0.976791382	1521	80.47
<b>0.41</b>	0.979937744	1314.8	80.26
<b>0.42</b>	0.982759094	1129.9	80.06
<b>0.43</b>	0.985084534	977.5	79.88
<b>0.44</b>	0.987443542	822.9	79.70
<b>0.45</b>	0.989321899	699.8	79.50
<b>0.46</b>	0.991046143	586.8	79.33
<b>0.47</b>	0.992192078	511.7	79.13
<b>0.48</b>	0.993443298	429.7	78.97
<b>0.49</b>	0.994566345	356.1	78.81
<b>0.50</b>	0.995423889	299.9	78.62

**Tabel A. 15 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-4}$  (3)**

$\Delta$ (e-4)	Mean ACC	Kesalahan Piksel	Mean SNR
<b>0.51</b>	0.9962677	244.6	78.45
<b>0.52</b>	0.996855164	206.1	78.30
<b>0.53</b>	0.997322083	175.5	78.14
<b>0.54</b>	0.997901917	137.5	77.99
<b>0.55</b>	0.998240662	115.3	77.84
<b>0.56</b>	0.998524475	96.7	77.67
<b>0.57</b>	0.99883728	76.2	77.53
<b>0.58</b>	0.999067688	61.1	77.39
<b>0.59</b>	0.999226379	50.7	77.23
<b>0.60</b>	0.999339294	43.3	77.08
<b>0.61</b>	0.999485779	33.7	76.96
<b>0.62</b>	0.999563599	28.6	76.80
<b>0.63</b>	0.999653625	22.7	76.68
<b>0.64</b>	0.999694824	20	76.54
<b>0.65</b>	0.999789429	13.8	76.42
<b>0.66</b>	0.999822998	11.6	76.28
<b>0.67</b>	0.999848938	9.9	76.15
<b>0.68</b>	0.999900818	6.5	76.01
<b>0.69</b>	0.999913025	5.7	75.89
<b>0.70</b>	0.999917603	5.4	75.77
<b>0.71</b>	0.999943542	3.7	75.65
<b>0.72</b>	0.999966431	2.2	75.53
<b>0.73</b>	0.999972534	1.8	75.42
<b>0.74</b>	0.999966431	2.2	75.30
<b>0.75</b>	0.999978638	1.4	75.18

**Tabel A. 16 Tabel Detil Hasil Uji Coba 2 pada Rentang Nilai  $\Delta = 0$  hingga  $1 \times 10^{-4}$  (4)**

$\Delta$ (e-4)	Mean ACC	Kesalahan Piksel	Mean SNR
<b>0.76</b>	0.999986267	0.9	75.06
<b>0.77</b>	0.999980164	1.3	74.95
<b>0.78</b>	0.999990845	0.6	74.82
<b>0.79</b>	0.999996948	0.2	74.72
<b>0.80</b>	0.999992371	0.5	74.62
<b>0.81</b>	0.999990845	0.6	74.51
<b>0.82</b>	0.999995422	0.3	74.40
<b>0.83</b>	0.999993896	0.4	74.29
<b>0.84</b>	0.999998474	0.1	74.19
<b>0.85</b>	0.999998474	0.1	74.09
<b>0.86</b>	1	0	73.99
<b>0.87</b>	1	0	73.89
<b>0.88</b>	1	0	73.79
<b>0.89</b>	1	0	73.71
<b>0.90</b>	1	0	73.61
<b>0.91</b>	1	0	73.51
<b>0.92</b>	1	0	73.41
<b>0.93</b>	1	0	73.30
<b>0.94</b>	1	0	73.21
<b>0.95</b>	1	0	73.13
<b>0.96</b>	1	0	73.03
<b>0.97</b>	1	0	72.94
<b>0.98</b>	1	0	72.85
<b>0.99</b>	1	0	72.75
<b>1.00</b>	1	0	72.67

