



TESIS KI142502

**PENINGKATAN KAPASITAS DAN PERBAIKAN  
KUALITAS PENYISIPAN AUDIO DATA HIDING  
DENGAN NEWTON'S DIVIDED-DIFFERENCE  
INTERPOLATING POLYNOMIALS**

Tegar Palyus Fiqar  
5114201069

DOSEN PEMBIMBING  
Tohari Ahmad, S.Kom., MIT., Ph.D.

PROGRAM MAGISTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2017



Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom.)  
di  
Institut Teknologi Sepuluh Nopember Surabaya

oleh:  
TEGAR PALYUS FIQAR  
NRP. 5114201069

Dengan judul :  
Peningkatan Kapasitas dan Perbaikan Kualitas Penyisipan Audio *Data Hiding* dengan *Newton's Divided-Difference Interpolating Polynomials*

Tanggal Ujian : 13-7-2017  
Periode Wisuda : 2016 Genap

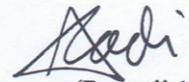
Disetujui oleh:

Tohari Ahmad, S.Kom, MIT, Ph.D  
NIP. 19750525 200312 1 002



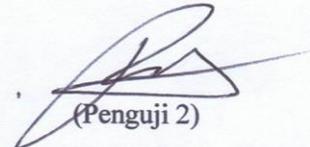
(Pembimbing 1)

Dr.Eng. Radityo Anggoro, S.Kom, M.Sc  
NIP. 19841016 200812 1 0002



(Penguji 1)

Royyana Muslim I, S.Kom, M.Kom, Ph.D  
NIP. 19770824 200604 1 001



(Penguji 2)

Waskitho Wibisono, S.Kom, M.Eng, Ph.D  
NIP. 19741022 200003 1 001



(Penguji 3)

Dekan Fakultas Teknologi Informasi,  
  
Dr. Agus Zainal Arifin, S.Kom., M.Kom.  
NIP. 19720809 199512 1 001

# **Peningkatan Kapasitas dan Perbaikan Kualitas Penyisipan *Audio Data Hiding* dengan *Newton's Divided-Difference Interpolating Polynomials***

Nama mahasiswa : Tegar Palyus Fiqar  
NRP : 5114201069  
Pembimbing I : Tohari Ahmad, S.Kom, MIT., Ph.D.

## **ABSTRAK**

Steganografi adalah sebuah cara mengubah data secara tidak terdeteksi untuk menyisipkan sebuah pesan pada data tersebut. Secara umum, system steganografi terdiri dari dua komponen utama yaitu encoder dan decoder. Saat ini penelitian mengenai steganografi telah banyak dilakukan, namun kebanyakan metode yang ada hanya diterapkan berkas citra dan hanya sedikit penelitian yang terkait dengan penyisipan berkas rahasia pada berkas audio. Beberapa metode yang ada juga memiliki kelemahan dimana berkas yang digunakan sebagai media penyisipan tidak dapat dikembalikan seperti semula, serta metode yang ada masih menggunakan berkas tambahan untuk *location maps* dan juga masalah kapasitas penyisipan yang masih dirasa kurang optimum, sehingga diperlukan suatu metode yang bersifat *reversible* yang mampu menghilangkan berkas tambahan *location maps*.

Penelitian ini mengusulkan metode baru penyisipan data yang mengadopsi dari proses pembentukan kurva dari suatu data, hal ini didasarkan karena berkas audio memiliki karakteristik dan bentuk yang dihasilkan dari berkas audio menyerupai dengan kurva pada sekumpulan data. Metode yang diusulkan juga berperan sebagai *pivoting point* yang digunakan titik acuan penyisipan sebagai pengganti *location maps*.

Dalam skenario uji coba, *dataset* diperoleh dari *IRMAS dataset*. Ada 5 jenis instrumen musik yang digunakan dan 3 *genre* musik sebagai media *cover* dalam penelitian ini yaitu *country-folk*, klasik, dan *pop-rock*. Durasi berkas audio yang digunakan pada masing-masing jenis musik selama 3 detik. Format dari berkas pembawa yang digunakan berekstensi *.wav*, hal ini dipilih karena berkas audio format *wav* tanpa ada proses kompresi. Berkas yang digunakan sebagai data *payload* merupakan berkas teks yang diperoleh dari teks *generator*. Ada beberapa hal yang diuji pada penelitian ini antara lain kapasitas maksimum *payload* yang disisipkan, pengaruh jenis instrumen, pengaruh *genre* musik, serta proses reduksi signal. *Signal noise to ratio* dan perhitungan kapasitas digunakan untuk mengevaluasi hasil penyisipan sesuai dengan metode yang diusulkan. Hasil penelitian yaitu kapasitas maksimum diperoleh pada audio 3 sebesar 1262296 bit dengan *genre pop-rock*, instrumen cello. Kualitas terbaik meningkat 33.8% dari penelitian sebelumnya menggunakan audio *genre country-folk* dan instrumen *human voice*.

**Kata kunci:** Steganografi, Berkas Audio, *Newton's Divided-Difference Interpolating Polynomials*.

*[Halaman ini sengaja dikosongkan]*

## ***Enhancement Capacity and Improvement Quality Audio Data Hiding with Newton's Divided-Difference Interpolating Polynomials***

Student Name : Tegar Palyus Fiqar  
NRP : 5114201069  
Supervisor : Tohari Ahmad, S.Kom, MIT., Ph.D.

### **ABSTRACT**

Steganography is a way to change the undetected data to hide a message into it. Generally, the steganography system consists of two main components; namely encoder and decoder. Currently researches on steganography have been widely practiced, but most of them are only applied in image files and only some researches that related to the hiding data of secret files in audio files. Some of existing methods still have disadvantages e.g.; the files that used as insertion media can't be able to restored as the original one, it still used additional files for maps location, and the problem of hidden capacity is still not optimum yet. In order to solve the disadvantages, it is required a reversible method which is able to eliminate the additional maps location.

This research proposes a new method of data hiding which is adopted from the curve formation process of a data. It is based on the audio file exhibits the characteristics and also the result form of the audio beam resembles to the curve in the data set. The proposed method also become as a pivoting point which is used as a reference of insertion point instead of maps location.

In trial scenario, the dataset is derived from the IRMAS dataset. In this research, there are 5 types of musical instruments that is used and 3 genres of music as media cover; country-folk, classic, and pop-rock. The duration for audio files that is used in each type of music is 3 seconds. The format of the bearer file used .wav extension, because the wav format in audio file without any compression process. The file that is used as payload data is a text file which is obtained from the generator text. In this research, some parameters like the maximum payload capacity inserted; the influence of instrument type; the influence of the music genre; and the process of signal reduction are tested. According to the proposed method, the signal noise to ratio and the capacity calculation are used to evaluate the results of the dispersion. The final result is obtained the maximum capacity in audio 3 equal to 1262296 bit on pop-rock genre, cello instrument. The best quality increased 33.8% from the previous research which is used country-folk genre and human voice instruments audio.

**Keyword: Steganography, Audio File, Newton's Divided-Difference Interpolating Polynomials.**

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Segala puji syukur bagi Allah SWT atas segala rahmat, taufik, hidayah, serta kasih sayang yang selama ini telah diberikan sehingga penulis mampu menyelesaikan tesis dengan judul “Peningkatan Kapasitas dan Perbaikan Kualitas Penyisipan *Audio Data Hiding* dengan *Newton’s Divided-Difference Interpolating Polynomials*”. Shalawat serta salam tetap tercurahkan kepada Rasulullah Muhammad SAW yang menjadi tauladan bagi seluruh umat Islam, khususnya bagi penulis. Penulis selalu mencoba untuk menjadikan beliau sebagai rujukan dalam segala tindakan dan perbuatan.

Tak lupa penulis mengucapkan terima kasih kepada semua pihak yang telah membantu penulis hingga terselesaikannya tesis ini, khususnya kepada:

1. Bapak dan ibu dirumah yang selalu mendukung, memotivasi dan tanpa henti
2. Mendoakan segala kebaikan.
3. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. atas segala bimbingan dan masukan yang diberikan dalam memecahkan berbagai kendala dalam penyusunan tesis ini serta kesabaran selama membimbing saya.
4. Keluarga yang selalu mendukung diberbagai kondisi.
5. DIKTI yang telah memberikan beasiswa pendidikan dan biaya hidup selama proses pendidikan berlangsung.
6. Universitas Borneo Tarakan yang telah memberikan rekomendasi untuk melanjutkan pendidikan pascasarjana.
7. Keluarga besar mahasiswa pascasarjana Teknik Informatika angkatan 2014.
8. Semua pihak yang tidak dapat disebutkan satu persatu yang telah banyak memberikan berbagai macam bantuan hingga terselesaikannya tesis ini.

Penulis menyadari bahwa tesis ini jauh dari kesempurnaan, oleh karena itu diharapkan kritik dan saran yang bersifat membangun untuk perbaikan kedepan. Akhirnya penulis berharap agar tesis ini dapat memberikan kontribusi yang bermanfaat bagi ilmu pengetahuan di kemudian hari.

Surabaya, Juni 2017

Penulis

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

ABSTRAK .....	iii
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	3
1.3 Tujuan .....	3
1.4 Batasan Masalah .....	3
1.5 Kontribusi Penelitian .....	4
1.6 Manfaat .....	4
1.7 Sistematika Penulisan .....	4
BAB 2 DASAR TEORI DAN KAJIAN PUSTAKA.....	6
2.1 Berkas Audio .....	7
2.1.1 Analog menuju Digital .....	8
2.1.2 Jenis Format Berkas Audio WAV (WAVE-form) .....	9
2.2 Data Hiding.....	10
2.3 <i>Reversible</i> dan <i>Irreversible</i> .....	13
2.4 <i>Newton's Divided Difference Interpolating (NDDI) Polynomials</i> .....	15
2.5 <i>Jung &amp; Yoo Method</i> .....	22
BAB 3 METODE PENELITIAN.....	27
3.1 Langkah-langkah Penelitian.....	27
3.2 Studi Literatur .....	27
3.3 Perancangan Algoritma.....	28
3.1.1 <i>Preprocessing Payload</i> .....	29
3.1.2 Penyisipan Pesan .....	30
3.1.3 Ekstraksi Pesan dan <i>Recovery</i> Berkas Audio .....	42

3.1.4 <i>Pos Processing Payload</i> .....	45
3.4 Implementasi Algoritma.....	47
3.5 Perancangan Uji Coba Metode.....	47
3.6 Evaluasi .....	49
3.7 Penyusunan Laporan Akhir .....	50
3.7 Jadwal Kegiatan Penelitian .....	51
<b>BAB 4 HASIL DAN PEMBAHASAN</b> .....	<b>52</b>
4.1 Lingkungan Uji Coba .....	53
4.2 Pelaksanaan Uji Coba.....	53
4.3 Hasil dan Evaluasi .....	55
4.3.1 Skenario 1 .....	55
4.3.2 Skenario 2 .....	60
<b>BAB 5 KESIMPULAN DAN SARAN</b> .....	<b>65</b>
5.1 Kesimpulan.....	65
5.2 Saran.....	66
<b>DAFTAR PUSTAKA</b> .....	<b>67</b>
<b>BIODATA PENULIS</b> .....	<b>68</b>

## DAFTAR GAMBAR

Gambar 2.1 Representasi Gelombang Suara.....	7
Gambar 2.3 Representasi Proses <i>Sampling</i> .....	9
Gambar 2.4 Reversible <i>Data Hiding</i> .....	14
Gambar 2.5. Contoh Interpolasi Polinomial dengan Orde Pertama ( <i>linier</i> ) yang Menghubungkan Dua Buah Titik.....	16
Gambar 2.6. Contoh Interpolasi Polinomial dengan Orde Kedua ( <i>quadratic</i> /parabola) yang Menghubungkan Tiga Buah Titik. ....	16
Gambar 2.7. Contoh Interpolasi Polinomial dengan Orde Ketiga ( <i>qubic</i> ) yang Menghubungkan Empat Buah Titik. ....	16
Gambar 2.8. <i>Curve Fitting</i> .....	18
Gambar 2.9. <i>Lagrange interpolation polynomials</i> orde 1.....	21
Gambar 2.10. <i>Lagrange interpolation polynomials</i> orde 2.....	22
Gambar 2.12 Skema Jung & Yoo .....	22
Gambar 2.13 Interpolasi Jung & Yoo .....	23
Gambar 3.1 Diagram Alur Kerja Penelitian.....	27
Gambar 3.3 Sampling Audio .....	30
Gambar 3.4 Normalisasi.....	31
Gambar 3.5 NDDI Proses .....	31
Gambar 3.6 Penyisipan <i>Payload</i> .....	32
Gambar 3.7 Reduksi Signal .....	32
Gambar 3.8 Stego-Audio .....	33
Gambar 3.9 Proses <i>Encoding</i> .....	34
Gambar 3.10 Proses <i>Sampling</i> .....	35
Gambar 3.11 Proses Pemisahan <i>Sampling</i> .....	36
Gambar 3.12 Hasil <i>Sampling</i> Audio dan <i>Sampling</i> NDDI .....	36
Gambar 3.13 Evaluasi Kinerja Fungsi Parameter Jumlah Bit .....	38
Gambar 3.14 Proses <i>Smoothing</i> .....	40
Gambar 3.15 Skema <i>Decoding</i> .....	43
Gambar 3.16 Bagan Pengembalian Data <i>Payload</i> .....	46

Gambar 3.17. Signal Berkas Audio <i>Genre</i> Pop-Rock.....	48
Gambar 3.18. Signal Berkas Audio yang <i>Genre</i> Klasik.....	48
Gambar 4.1 Hasil Perbandingan Kapasitas .....	55
Gambar 4.2 Hasil Perbandingan Kualitas .....	61

## DAFTAR TABEL

Tabel 2.1 Karakteristik Berkas Audio.....	10
Tabel 3.2 Jadwal Kegiatan Penelitian .....	51
Tabel 4.1 Lingkungan Implementasi Penelitian.....	53
Tabel 4.2 Dataset Berkas Audio.....	54
Tabel 4.3 Karakteristik Audio.....	54
Tabel 4.4 Kapasitas Maksimum Metode Usulan dan Pembanding.....	56
Tabel 4.5 Selisih Kapasitas Antara Metode Usulan Terhadap Metode Pembanding.....	59
Tabel 4.6 Hasil Rerata Kualitas Stego-audio .....	62

*[Halaman ini sengaja dikosongkan]*

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Berkembangnya teknologi jaringan terutama internet memberikan dampak besar pada distribusi dan pengiriman berkas digital. Ada kalanya berkas digital yang dikirimkan bersifat rahasia dan *confidential* seperti, data catatan kesehatan pasien atau pesan militer, sehingga harus dipastikan keamanan isi dari berkas hanya diketahui oleh pihak yang berhak. Salah satu metode yang dapat digunakan untuk menjaga kerahasiaan adalah steganografi.

Stegnaografi adalah sebuah praktik mengubah data secara tidak terdeteksi untuk menyisipkan pesan (Ingemar dkk, 2008). Secara umum, sistem steganografi terdiri dari dua komponen yaitu *encoder* dan *decoder*. Bagian *encoder* mempunyai dua buah masukan, yaitu data rahasia atau yang biasa disebut *payload* dan *cover*, yaitu berkas yang digunakan sebagai media untuk menyisipkan data rahasia. Kedua masukan ini kemudian diproses untuk menghasilkan *encode* data, yaitu data yang telah disisipkan pesan rahasia. Pada bagian *decoder*, *encode* data menjadi masukan yang selanjutnya diproses untuk didapatkan kembali pesan rahasia yang tersimpan didalamnya (Ingemar dkk,2008).

Penelitian pada bidang keamanan informasi dan steganografi telah banyak dilakukan dan berkembang pada beberapa tahun belakangan ini. Namun, kebanyakan metode yang telah ada hanya dapat diterapkan pada berkas citra, sedangkan metode yang dapat digunakan pada berkas bertipe audio masih sangat terbatas. Beberapa metode yang telah ada mencakup, *Low Bit Encoding* (Bilal dkk, 2014), *Echo Hiding* (Zhou dkk, 2013), *Spread Spectrum* (Lili dkk, 2010), *Phase Encoding* (Iynkaran dkk, 2010), dan juga menggunakan transformasi *wavelet* (Patel dkk, 2013). Namun, metode ini memiliki kelemahan dimana berkas audio *cover* dan *secret message* tidak dapat dikembalikan persis seperti semula dikarenakan adanya modifikasi pada *domain* spasial ataupun domain frekuensi. Modifikasi tersebut bersifat *irreversible* atau tidak dapat dikembalikan. Hal ini tentu harus dihindari

karena informasi pada *secret message* bersifat sensitif dan harus dikembalikan tanpa adanya informasi yang hilang.

Untuk mendapatkan kembali data *cover* dan *secret message* tanpa kehilangan informasi, harus dilakukan dengan menggunakan modifikasi yang bersifat *reversible* pada berkas. Hal ini dapat dilakukan dengan menggunakan metode transformasi *Difference expansion* yang pertama kali di usulkan oleh Tian (Tian dkk, 2003). Banyak metode yang diajukan sebagai pengembangan metode ini seperti *Reduced expansion* (Tohari dkk, 2014) dan *Quad Smoothness* (Tohari dkk, 2014), tetapi metode ini hanya dapat diaplikasikan pada berkas citra. Penelitian selanjutnya dilanjutkan dengan mengembangkan metode *Generalized Reduce Expansion* dengan algoritma *Genetic* yang diterapkan pada berkas audio (Tohari dkk, 2015).

Penelitian mengenai data *hiding* pada berkas audio yang digunakan sebagai media *cover* dengan menggunakan metode interpolasi diawali pada 2006 (Fujimoto dkk, 2006). Pada penelitian ini dikembangkan dengan menggunakan *spline* interpolasi, penerapan *spline* interpolasi pada berkas audio dengan cara memperoleh nilai baru pada berkas audio asli, namun nilai baru ini diperoleh dari *slope* signal yang dihasilkan, sehingga akan signal baru akan terbentuk dari *frame* berkas audio. Karena penelitian ini mengembang pada *slope* dari berkas audio sehingga tidak dapat penyisipan secara maksimum. Penelitian ini juga tidak menyelesaikan permasalahan multi lapis. Pengembangan penyisipan data dengan interpolasi dilanjutkan dengan menggunakan *nearest neighbor* interpolasi (Jung-yoo dkk, 2009) namun penelitian ini hanya diterapkan pada media citra sehingga memiliki keterbatasan dalam melakukan banyaknya orde yang digunakan.

Penelitian ini mengajukan, metode peyisipan berkas data rahaisa dengan *langrange interpolasi polynomial* yang awalnya diterapkan pada berkas citra akan dicoba diterapkan pada berkas audio dengan ditambah beberapa modifikasi untuk memaksimalkan kuantitas serta kualitas dari berkas audio. Sebelum metode dapat diimplemnetasikan harus dilakukan beberapa tahap *pre-processing* agar metode tersebut dapat diimplentasikan pada berkas audio. Tahapan ini meliputi interpolasi signal audio dan penyisipan berkas rahaisa pada *sample* audio serta penyesuaian *sample rate* pada berkas audio yang telah disisip data rahaisa agar durasi audio terdengar sesuai dengan berkas audio sebelum disisipi berkas rahaisa.

## 1.2 Perumusan Masalah

Dari latar belakang diatas didapatkan permasalahan sebagai berikut:

1. Bagaimana melakukan penyisipan tanpa mempengaruhi dan merusak *sampling* data audio asli?
2. Bagaimana penyisipan data rahasia tanpa *location map*?
3. Bagaimana meningkatkan kapasitas penyisipan *payload* pada berkas audio kanal tunggal dengan *newton's divided-difference interpolating polynomials*?
4. Bagaimana memperbaiki kualitas penyisipan *payload* pada berkas audio kanal tunggal dengan *newton's divided-difference interpolating polynomials*?
5. *Genre* apa saja yang dapat menyisipkan kapasitas maksimum serta memiliki kualitas sisipan terbaik?

## 1.3 Tujuan

Penelitian ini memiliki beberapa tujuan yang diperoleh dari rumusan masalah antara lain:

1. Melakukan penyisipan data *payload* pada signal audio yang diperluas dengan *newton's divided-difference interpolating polynomials*.
2. Meningkatkan kapasitas penyisipan *payload* pada berkas audio kanal tunggal dengan metode *newton's divided-difference interpolating polynomials*.
3. Meningkatkan kualitas penyisipan *payload* pada berkas audio kanal tunggal dengan metode *newton's divided-difference interpolating polynomials*.
4. Menerapkan konsep *pivoting point* sebagai pengganti *location map*.
5. Mengetahui *genre* yang paling banyak dapat disisipkan dengan kualitas terbaik.

## 1.4 Batasan Masalah

Permasalahan yang dibahas dalam penelitian ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Metode hanya diterapkan pada berkas audio dengan format wav.
2. Berkas audio yang digunakan tanpa proses kompresi.
3. Berkas audio yang digunakan terdiri dari kanal tunggal.
4. *Bit-depth* atau panjang *sample* dari audio yang digunakan sebagai *cover* adalah sign 16-bit.
5. *Payload* yang digunakan adalah teks dengan format txt.

## 1.5 Kontribusi Penelitian

Kontribusi pada penelitian ini adalah melakukan perbaikan kinerja penyisipan audio *data hiding* pada metode Jung dan Yoo dengan mengembangkan fungsi baru *nonlinear* berpangkat logaritmik agar diperoleh peningkatan kapasitas penyisipan data rahasia, serta penambahan proses *smoothing* signal stego-audio yang bertujuan untuk memperoleh perbaikan kualitas hasil sisipan pada media pembawa.

## 1.6 Manfaat

Manfaat yang didapat dari penelitian ini dapat dipandang dari tiga aspek yaitu peneliti, keilmuan, dan manfaat dari pembaca. Adapun manfaat tersebut adalah sebagai berikut.

1. Peneliti:
  - a. Dapat memperdalam pemahaman dan implementasi konsep/teori tentang penyembunyian data pada berkas audio.
  - b. Sebagai sarana untuk mengetahui perbandingan kinerja metode penyembunyian data yang diusulkan peneliti terhadap beberapa metode yang ada sebelumnya.
2. Keilmuan: menghasilkan metode baru *newton's divided-difference interpolating polynomials* untuk melakukan penyembunyian data pada media berkas audio yang mampu meningkatkan kapasitas data rahasia yang disisipkan dan meningkatkan stego-audio.
3. Pembaca: metode yang dihasilkan dapat dijadikan sebagai metode acuan untuk melakukan penyembunyian data rahasia dapat berkas audio.

## 1.7 Sistematika Penulisan

Sistematika penulisan pada penelitian ini adalah sebagai berikut:

1. **BAB I PENDAHULUAN.** Bab ini berisi pendahuluan yang menjelaskan latar belakang, perumusan masalah, tujuan, batasan masalah, tujuan, manfaat, kontribusi penelitian dan sistematika penulisan terkait dengan penelitian.
2. **BAB II DASAR TEORI DAN TINJAUAN PUSTAKA.** Bab ini berisi dasar-dasar teori dan tinjauan pustaka terkait penelitian, diantaranya berkas

audio, konsep *data hiding*, *interpolating langrange polynomials* dan teori pembandingan.

3. **BAB III METODOLOGI PENELITIAN.** Bab ini berisi tahapan penelitian meliputi studi literatur, rancangan penelitian dan skenario pengujian terhadap penelitian ini.
4. **BAB IV HASIL DAN PEMBAHASAN.** Bab ini berisi penyajian hasil penelitian serta pembahasan dari skenario yang dilakukan.
5. **BAB IV KESIMPULAN DAN SARAN.** Bab ini bagian terakhir dari naskah *thesis* yang berisi kesimpulan dari penelitian yang dilakukan serta saran untuk penelitian dikemudian hari.