**Tabel A. 17 Tabel Hasil Uji Coba Metode WHT pada Nilai  $\Delta$  di sekitar  $31 \times 10^{-6}$  (1)**

Gambar	$\Delta$ (e-6) :	26	27	28	29	30
1	ACC	1	1	1	1	1
	SNR	36.79	36.47	36.15	35.82	35.54
	TIME	1.66	1.58	1.62	1.62	1.65
2	ACC	1	1	1	1	1
	SNR	36.80	36.47	36.15	35.84	35.53
	TIME	1.80	1.63	1.62	1.64	1.63
3	ACC	1	1	1	1	1
	SNR	36.79	36.46	36.15	35.83	35.53
	TIME	1.67	1.63	1.63	1.63	2.34
4	ACC	1	1	1	1	1
	SNR	36.83	36.50	36.18	35.87	35.58
	TIME	1.62	1.64	1.60	1.63	1.64
5	ACC	1	1	1	1	1
	SNR	36.81	36.49	36.15	35.83	35.52
	TIME	1.63	1.65	1.61	1.62	1.65
6	ACC	1	1	1	1	1
	SNR	36.80	36.48	36.16	35.85	35.56
	TIME	1.66	2.46	1.65	1.64	2.36
7	ACC	1	1	1	1	1
	SNR	36.79	36.48	36.13	35.81	35.53
	TIME	2.30	1.63	2.16	1.72	1.93
8	ACC	1	1	1	1	1
	SNR	36.79	36.47	36.14	35.80	35.51
	TIME	1.62	1.58	1.63	2.16	1.64

**Tabel A. 18 Tabel Hasil Uji Coba Metode WHT pada Nilai  $\Delta$  di sekitar  $31 \times 10^{-6}$  (2)**

Gambar	$\Delta$ (e-6) :	31	32	33	34	35
1	ACC	1	1	1	1	1
	SNR	35.24	34.98	34.71	34.43	34.19
	TIME	1.67	2.35	1.64	1.64	1.62
2	ACC	1	1	1	1	1
	SNR	35.23	34.94	34.67	34.41	34.15
	TIME	1.63	1.61	2.21	1.66	1.62
3	ACC	1	1	1	1	1
	SNR	35.23	34.94	34.67	34.40	34.13
	TIME	1.63	2.43	1.62	2.46	1.62
4	ACC	1	1	1	1	1
	SNR	35.27	35.01	34.72	34.44	34.18
	TIME	1.62	1.62	1.62	1.62	1.62
5	ACC	1	1	1	1	1
	SNR	35.24	34.97	34.69	34.44	34.18
	TIME	2.22	1.63	1.62	1.64	1.64
6	ACC	1	1	1	1	1
	SNR	35.26	34.99	34.70	34.42	34.19
	TIME	1.63	2.38	1.62	1.59	2.46
7	ACC	1	1	1	1	1
	SNR	35.23	34.96	34.67	34.42	34.15
	TIME	1.61	1.94	1.64	2.01	1.61
8	ACC	1	1	1	1	1
	SNR	35.22	34.94	34.67	34.40	34.16
	TIME	2.40	1.63	1.63	1.62	2.35

**Tabel A. 19 Tabel Hasil Uji Coba Metode WHT pada Nilai  $\Delta$  di sekitar  $31 \times 10^{-6}$  (3)**

Gambar	$\Delta$ (e-6) :	26	27	28	29	30
9	ACC	1	1	1	1	1
	SNR	36.78	36.46	36.13	35.81	35.54
	TIME	2.39	1.62	1.66	1.67	1.66
10	ACC	1	1	1	1	1
	SNR	36.80	36.47	36.14	35.80	35.52
	TIME	1.62	2.39	1.63	1.62	1.62
Gambar	$\Delta$ (e-6) :	31	32	33	34	35
9	ACC	1	1	1	1	1
	SNR	35.24	34.95	34.68	34.44	34.17
	TIME	2.22	1.64	2.07	1.62	1.67
10	ACC	1	1	1	1	1
	SNR	35.21	34.95	34.67	34.42	34.18
	TIME	1.61	1.62	1.60	1.63	1.83

**Tabel A. 20 Rekap Hasil Uji Coba Metode WHT pada Tabel A. 17 hingga Tabel A. 19**

Gambar	$\Delta$ (e-6) :	26	27	28	29	30
1 - 10	Mean SNR :	36.80	36.48	36.15	35.83	35.53
Gambar	$\Delta$ (e-6) :	31	32	33	34	35
1 - 10	Mean SNR :	35.24	34.96	34.69	34.42	34.17