*[Halaman ini sengaja dikosongkan]*

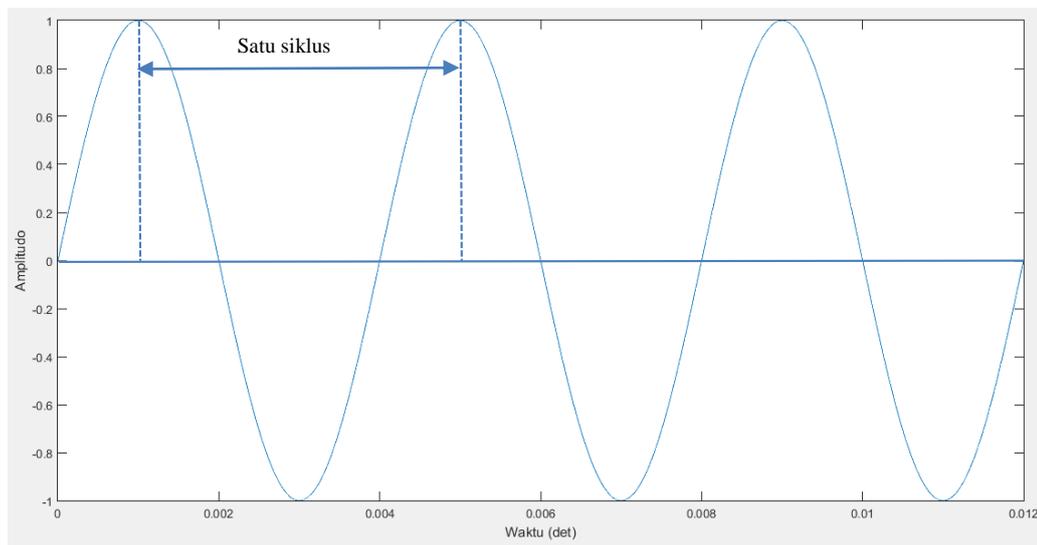
## BAB 2

### DASAR TEORI DAN KAJIAN PUSTAKA

Pada bab kedua akan dibahas beberapa kajian pustaka untuk memberikan gambaran yang jelas berkaitan dengan penelitian ini. Kajian pustaka terbagi dalam lima garis besar pembahasan, yaitu berkas audio, *data hiding*, *reversible* dan *irreversible*, *interpolating polynomials*, dan metode pembandingan.

#### 2.1 Berkas Audio

Suara merupakan fenomena fisik alami yang dihasilkan oleh suatu getaran. Getaran ini menghasilkan tekanan yang berbeda-beda melalui media udara. Pola osilasi yang terjadi diudara tersebut diistilahkan dengan gelombang. Bentuk gelombang yang dihasilkan merupakan gelombang analog atau kontinu yang membawa informasi. Dua karakteristik yang dimiliki oleh gelombang analog adalah amplitudo dan frekuensi. Amplitudo merupakan ukuran tinggi rendahnya tegangan dari gelombang analog. Frekuensi merupakan jumlah gelombang analog dalam satu detik.



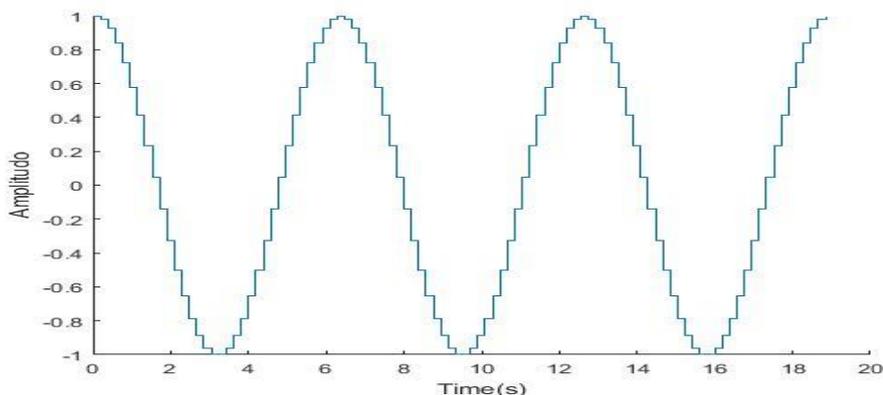
Gambar 2.1 Representasi Gelombang Suara

Gelombang suara memiliki lembah dan bukit, satu buah lembah dan bukit akan menghasilkan satu siklus. Siklus ini berlangsung secara berulang-ulang dan perulangan siklus tiap detiknya disebut frekuensi. Satu unit frekuensi dinamakan

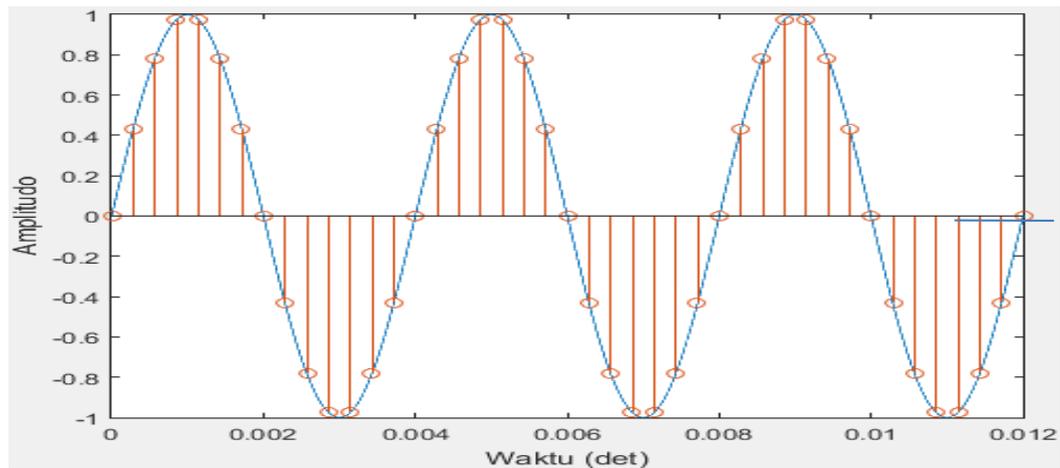
sebagai Hertz (Hz). Telinga manusia dapat mendengar suara antara 20 Hz hingga 20 kHz. Artinya jika sebuah benda dapat bergetar dan menghasilkan siklus setiap detiknya sebesar 20 kali, maka telinga dapat menangkap suara dari getaran benda tersebut.

### 2.1.1 Analog menuju Digital

Gelombang suara analog yang dihasilkan tidak dapat langsung direpresentasikan atau direkam oleh komputer. Komputer perlu mengukur amplitudo pada satuan waktu tertentu untuk menghasilkan sebuah angka. Proses signal analog menjadi signal digital diawali dengan proses kuantisasi. Proses kuansisasi ini seperti pada Gambar 2.2. Setelah tahapan kuantisasi Komputer melakukan penyimpanan angka tersebut kedalam sebuah berkas audio, berkas audio tersebut nantinya digunakan saat diakses/*decode* menjadi suara. Setiap pengukuran ini dinamakan *sampling*. Proses *sampling* hasil dari kuantisasi ditunjukann pada Gambar 2.3. File audio memiliki *sampling rate* sebesar 44.100Hz. Artinya dalam satu detik, sample yang diambil sebanyak 44.100. Pada setiap *sample* diperlukan 2 byte / 16 bit data. Pada jenis audio stereo memiliki jumlah kanal lebih dari satu. Jika jenis berkas audio *stereo* yang digunakan memiliki dua kanal maka diperlukan  $2 \times 2 \text{ byte} = 4 \text{ byte}$ , sehingga untuk 1 detik yang terdiri dari 44.100 sample, besar berkas audio tersebut  $4 \times 44.100$  atau 176.400 byte (172 Kb). Jika durasi file audio tersebut 4 menit, maka ukuran file sebesar  $172 \text{ Kb} \times 4 \times 60 \text{ detik} = 41.280 \text{ Kb}$ .



Gambar 2.2 Representasi Proses *Kuantisasi*



Gambar 2.3 Representasi Proses *Sampling*

### 2.1.2 Jenis Format Berkas Audio WAV (WAVE-form)

*Waveform audio format* (WAV) merupakan standar format berkas audio yang dikembangkan oleh Microsoft dan IBM, WAV memiliki format utama untuk menyimpan sata audio mentah pada Windows dan menggunakan metode yang sama dengan AIFF Apple untuk menyimpan data audio. WAV umumnya digunakan untuk menyimpan audio yang tak termampatkan, berkas audio yang berkualitas tinggi dengan ukuran berkas besar. Extensi berkas audio yang digunakan pada WAV yaitu .wav atau .wv. Kelebihan format berkas audio WAV yaitu menggunakan *coding* PCM (*Pulse Code Modulation*) yang mana proses analog menuju digital dan proses penyimpanan tidak akan kehilangan informasi, berkas audio tidak terkompres sehingga keseluruhan *sample* audio akan tersimpan. Kekurangan format berkas audio WAV yaitu maksimal ukuran berkas sebesar 2 GB, jarang digunakan diinternet karena ukuran yang *relative* besar.

Ada beberapa format audio yang dapat dibaca dan diolah dengan perangkat lunak MATLAB untuk seluruh *platform* antara lain: WAVE (.wav), OGG (ogg), FLAC (.flac) AU (.AU). Dua data tipe yang dapat dibaca oleh MATLAB yaitu *double* dan *native*. Data tipe *double* digunakan untuk audio dengan proses *sampling* yang sudah ternormalisasi, sehingga memiliki nilai *magnitude sampling* berada diantara -1 dan 1. Data tipe *native* digunakan untuk audio tanpa proses *sampling* yang ternormalisasi, sehingga memiliki nilai *magnitude sampling* tanpa mengalami

perubahan, semisal audio wave yang memiliki 16bit maka *range* nilai berada - 32768 dan 32767.

Setiap audio memiliki bit per sample yang berbeda-beda serta data tipe yang berbeda. Karakteristik dari jenis audio seperti Tabel 2.1.

Tabel 2.1 Karakteristik Berkas Audio

<i>Format File</i>	<i>Bits Per Sample</i>	<i>Data Type</i>	<i>Range Data</i>
Wave (.wav)	8	UInt8	$0 \leq y \leq 255$
	16	Int16	$-32768 \leq y \leq +32767$
	24	Int32	$-2^{32} \leq y \leq 2^{32}-1$
	32	Int32	$-2^{32} \leq y \leq 2^{32}-1$
	32	Single	$-1 \leq y \leq 1$
	64	Double	$-1 \leq y \leq 1$
FLAC (.flac)	8	UInt8	$-32768 \leq y \leq +32767$
	16	Int16	$-2^{32} \leq y \leq 2^{32}-1$
	24	Int32	$-1 \leq y \leq 1$

## 2.2 Data Hiding

Steganografi berasal dari kata Yunani, steganografi adalah ilmu dan seni menyembunyikan keberadaan komunikasi. Dengan menggunakan steganografi pesan rahasia dapat disisipkan kedalam sebuah media yang tidak mencurigakan dan mengirimnya tanpa ada seorangpun yang mengetahui keberadaan pesan tersebut. Steganografi berbeda dengan kriptografi yaitu terletak pada hasil keluarannya. Hasil dari kriptografi memiliki bentuk yang berbeda dengan data asli, sehingga informasi yang ada pada data tersebut diketahui tetapi tidak dapat dimengerti karena informasi tersebut dikodekan terlebih dahulu. Sedangkan hasil keluaran steganografi memiliki bentuk yang sama dengan data aslinya, sehingga keberadaan informasi yang disembunyikan tidak terlihat menurut persepsi indra manusia tetapi tidak oleh komputer atau pengolah data digital lainnya. Secara umum steganografi dan kriptografi memiliki tujuan yang sama yaitu menyembunyikan informasi supaya tidak dapat dibaca, dimengerti atau diketahui secara langsung. Steganografi memanfaatkan keterbatasan yang ada pada indera manusia seperti mata dan telinga. Dengan keterbatasan inilah maka steganografi dapat diterapkan dalam berbagai

media digital. Kualitas dari steganografi terletak pada ada tidaknya pesan rahasia, keamanan steganografi menjadi patah jika kunci rahasia diketahui meskipun isi pesan belum diketahui. Menyisipkan data yang ingin disembunyikan ke dalam sebuah media membutuhkan dua buah arsip. Pertama adalah arsip media penampung seperti citra, suara, video dan sebagainya yang terlihat tidak mencurigakan untuk menyimpan pesan rahasia. Arsip kedua adalah arsip pesan yang ingin disembunyikan, dapat berupa *plaintext*, *chipertext*, citra lain atau apapun yang dapat disembunyikan dalam *bit stream*.

Terdapat beberapa istilah yang berkaitan dengan steganografi, yaitu:

- *Embedded message* atau *hiddentext*, yaitu pesan rahasia yang disembunyikan.
- *Cover-object* atau *covertext*, yaitu media yang digunakan untuk menyembunyikan *hiddentext*. Media ini juga disebut media pembawa, citra penutup, *cover-image* dan *cover medium*.
- *Stego-object* atau *stegotext* yaitu, media yang sudah berisi *hiddentext*. Untuk media berupa citra sering disebut *stego-image* atau stegogram.
- *Stego-key* yaitu, kunci yang digunakan untuk mengacak posisi pesan pada penyisipan pesan dan mengekstraksi pesan dari stego-object misalnya *password*.
- Steganografer yaitu, orang yang merancang metode steganografi.
- Steganalisis yaitu, seni dan ilmu dalam mendeteksi ada tidaknya pesan tersembunyi dalam sebuah objek.
- Steganalis yaitu, orang yang berusaha untuk memecahkan metode steganografi dengan menggunakan berbagai metode steganalisis.