**Tabel A. 21 Tabel Hasil Uji Coba Metode DCT pada Nilai  $\Delta$  di sekitar  $61 \times 10^{-4}$  (1)**

Gambar	$\Delta$ (e-4) :	56	57	58	59	60
1	ACC	1	1	1	1	1
	SNR	35.90	35.74	35.59	35.43	35.29
	TIME	0.05	0.04	0.05	0.04	0.06
2	ACC	1	1	1	1	1
	SNR	35.91	35.75	35.59	35.44	35.30
	TIME	0.04	0.04	0.05	0.04	0.04
3	ACC	1	1	1	1	1
	SNR	35.91	35.75	35.59	35.44	35.29
	TIME	0.04	0.04	0.04	0.05	0.04
4	ACC	1	1	1	1	1
	SNR	35.92	35.76	35.61	35.46	35.31
	TIME	0.04	0.04	0.04	0.05	0.04
5	ACC	1	1	1	1	1
	SNR	35.90	35.75	35.59	35.44	35.29
	TIME	0.04	0.04	0.04	0.04	0.04
6	ACC	1	1	1	1	1
	SNR	35.91	35.75	35.60	35.45	35.30
	TIME	0.04	0.04	0.04	0.04	0.04
7	ACC	1	1	1	1	1
	SNR	35.91	35.75	35.60	35.45	35.30
	TIME	0.04	0.04	0.04	0.04	0.04
8	ACC	1	1	1	1	1
	SNR	35.89	35.72	35.57	35.41	35.27
	TIME	0.04	0.04	0.04	0.04	0.04



**Tabel A. 22 Tabel Hasil Uji Coba Metode DCT pada Nilai  $\Delta$  di sekitar  $61 \times 10^{-4}$  (2)**

Gambar	$\Delta$ (e-4) :	61	62	63	64	65
1	ACC	1	1	1	1	1
	SNR	35.15	35.01	34.87	34.73	34.59
	TIME	0.04	0.04	0.04	0.04	0.04
2	ACC	1	1	1	1	1
	SNR	35.15	35.01	34.87	34.73	34.59
	TIME	0.04	0.04	0.04	0.04	0.04
3	ACC	1	1	1	1	1
	SNR	35.15	35.00	34.86	34.72	34.58
	TIME	0.05	0.04	0.06	0.04	0.04
4	ACC	1	1	1	1	1
	SNR	35.18	35.03	34.88	34.74	34.61
	TIME	0.04	0.04	0.04	0.04	0.04
5	ACC	1	1	1	1	1
	SNR	35.15	35.00	34.86	34.72	34.58
	TIME	0.04	0.04	0.04	0.04	0.04
6	ACC	1	1	1	1	1
	SNR	35.16	35.02	34.87	34.73	34.59
	TIME	0.04	0.04	0.04	0.04	0.04
7	ACC	1	1	1	1	1
	SNR	35.16	35.02	34.88	34.73	34.59
	TIME	0.04	0.04	0.04	0.04	0.04
8	ACC	1	1	1	1	1
	SNR	35.12	34.98	34.84	34.70	34.56
	TIME	0.04	0.04	0.04	0.04	0.04

**Tabel A. 23 Tabel Hasil Uji Coba Metode DCT pada Nilai  $\Delta$  di sekitar  $61 \times 10^{-4}$  (3)**

Gambar	$\Delta$ (e-4) :	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>
9	ACC	1	1	1	1	1
	SNR	35.87	35.72	35.57	35.41	35.26
	TIME	0.04	0.04	0.04	0.04	0.04
10	ACC	1	1	1	1	1
	SNR	35.94	35.79	35.63	35.48	35.33
	TIME	0.04	0.04	0.04	0.04	0.04
Gambar	$\Delta$ (e-4) :	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	<b>65</b>
9	ACC	1	1	1	1	1
	SNR	35.12	34.97	34.84	34.71	34.57
	TIME	0.04	0.04	0.04	0.04	0.04
10	ACC	1	1	1	1	1
	SNR	35.18	35.04	34.90	34.76	34.62
	TIME	0.04	0.04	0.04	0.04	0.06

**Tabel A. 24 Rekap Hasil Uji Coba Metode DCT pada Tabel A. 21 hingga Tabel A. 23**

Gambar	$\Delta$ (e-4) :	<b>56</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>
1 - 10	Mean SNR :	35.91	35.75	35.59	35.44	35.29
Gambar	$\Delta$ (e-4) :	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	<b>65</b>
1 - 10	Mean SNR :	35.15	35.01	34.87	34.73	34.59

## BIODATA PENULIS



Ebenhaezer Wiramarta Yogaswara lahir di Jayapura, pada tanggal 29 Mei 1994. Penulis menempuh pendidikan mulai dari SD YPPI Surabaya (1999-2005), SMP Kr. Petra 3 Surabaya (2005-2008), SMA Kr Petra 2 Surabaya (2008-2011) dan S1 Teknik Informatika ITS (2011-2015).

Selama masa kuliah penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTC). Penulis juga aktif dalam kegiatan kepanitiaan Schematics. Diantaranya penulis pernah menjadi staff seminar nasional NST Schematics 2012 dan staff Humas Schematics 2013.

Selama kuliah di Teknik Informatika ITS, penulis mengambil bidang minat Komputasi Cerdas Visual (KCV). Komunikasi dengan penulis dapat melalui email: [ebenwy@gmail.com](mailto:ebenwy@gmail.com).