### **2.2.1 Metode Steganografi**

Secara umum steganografi merupakan ilmu yang mempelajari, meneliti, dan mengembangkan seni menyembunyikan sesuatu informasi. Secara teori, semua *file* umum yang ada di dalam komputer dapat digunakan sebagai media seperti *file* citra berformat JPEG, GIF, BMP atau didalam musik mp3 atau bahkan di dalam sebuah film dengan format WAV atau AVI. Semua dapat dijadikan tempat bersembunyi,

asalkan *file* tersebut memiliki bit-bit data redundan yang dapat dimodifikasi. Setelah dimodifikasi file media tersebut tidak akan banyak diganggu fungsinya dan kualitasnya tidak akan jauh berbeda dengan aslinya.

Tujuan dari steganografi adalah menyembunyikan keberadaan pesan dan dapat dianggap sebagai pelengkap dari kriptografi yang bertujuan untuk menyembunyikan isi pesan. Oleh karena itu, berbeda dengan kriptografi, dalam steganografi pesan disembunyikan sedemikian rupa sehingga pihak lain tidak dapat mengetahui adanya pesan rahasia. Pesan rahasia tidak diubah menjadi karakter aneh seperti halnya kriptografi. Pesan tersebut hanya disembunyikan ke dalam suatu media berupa citra, teks, musik, atau media digital lainnya dan terlihat seperti pesan biasa.

Beberapa contoh media penyisipan pesan rahasia yang digunakan dalam teknik steganografi antara lain adalah:

- Teks

Dalam algoritma steganografi yang menggunakan teks sebagai media penyisipannya biasanya digunakan teknik NLP sehingga teks yang telah disisipi pesan rahasia tidak akan mencurigakan untuk orang yang melihatnya.

- Audio

Format ini pun sering dipilih karena biasanya berkas dengan format ini berukuran relatif besar. Sehingga dapat menampung pesan rahasia dalam jumlah yang besar pula.

- Citra

Format ini juga sering digunakan, karena format ini merupakan salah satu format *file* yang sering dipertukarkan dalam dunia internet. Alasan lainnya adalah banyaknya tersedia algoritma steganografi untuk media penampung yang berupa citra.

- Video

Format ini memang merupakan format dengan ukuran *file* yang relatif sangat besar namun jarang digunakan karena ukurannya yang terlalu besar sehingga mengurangi kepraktisannya dan juga kurangnya algoritma yang mendukung format ini.

### 2.2.2 Aspek Steganografi

Penilaian sebuah algoritma steganografi yang baik dapat dinilai dari beberapa faktor yaitu:

- *Imperceptibility*

Keberadaan pesan rahasia dalam media penampung tidak dapat dideteksi oleh inderawi. Misalnya jika *coverttext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan *coverttext*-nya. Jika *coverttext* berupa audio, maka indera telinga tidak dapat mendeteksi perubahan pada *stegotext*-nya.

- *Fidelity*

Mutu media penampung tidak akan berubah banyak akibat penyisipan. Perubahan ini tidak dapat dipersepsi oleh inderawi. Misalnya, jika *coverttext* berupa citra, maka penyisipan pesan dapat membuat *stegotext* sukar dibedakan oleh mata dengan citra *coverttext*-nya. Jika *coverttext* berupa audio, maka audio *stegotext* tidak rusak dan indera telinga tidak dapat mendeteksi perubahan pada file *stegotext*-nya.

- *Recovery*

Pesan yang disembunyikan harus dapat diungkapkan kembali. Karena tujuan steganografi adalah data yang tersembunyi, maka sewaktu-waktu pesan rahasia di dalam *stegotext* harus dapat diambil kembali untuk digunakan lebih lanjut.

- *Size*

Ukuran dari berkas media pembawa dan media hasil sisipan tidak berbeda jauh, semisal berkas audio pembawa 512kB maka setelah penyisipan tidak berbeda jauh dari 512kB.

### 2.3 *Reversible dan Irreversible*

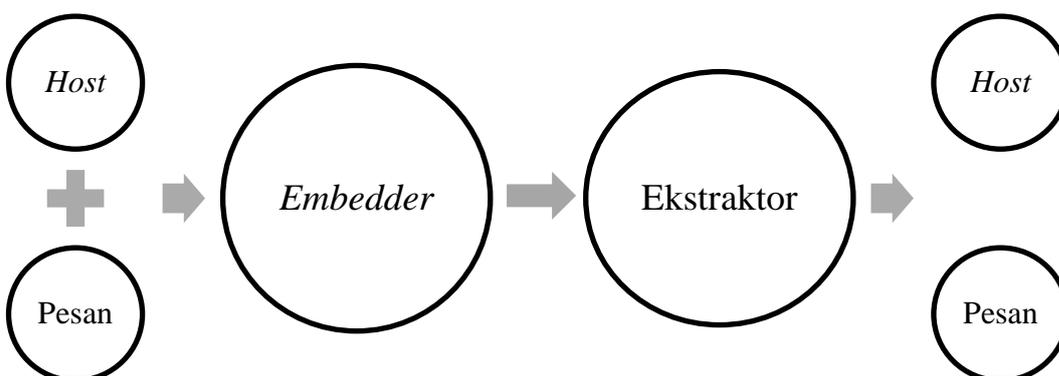
Metode *data hiding* yang tidak menyimpan informasi mengenai *recovery* dari stego image, sehingga stego image mengalami kerusakan secara permanen dikategorikan sebagai metode *irreversible data hiding* (Lin dkk, 2008). Umumnya, metode ini lebih fokus pada data rahasia yang disisipkan, tanpa memperhatikan media yang digunakan. Oleh karena itu, media yang digunakan adalah media-media

yang tidak membutuhkan akurasi, yaitu media yang dapat mentotolerir adanya distorsi. Contoh dari metode ini adalah substitusi *Least Significant Bit* (LSB).

*Reversible data hiding* berkerja seperti metode *data hiding* biasa yang mampu menyembunyikan pesan rahasia kedalam media yang digunakan, akan tetapi setelah pesan diekstrak metode ini mampu mengembalikan media kondisi semula tanpa meninggalkan distorsi (Feng dkk, 2006).

Umumnya, metode *data hiding* akan merubah media yang digunakan sebagai *cover* ketika proses penyisipan pesan, sehingga media yang digunakan akan mengalami distorsi. Pada beberapa kasus khusus membutuhkan presisi yang absolut, dimana distorsi tidak dapat diterima, seperti pada bidang medis dan militer.

Gambar 2.4 menunjukan alur dari metode *reversible data hiding*. *Embedder* akan menyisipkan pesan kedalam *host*. Pada proses ekstraksi, selain pesan, *host* juga di-*recovery* kekeadaan semula, sehingga distorsi dari proses penyisipan dapat dihilangkan.



Gambar 2.4 Reversible Data Hiding

Metode *reversible data hiding* juga harus memenuhi persyaratan yang digunakan oleh metode *data hiding* pada umumnya seperti *robustness*, *imperceptibility*, proses penyisipan dan proses ekstraksi dapat dilakukan dengan mudah. Namun untuk metode *Reversible data hiding* diperlukan dua persyaratan tambahan (Feng dkk, 2006), antara lain:

1. *Blind*

Beberapa metode *data hiding* membutuhkan keberadaan berkas asli untuk melakukan proses ekstraksi. Metode *Reversible data hiding* mampu mengembalikan berkas stego keadaan semula, sehingga menjadi seperti berkas asli.

Oleh karena itu, metode *reversible data hiding* harus memiliki kemampuan *blind* yang berarti proses ekstraksi tidak membutuhkan adanya berkas asli.

## 2. Kapasitas penyisipan besar

Kapasitas penyisipan diartikan sebagai jumlah informasi yang dapat disisipkan ke dalam suatu berkas. Karena metode *reversible data hiding* harus menyisipkan informasi mengenai *recovery* ke dalam berkas, maka kebutuhan kapasitas penyisipan menjadi lebih besar dibandingkan dengan metode *data hiding* biasa.

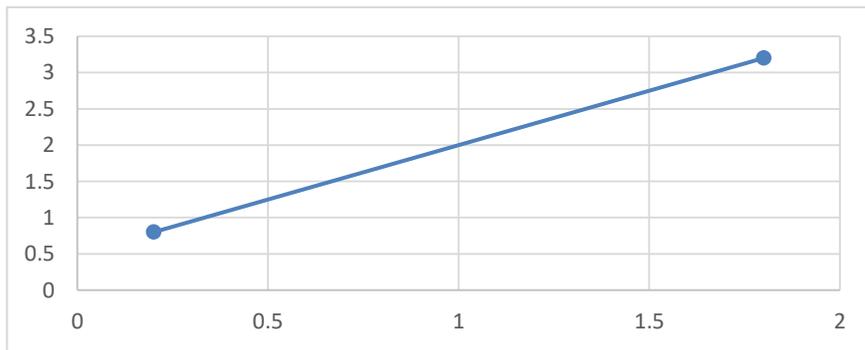
### 2.4 *Newton's Divided Difference Interpolating (NDDI) Polynomials*

Dalam kehidupan sehari-hari sering ditemukan kebutuhan untuk memperkirakan nilai menengah antara titik data yang tepat. Seperti contoh dalam proses pembentukan objek animasi tiga dimensi, posisi pergerakan suatu objek, grafik perubahan nilai mata uang, prediksi suatu nilai saham dan lain-lain. Metode yang paling umum digunakan untuk menyelesaikan permasalahan ini adalah dengan interpolasi polinomial.

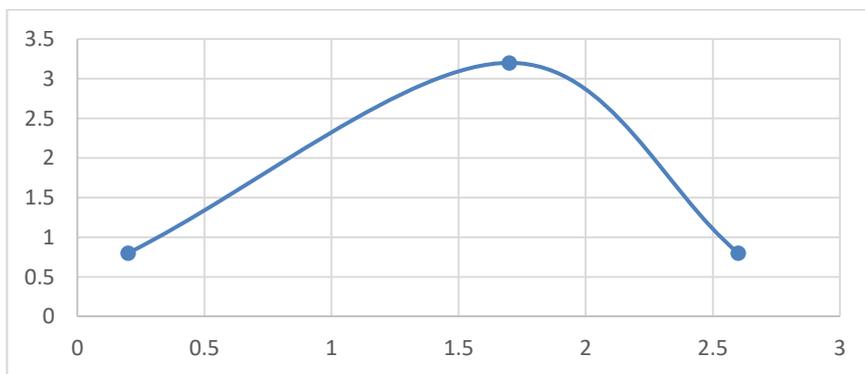
Persamaan rumus umum yang digunakan untuk polinomial dengan n-orde seperti berikut:

$$f(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (1)$$

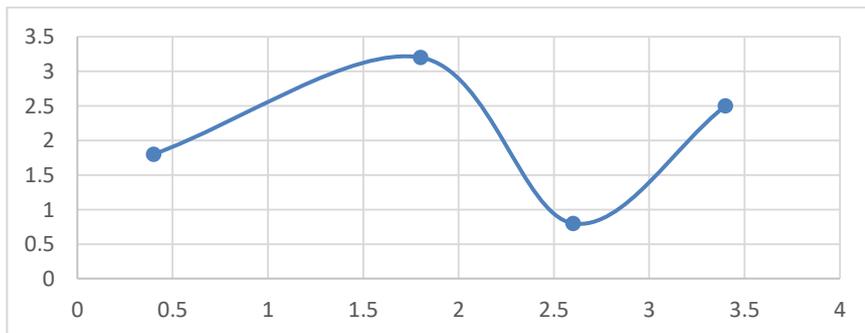
Dimana untuk  $n + 1$  titik data, ada satu dan hanya satu polinomial orde  $n$  yang melewati semua poin. Sebagai contoh, hanya ada satu garis lurus yaitu polinomial orde pertama yang menghubungkan dua titik pada Gambar 2.5. Demikian pula, hanya satu parabola menghubungkan satu himpunan titik yang terdiri dari tiga titik seperti pada Gambar 2.6. Pada interpolasi polinomial, untuk menentukan polinomial orde ke  $n$  maka banyaknya data yang sesuai yaitu sebanyak  $n + 1$  titik data. Polinomial ini selanjutnya memiliki formula untuk menghitung nilai menengah dari banyak titik data.



Gambar 2.5. Contoh interpolasi polinomial dengan orde pertama (*linier*) yang menghubungkan dua buah titik.



Gambar 2.6. Contoh interpolasi polinomial dengan orde kedua (*quadratic/parabola*) yang menghubungkan tiga buah titik.



Gambar 2.7. Contoh interpolasi polinomial dengan orde ketiga (*qubic*) yang menghubungkan empat buah titik.

Pada dasarnya interpolasi *lagrange* polinomial digunakan dalam proses pembentukan kurva dengan metode pendekatan dari keberadaan titik yang telah diketahui. Metode pendekatan ini ditujukan karena data-data yang digunakan berbentuk diskrit, sedang jika digunakan data yang bersifat kontinu maka data digunakan dengan model persamaan matematis.

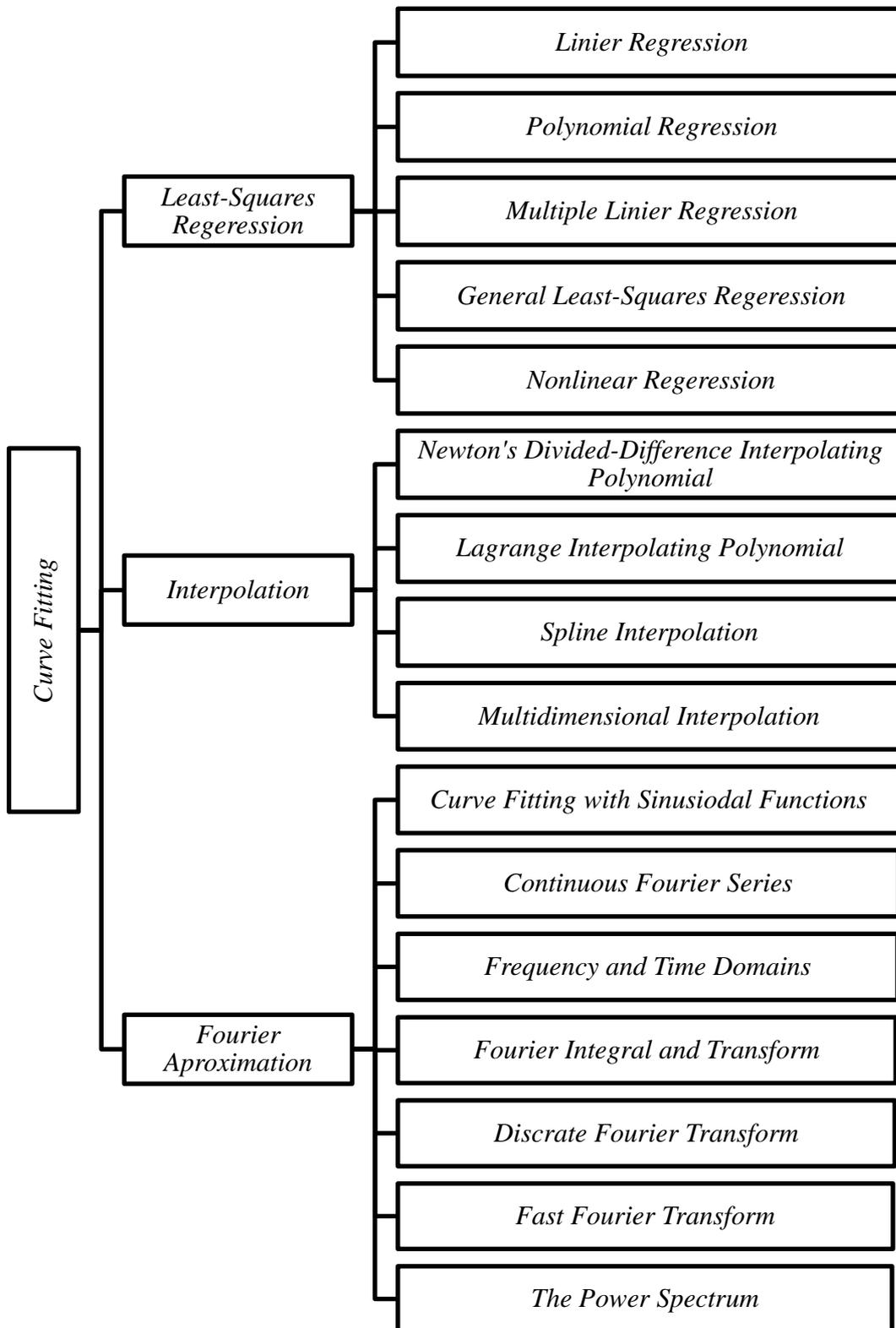
Ada tiga pendekatan untuk proses pembentukan kurva yaitu *least-squares regression*, *interpolation*, dan *approximation fourier*. Gambar 2.8 menunjukkan pembagian dari metode-metode yang dapat digunakan dalam proses pembentukan kurva dari sekumpulan atau himpunan data.

Metode pertama, di mana data menunjukkan tingkat signifikan kesalahan atau "noise", strategi ini adalah untuk memperoleh kurva tunggal yang mewakili kecenderungan umum dari data. Karena setiap titik data individu mungkin salah, dan tidak berusaha untuk memotong setiap titik. Sebaliknya, kurva dirancang untuk mengikuti pola titik-titik yang diambil sebagai sebuah kelompok. Salah satu pendekatan dalam ini disebut *least-squares regression*. *Least-squares regression* ada beberapa metode yang dikembangkan antara lain *linier regression*, *polynomial regression*, *multiple linier regression*, *general least-squares regression*, dan *nonlinear regression*.

Metode kedua, di mana data tersebut dikenal sangat tepat, pendekatan dasar adalah untuk menyesuaikan kurva atau serangkaian kurva yang lurus langsung melalui masing-masing titik. Data tersebut biasanya berasal dari tabel yang telah tersedia. Contohnya adalah nilai untuk kepadatan air atau untuk kapasitas panas gas sebagai fungsi temperatur. Estimasi nilai antara titik diskrit terkenal disebut interpolasi. Beberapa pengembangan metode interpolasi yang disesuaikan dengan kebutuhan dalam penelitian antara lain yaitu *newton's divided-difference interpolating polynomial*, *lagrange interpolation polynomial*, *spline interpolation*, dan *multi medimensional interpolation*.

Metode ketiga yaitu *Fourier approximation*. Metode ini digunakan untuk membentuk kurva dari beberapa titik, namun dengan merubah domain spasial menjadi domain frekuensi sehingga diperlukannya formula untuk transformasi sesuai dengan kebutuhan. Metode ini memiliki beberapa formula yang dikembangkan antara lain *curve fitting with sinusional functions*, *continuous fourier series*, *frequency and time domains*, *fourier integral and transform*, *discrete fourier transform*, *fast fourier transform*, dan *power spectrum*.

Penelitian ini merepakan pembentukan kurva dengan menggunakan metode *interpolating polynomials*. Metode yang digunakan yaitu *newton's divided-difference interpolating polynomial*.



Gambar 2.8. Curve Fitting

Sebagaimana dinyatakan diatas, ada berbagai bentuk alternatif untuk mengekspresikan bentuk dari interpolasi polinomial, *newton's divided-difference interpolating polynomials* merupakan bentuk yang paling populer dan paling sederhana. *Newton's divided-difference interpolating polynomial* memiliki 3 bentuk persamaan yang mana terdiri dari orde 1, orde 2 dan orde 3. Orde 1 juga dikenal dengan *linier* interpolasi. *Linear* interpolasi merupakan bentuk yang paling sederhana dengan cara menghubungkan dua buah titik data dengan sebuah garis lurus, bentuk dari *linear* interpolasi dari *newton's divided-difference interpolating polynomial* seperti ditunjukkan pada Gambar 2.5. Bentuk dari formula interpolasi linear seperti persamaan 2.

$$\frac{f_1(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (2)$$

Selanjutnya dapat disusun kembali menjadi persamaan 3.

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0} (x - x_0) \quad (3)$$

Notasi  $f_1(x)$  merupakan orde pertama dari interpolasi *polynomials*.  $\frac{f(x_1)-f(x_0)}{x_1-x_0}$  merupakan pendekatan yang digunakan sebagai pembatasan dari *divided-diffrence* dari *derivative* pertama, notasi ini merepresentasikan kemiringan garis yang menghubungkan antar titik satu ke titik yang lain. Secara umum, semakin kecil nilai interval antara titik-titik data, semakin baik pendekatan tersebut, hal ini disebabkan oleh nilai interval berkurang maka fungsi kontinu lebih baik mendekati dengan garis lurus.

*Lagrange interpolation polynomial* merupakan pengembangan formula persamaan dari *newton's divided-difference interpolating polynomial*, hanya saja pada *newton's divided-difference interpolating polynomial* terbatas hanya pada orde 3 sedangkan pada *Lagrange interpolation polynomial* dapat digunakan pada orde n, sehingga dapat dilakukan pendekatan nilai antar titik menjadi lebih luas. Formula dari *Lagrange interpolation polynomial* dirumuskan menjadi persamaan 4 dan persamaan 5.

$$f_n(x) = \sum_{i=0}^n L_i(x) f(x_i) \quad (4)$$

Dimana,

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad (5)$$

Dimana,  $\prod$  merupakan “product of”.

$$f_n(x) = \sum_{i=0}^n \left( \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \right) f(x_i) \quad (6)$$

Maka akan diperoleh bentuk persamaan dari *Lagrange interpolation polynomial* seperti persamaan 6. Sebagai contoh dari *Lagrange interpolation polynomial* dengan menggunakan orde 1 maka akan diperoleh formula matematis seperti persamaan 7.

$$f_1(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \quad (7)$$

Contoh persamaan dari *Lagrange interpolation polynomial* dengan menggunakan orde 2 maka akan diperoleh formula matematis seperti persamaan 8.

$$\begin{aligned} f_2(x) = & \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \\ & + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \end{aligned} \quad (8)$$

Contoh persamaan dari *Lagrange interpolation polynomial* dengan menggunakan orde 3 maka akan diperoleh formula matematis seperti persamaan 9.

$$\begin{aligned} f_3(x) = & \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} f(x_0) \\ & + \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} f(x_1) \\ & + \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} f(x_2) \\ & + \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} f(x_3) \end{aligned} \quad (9)$$

Contoh *Lagrange interpolation polynomial* dengan menggunakan orde 1 yang diterapkan pada data.

$$x_0 = 1 \quad f(x_0) = 80$$

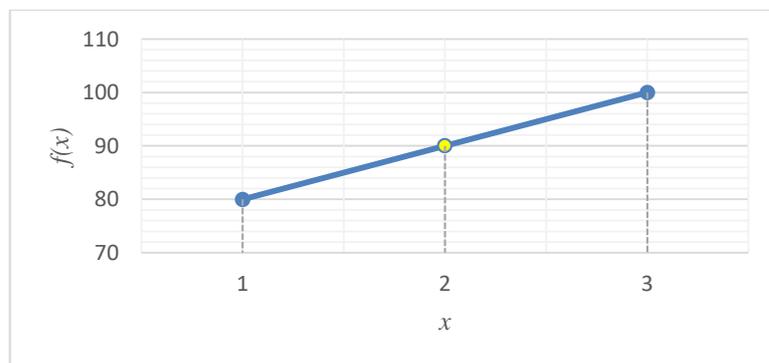
$$x_1 = 3 \quad f(x_1) = 100$$

Menentukan nilai  $f(x)$  dari  $x = 2$ , dari persamaan orde 1, maka diperoleh

$$f_1(2) = \frac{2 - 3}{1 - 3} 80 + \frac{2 - 1}{3 - 1} 100$$

$$f_1(2) = 90$$

Sehingga divisualisasikan kedalam bentuk grafik menjadi seperti pada Gambar 2.9. Titik berwarna biru adalah nilai pertama dan nilai kedua dari yang diketahui, sedangkan titik berwarna kuning merupakan nilai yang diperoleh dari hasil *Lagrange interpolation polynomial* orde 1.



Gambar 2.9. *Lagrange interpolation polynomial* orde 1

Contoh *Lagrange interpolation polynomial* dengan menggunakan orde 2 yang diterapkan pada data.

$$x_0 = 1 \quad f(x_0) = 80$$

$$x_1 = 2 \quad f(x_1) = 86$$

$$x_2 = 4 \quad f(x_2) = 100$$

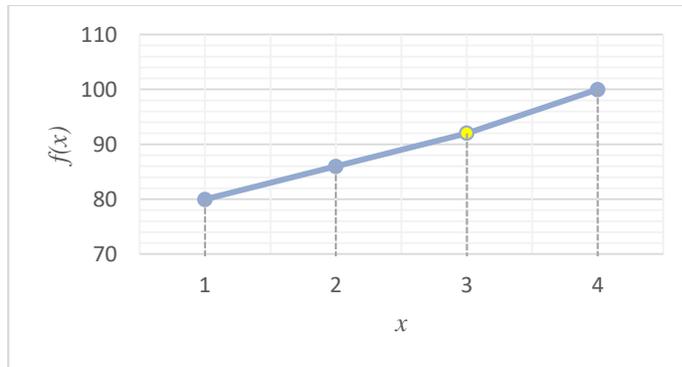
Menentukan nilai  $f(x)$  dari  $x = 3$ , dari persamaan 2, maka diperoleh

$$f_2(3) = \frac{(3 - 2)(3 - 4)}{(1 - 2)(1 - 4)} 80 + \frac{(3 - 1)(3 - 4)}{(2 - 1)(2 - 4)} 86 + \frac{(3 - 1)(3 - 2)}{(4 - 1)(4 - 2)} 100$$

$$f_2(3) = 92$$

Sehingga divisualisasikan kedalam bentuk grafik menjadi seperti pada Gambar 2.10. Titik berwarna biru adalah nilai pertama, nilai kedua dan nilai ketiga dari

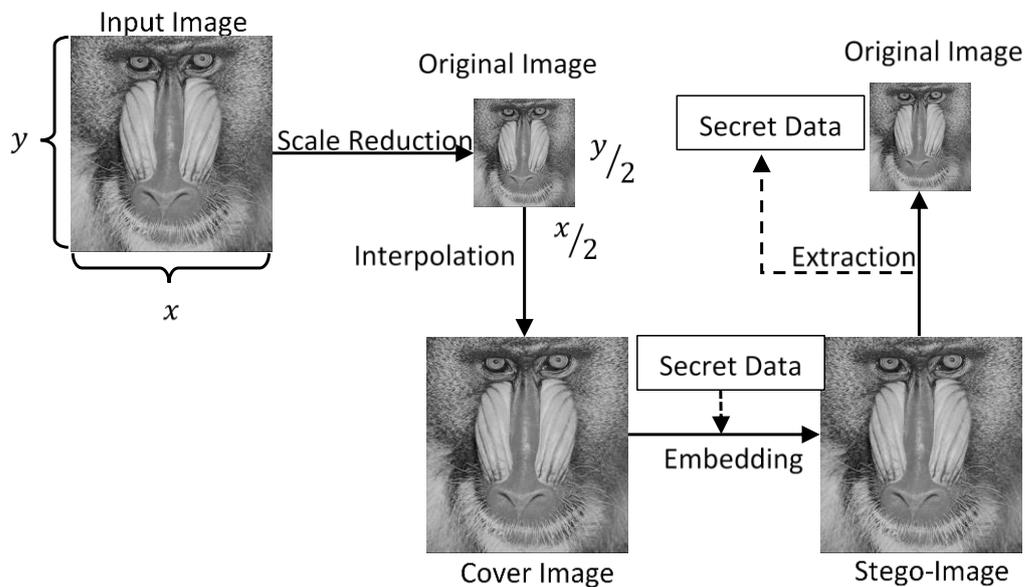
yang diketahui, sedangkan titik berwarna kuning merupakan nilai yang diperoleh dari hasil *Lagrange interpolation polynomial* orde 2.



Gambar 2.10. *Lagrange interpolation polynomial* orde 2

### 2.5 Jung & Yoo Method

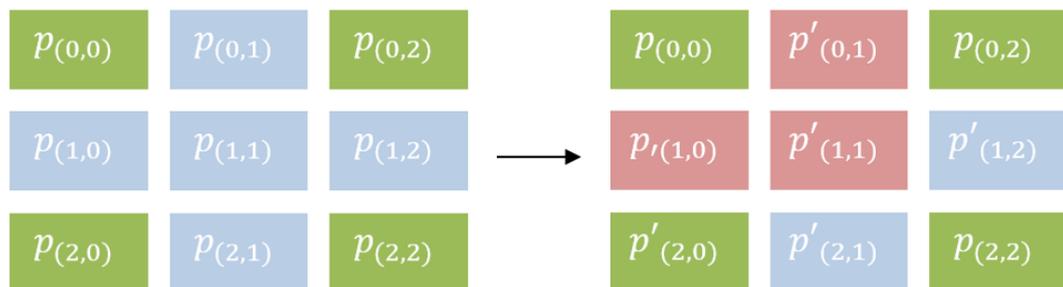
Pada tahun 2009 Jung dan Yoo mengusulkan metode baru *data hiding*, penelitian menggunakan berkas citra sebagai media pembawanya. Skema penyisipan yang disululkan dirancang dengan beberapa fungsi baru. Interpolasi digunakan untuk memeperoleh piksel baru yang digunakan sebagai tempat penyisipan. Skema yang diusulkan mereka seperti Gambar 2.11.



Gambar 2.12 Skema Jung & Yoo

Penelitian yang diajukan Jung dan Yoo memiliki beberapa tahapan, tahapan 1-6 merupakan tahapan *encoding*, tahapan 7 sampai 9 merupakan tahapan *decoding*. Tahapan-tahapan ini antara lain :

1. Tahapan pertama yaitu input berkas citra, pada penelitian menggunakan berkas citra satu *layer* dan berukuran 2 dimensi  $(x, y)$ .
2. Tahapan kedua, ukuran citra masukan direduksi sebesar setengah dari ukuran *input*. Reduksi ini diterapkan pada dimensi  $x$  dan dimensi  $y$ .
3. Tahapan ketiga, hasil dari reduksi ukuran skala ini selanjutnya disebut sebagai citra original yang memiliki ukuran lebih kecil dari pada citra *input*.
4. Tahapan keempat melakukan interpolasi dari citra original menjadi citra baru, hasil dari interpolasi ini menghasilkan citra yang berukuran sama dengan citra *input*, tujuan dari interpolasi ini menjadi ruang penyisipan data rahasia.
5. Tahapan kelima citra hasil dari interpolasi selanjutnya dijadikan *cover image* atau *host media* pembawa.
6. Tahapan keenam, dilakukan penyisipan data rahasia dengan menggunakan beberapa fungsi yang diusulkan.
7. Hasil dari penyisipan selanjutnya menjadi *stego-image*, yang sudah memiliki pesan didalam berkasnya.
8. Setelah jadi *stego-image* maka dilakukan ekstraksi pemisahan antara original *image* dan data rahasia.
9. Original *image* selanjutnya tetap berukuran setengah dari input citra *encoding*, sehingga tahap akhir ini original tidak digunakan. Penelitian Jung & Yoo melakukan evaluasi pengujian antara *stego-image* dengan *cover*, bukan *stego-image* dengan original *image*, hal ini karena ukuran dimensi yang berbeda.



Gambar 2.13 Interpolasi Jung & Yoo

Interpolasi yang digunakan pada penelitian Jung dan Yoo merujuk pada Gambar 2.13. Piksel asli dari citra original yaitu  $P(0,0), P(0,2), P(2,0), P(2,2)$  dan piksel interpolasi yaitu  $P(0,1), P(1,0), P(1,1), P(1,2), P(2,1)$ . Fungsi yang digunakan untuk interpolasi ini menggunakan persamaan (10).

$$p'(i, j) = \begin{cases} p(i, j), & \text{if } i = 2 \times m, j = 2 \times n \\ (p'(i, j-1) + p'(i, j+1))/2, & \text{if } i = 2 \times m, j = 2 \times n + 1 \\ (p'(i-1, j) + p'(i+1, j))/2, & \text{if } i = 2 \times m + 1, j = 2 \times n \\ (p(i-1, j-1) + p'(i-1, j) + p'(i, j-1))/3, & \text{otherwise} \end{cases} \quad (10)$$

Notasi  $p'(i, j)$  merupakan piksel hasil nilai interpolasi, sebagai contoh  $P(0,0)$  memiliki intensitas 46,  $P(0,2)$  memiliki intensitas 112,  $P(2,0)$  memiliki intensitas 210, beturut-turut piksel sebagai berikut  $P(0,0) = 36, P(0,2) = 112, P(2,0) = 210$ . Contoh penggunaan fungsi (10) untuk interpolasinya Jung dan Yoo seperti berikut:

$$\begin{aligned} p'(0,0) &= p(0,0) = 46 \\ p'(0,1) &= (46 + 112)/2 = 79 \\ p'(1,0) &= (46 + 210)/2 = 128 \\ p'(1,1) &= (46 + 79 + 128)/3 = 84 \end{aligned}$$

Pada proses *encoding* setelah tahapan interpolasi yaitu proses mencari nilai selisih antara dua piksel, nilai selisih ( $d$ ) ini menjadi nilai paramater untuk penentuan banyaknya data rahasia yang akan disisipkan. Formula yang digunakan untuk memperoleh nilai ( $d$ ) seperti berikut (11).

$$d(i, j) = p'(i, j) - p'(orientation) \quad (11)$$

dimana  $p'(orientation)$  adalah piksel acuan yang tidak mengalami perubahan nilai, dari persamaan (11) diperoleh nilai seperti berikut

$$\begin{aligned} d(0,1) &= 79 - 46 = 33 \\ d(1,0) &= 128 - 46 = 82 \\ d(1,1) &= 84 - 46 = 38 \end{aligned}$$

Tahapan *encoding* selanjutnya penentuan banyak bit ( $n$ ) yang dapat disispkan pada piksel interpolasi. Penentuan jumlah bit ini menggunakan persamaan (12).

$$n(i, j) = \log_2 |d(i, j)| \quad (12)$$

sehingga diperoleh banyak bit pada masing-masing piksel interpolasi seperti berikut.

$$n(0,1) = \log_2|33| = 5$$

$$n(1,0) = \log_2|82| = 6$$

$$n(1,1) = \log_2|38| = 5$$

Pada contoh penelitian Jung dan Yoo diberikan data *stream* rahasia seperti berikut,

$$\text{secret bits} = 1001101011000101_2$$

selanjutnya proses pemotongan data rahasia sesuai dengan jumlah bit ( $b$ ) yang telah diperoleh dari nilai ( $n$ ) dan dilanjutkan konversi bit menjadi nilai integer. Hasil dari proses ini diperoleh seperti berikut.

$$b(0,1) = (10011)_2 \rightarrow 19$$

$$b(1,0) = (010110)_2 \rightarrow 22$$

$$b(1,1) = (00101)_2 \rightarrow 5$$

Bagian akhir yaitu mencari nilai piksel baru ( $\overline{p'(i,j)}$ ), untuk memperoleh nilai piksel baru, proses ini melakukan penjumlahan piksel interpolasi dengan nilai ( $b$ ) dengan menggunakan persamaan (13).

$$\overline{p'(i,j)} = p'(i,j) + b(i,j) \quad (13)$$

hasil piksel baru yang diperoleh dari persamaan (11) seperti berikut,

$$\overline{p'(0,1)} = 79 + 19 = 98$$

$$\overline{p'(1,0)} = 128 + 22 = 150$$

$$\overline{p'(1,1)} = 84 + 5 = 89$$

Proses *decoding* yang dilakukan pada penelitian Jung & Yoo relatif lebih mudah dan sederhana, perhitungan untuk mengekstraksi proses ini sebagai berikut.

$$b(0,1) = 98 - \left\lfloor \frac{46 + 112}{2} \right\rfloor = 19 \rightarrow (10011)_2$$

$$b(1,0) = 150 - \left\lfloor \frac{46 + 210}{2} \right\rfloor = 22 \rightarrow (010110)_2$$

$$b(1,1) = 89 - \left\lfloor \frac{46 + 112 + 210}{3} \right\rfloor = 5 \rightarrow (00101)_2$$

Jika dengan skema dan proses penyisipan yang sama tetapi berkas *cover* yang digunakan berbeda seperti penelitian Jung dan Yoo berupa berkas citra

sedangkan pada penelitian ini menggunakan berkas audio, maka dibutuhkan penyesuaian beberapa fungsi yang digunakan, baik untuk proses *encoding* dan *decoding*, hal ini dikarenakan dimensi berkas yang digunakan berbeda jumlahnya. Pada citra menggunakan dua dimensi dan pada berkas audio menggunakan satu dimensi, sehingga persamaan Jung dan Yoo menjadi seperti persamaan (14), (15), (16), (17), dan (19).

$$y_n = \frac{y_{n-1} + y_{n+1}}{2} \quad (14)$$

Persamaan (14) digunakan untuk memperoleh signal baru,  $y_n$  merupakan signal interpolasi,  $y_{n-1}$  adalah signal yang berindeks sebelum signal interpolasi,  $y_{n+1}$  adalah signal yang berindeks setelah signal interpolasi.

$$d = y_n - y_{n-1} \quad (15)$$

Persamaan (15) digunakan untuk mencari nilai selisih antara kedua signal dan menjadi nilai faktor penentu jumlah bit yang disisipkan.

$$n = \log_2^{|d|} \quad (16)$$

Persamaan (16) digunakan untuk mencari banyaknya bit yang disisipkan. Persamaan (17) dan (18) digunakan untuk konversi bit menjadi desimal.

$$b = n \text{ (bit sec)} \quad (17)$$

$$b = \text{bit2dec}(n) \quad (18)$$

Persamaan (19) digunakan untuk memperoleh signal baru yang terdiri dari signal gabungan dan data rahasia.

$$y'_n = y_n + b \quad (19)$$

Proses pengembalian data audio asli dan data rahasia menggunakan persamaan (20) dan (21).

$$b = y'_n - \frac{y_{n-1} + y_{n+1}}{2} \quad (20)$$

$$n = \text{dec2bin}(b) \quad (21)$$

Metode Jung & Yoo ini menjadi rujukan penelitian dan digunakan sebagai metode pembandingan.

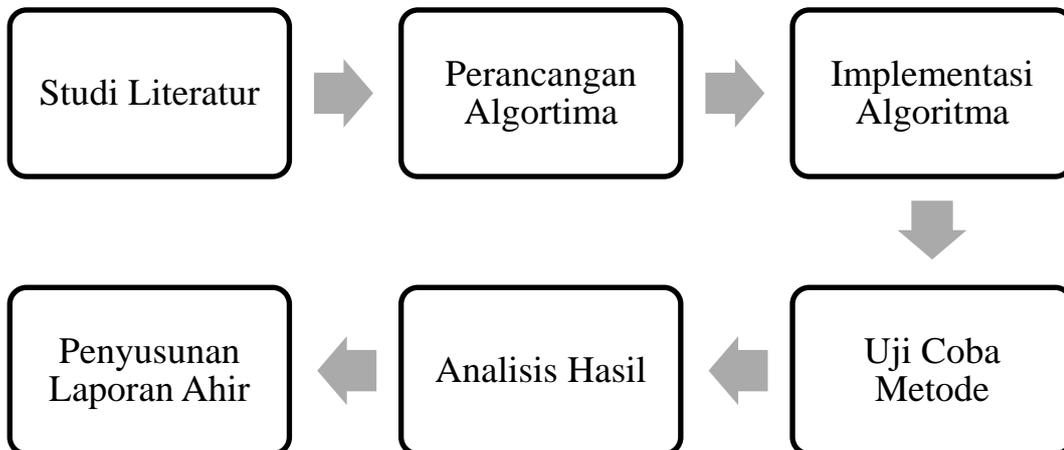
## BAB 3

### METODE PENELITIAN

Bab 3 berisi pembahasan mengenai langkah penelitian dan jadwal kegiatan penelitian. Langkah penelitian dibagi menjadi enam tahap yaitu studi literatur, Perancangan algoritma, implementasi algoritma, ujicoba metode, analisis hasil dan penyusunan laporan akhir. Selanjutnya jadwal kegiatan penelitian memuat durasi waktu dari semua kegiatan penelitian selama empat bulan.

#### 3.1 Langkah-langkah Penelitian

Pada penelitian ini, terdapat beberapa tahapan penyelesaian yang akan dilakukan. Tahapan tersebut dijelaskan pada Gambar 3.1. Secara umum, luaran yang diharapkan dari penelitian ini adalah metode penyisipan data adaptif pada berkas audio yang mampu menghasilkan *stego* audio dengan kualitas yang baik dan kapasitas memiliki kapasitas penyisipan yang besar. Adapaun langkah yang digunakan dijelaskan pada subbab berikutnya.



Gambar 3.1 Diagram Alur Kerja Penelitian

#### 3.2 Studi Literatur

Proses ini meliputi pencarian referensi pendukung yang sesuai dengan topik yang diangkat dan dapat dipertanggungjawabkan secara keilmuan. Referensi yang digunakan untuk penelitian ini berupa buku, jurnal, maupun artikel. Informasi yang didapat nantinya berguna untuk mendesain algoritma untuk memecahkan permasalahan yang sedang diteliti. Setelah studi literatur dilakukan, pendefinisian

masalah dapat diketahui lebih deatail. Adapun yang menjadi topik bahasan dalam penelitian ini antara lain:

1. Metode penyisipan pada berkas audio
2. Metode dan penerapan interpolasi untuk penyisipan baik pada berkas audio maupun berkas citra.
3. Metode penyisipan data yang *reversible* pada berkas audio agar informasi penyisipan dapat diperoleh kembali.
4. Metode adaptif yang dapat digunakan pada penyisipan berkas audio untuk memperoleh nilai maksimum dari berbagai parameter yang digunakan.
5. Model matematika untuk memperbaiki kualitas yang dihasilkan.
6. Model matematika untuk meningkatkan kapatisas.

### 3.3 Perancangan Algoritma

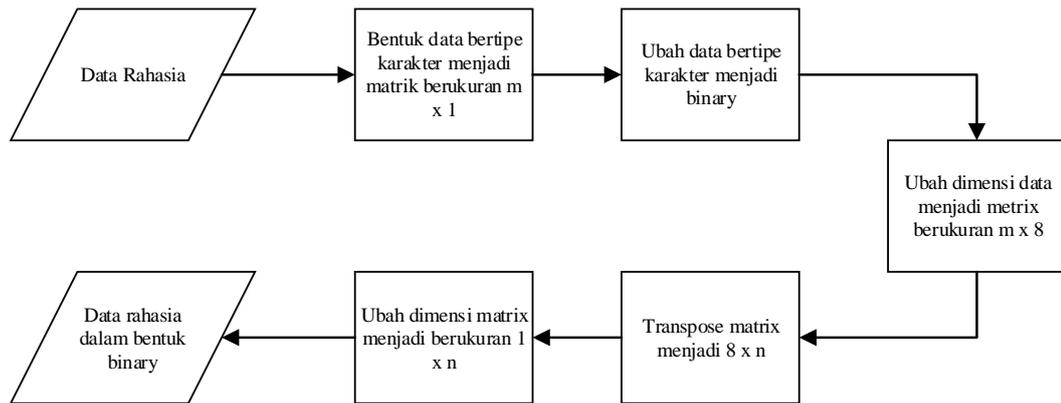
Pada sub bab perancangan algoritma akan dijelaskan proses utama dari usulan penelitian yang diajukan. Proses yang dilakukan untuk mengimplementasikan usulan, usulan ini digambarkan melalui diagram alur agar mempermudah pemahaman dan penerapan metode *data hiding* pada audio yang diusulkan.

Terdapat dua bagian utama yang terpenting dalam proses *data hiding* audio yang diusulkan yaitu antralain: proses *encoding* dan proses *decoding*. Proses *encoding* juga dikenal dengan proses penyisipan data *payload* pada media pembawa dalam hal ini berkas audio. *Preprocessing payload* merupakan bagian awal pada tahapan usulan penelitian. *Preprocessing payload* merupakan bagian pengolahan data teks menjadi *binary*. Proses *decoding* juga dikenal dengan proses ekstraksi data *payload* dari media pembawa. *Postprocess payload* merupakan bagian akhir pada tahapan usulan penelitian. *Postprocess* bertujuan untuk mengolah data *payload* berbentuk *binary* menjadi teks.

Penjelasan mengenai *preprocessing payload*, proses *encoding*, proses *decoding* serta *postprocessing payload* dari metode yang diusulkan akan dipaparkan kedalam subbab penyisipan pesan dan subbab ekstraksi pesan.

### 3.1.1 Preprocessing Payload

Pada subbab ini menjelaskan tentang rancangan metode penyisipan pesan yang diusulkan. Pada tahapan awal dijelaskan mengenai skema data *payload* yang digunakan. Tahapan proses ini berperan untuk menyiapkan data rahasia dari teks menjadi *payload* berbentuk *binary*, adapun prosesnya seperti berikut.



Gambar 3.2 Bagan Data Payload

Pada tahap pertama menjelaskan tentang data rahasia yang digunakan, data rahasia atau *payload* yang digunakan tidak hanya melalui proses perubahan dari karakter menjadi biner, tetapi ada beberapa tahapan yang dilalui. Tahapan-tahapan ini bertujuan agar meningkatkan keamanan dari data tersebut agar tidak mudah diketahui oleh pihak yang tidak berkepentingan. Tahapan ini digambarkan sesuai dengan Gambar 3.2. Adapun tahapannya antara lain:

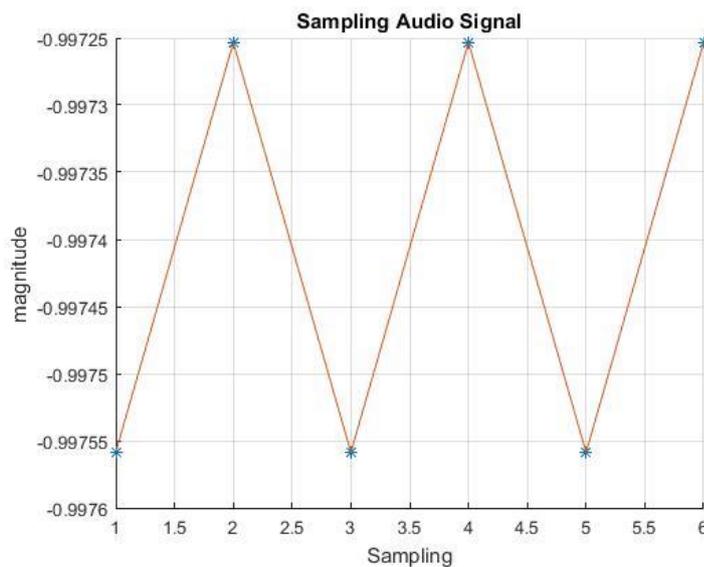
1. Menghitung jumlah karakter yang digunakan sebagai data rahasia. Pada contoh karakter yang digunakan sebanyak sebelas karakter. Setelah perhitungan karakter maka selanjutnya merubah karakter tersebut menjadi sebuah matrik seukuran  $m \times 1$ , maka akan terbentuk *matrix* ukuran sebesar  $11 \times 1$ .
2. Tahapan ketiga merubah seluruh karakter data rahasia menjadi biner, atau proses ini disebut binerisasi, proses ini merubah karakter menjadi bilangan 0 dan sesuai dengan ASCII masing-masing karakter.
3. Setelah masing-masing karakter menjadi bilangan biner maka selanjutnya merubah matrik menjadi menjadi berukuran  $m \times 8$ , angka 8 diperoleh dari perubahan karakter menjadi binari 8 bit, sehingga diperoleh matrik baru sebesar  $11 \times 8$ .

4. Tahapan kelima yaitu melakukan *transpose* matrik dari data rahasia sebelumnya menjadi matrik  $8 \times n$ . Sehingga diperoleh matrik baru berukuran  $8 \times 11$ .
5. Tahapan keenam melakukan perubah ukuran matriks dari  $8 \times n$  menjadi  $1 \times n$ . Sehingga dari contoh akan diperoleh perubahan matrik dari  $8 \times 11$  menjadi  $1 \times 88$ .
6. Bagian akhir data rahasia siap digunakan dan sudah berbentuk *binary*.

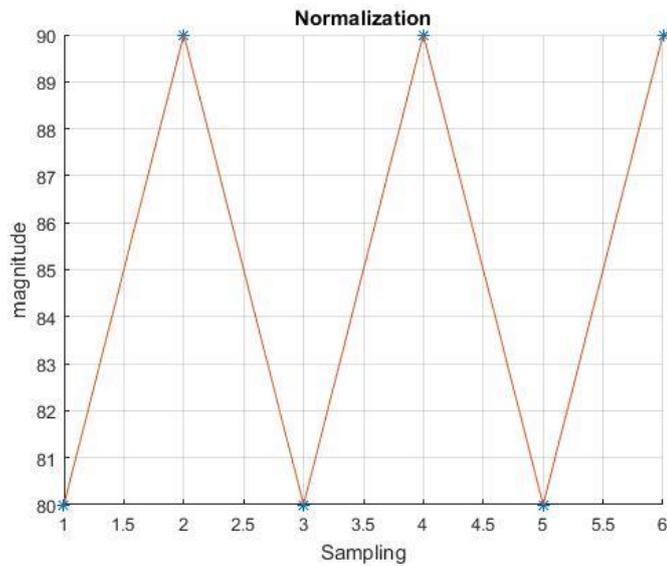
### 3.1.2 Penyisipan Pesan

Pada bagian kedua merupakan inti proses penyisipan data rahasia, bagian ini menjelaskan tahapan mengenai sekema penyisipan yang diusulkan. Ada beberapa hal penting terkait tujuan dari penelitian pada bagian kedua ini, antarlain : Peningkatan kapasitas yang dilakukan dengan cara meningkatkan ruang penyisipan pada signal audio, peningkatan kapasitas dengan menggunakan persamaan yang diusulkan, membuat *pivoting point* sebagai pengganti *location maps*, serta perbaikan kualitas stego audio.

Secara umum metode yang diusulkan seperti ditunjukkan pada Gambar 3.3 – Gambar 3.8. Pada Gambar 3.3 menunjukkan proses *sampling* audio setelah proses pembacaan berkas audio.

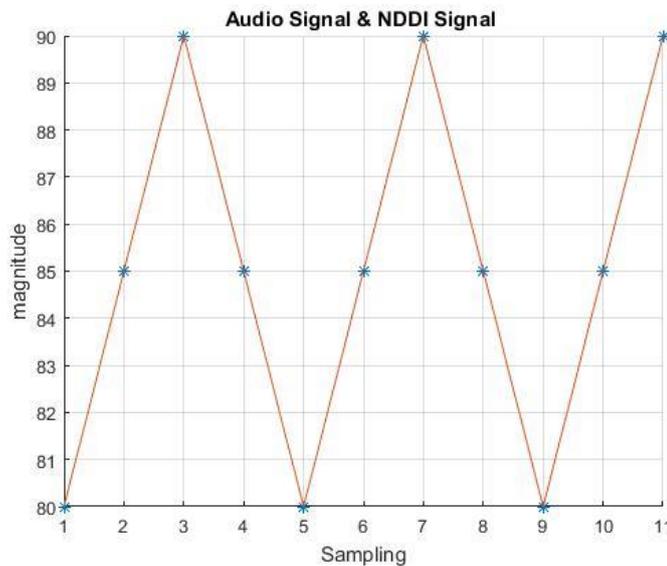


Gambar 3.3 Sampling Audio



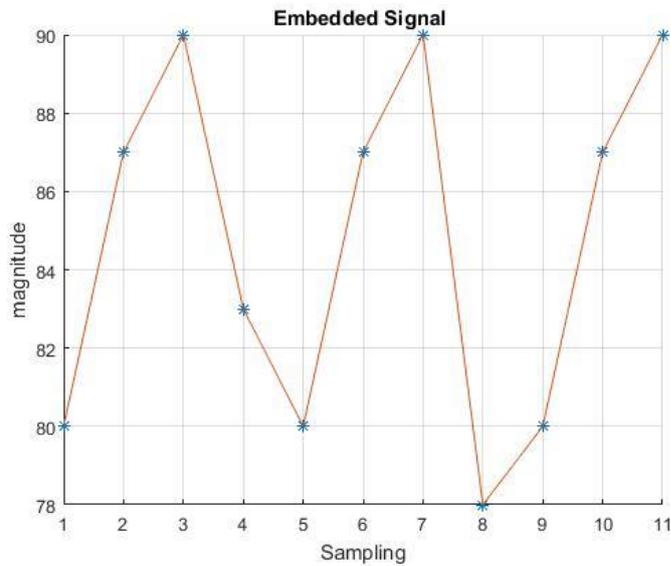
Gambar 3.4 Normalisasi

Tahapan selanjutnya adalah proses normalisasi seperti pada Gambar 3.4, proses normalisasi ini bertujuan merubah tipe data audio agar bisa disisipkan.



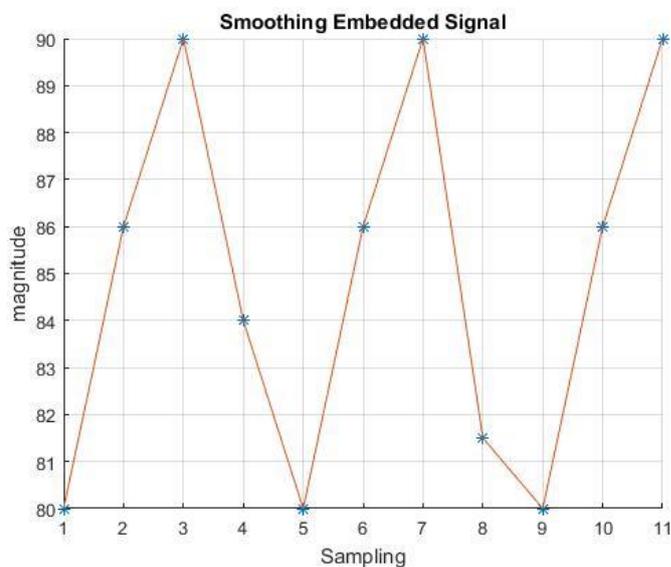
Gambar 3.5 NDDI Proses

Bagian ketiga melakukan interpolasi signal dari signal asli menjadi satu signal baru dengan menggunakan metode NDDI, signal yang selanjutnya dijadikan ruang penyisipan dan *pivoting point*. Hasil dari proses seperti pada Gambar 3.5



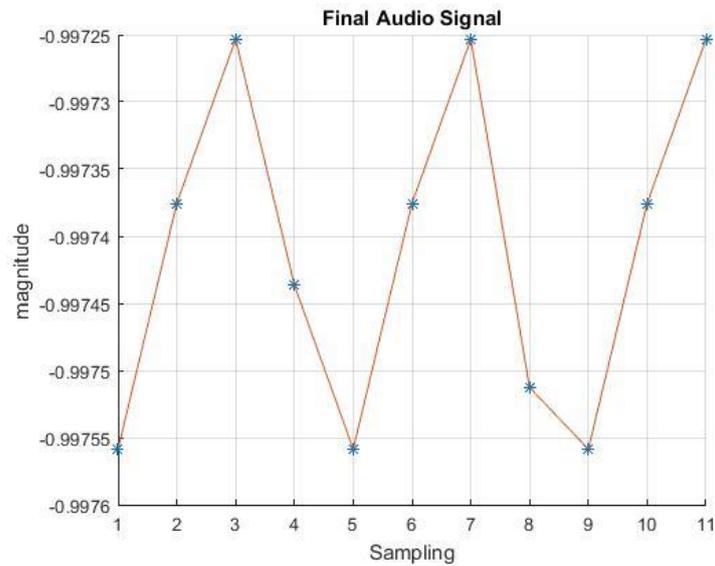
Gambar 3.6 Penyisipan Payload

Tahapan selanjutnya penyisipan data, dalam proses penyisipan skema yang lebih kompleks dan dibahas secara mendetail pada bagian selanjutnya. Hasil signal penyisipan seperti ditunjukkan pada Gambar 3.6



Gambar 3.7 Reduksi Signal

Agar kualitas signal sisipan menjadi lebih baik maka diperlukan proses reduksi signal atau disebut dengan *smoothing* proses, proses ini memiliki persamaan matematis untuk mereduksi signal, persamaan ini diperoleh dari pengembangan persamaan garis. Hasil dari proses reduksi ditunjukkan pada Gambar 3.7.

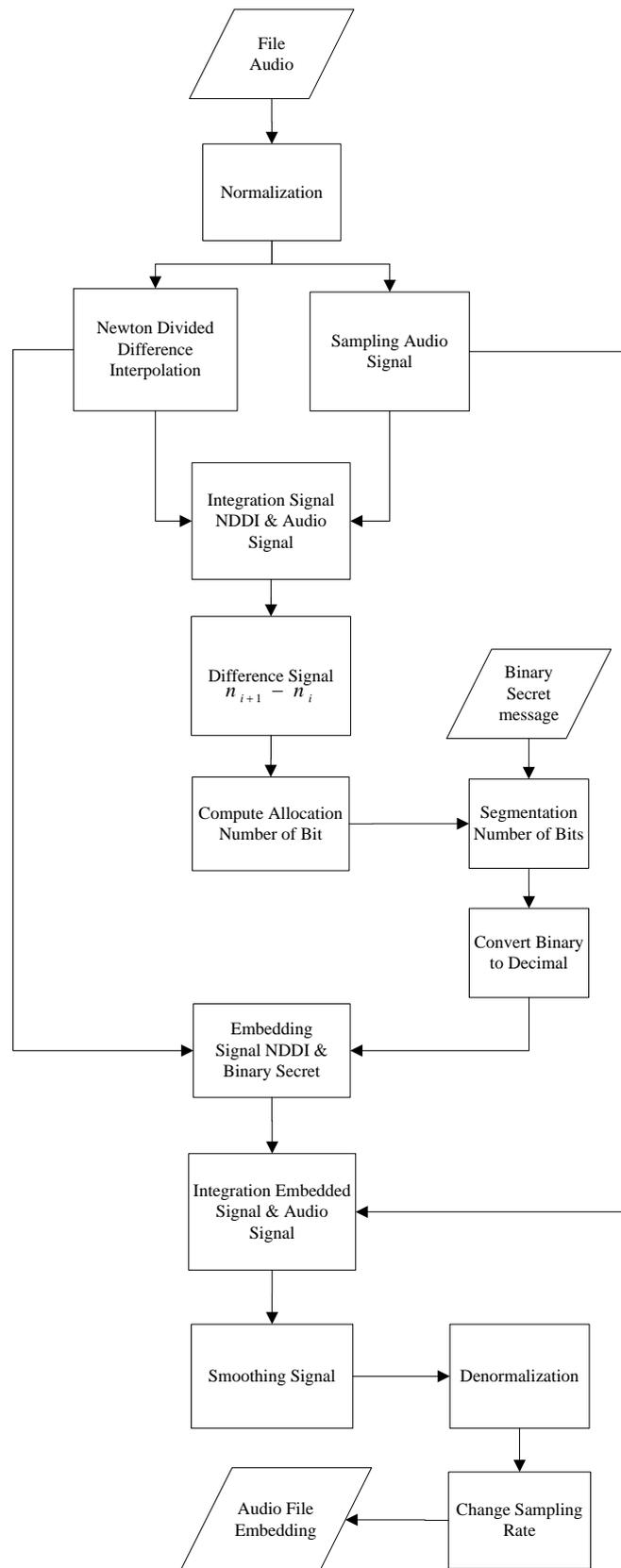


Gambar 3.8 Stego-Audio

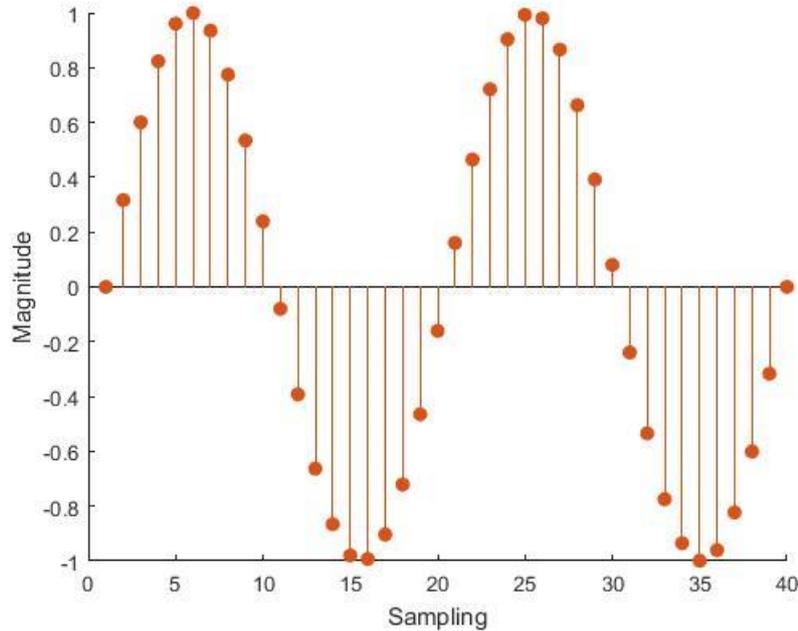
Bagian paling akhir ialah denormalisasi signal yang siap dibaca oleh *player* audio, hasil keluaran dari proses ini disebut stego-audio. Hasil dari proses ini seperti pada Gambar 3.8.

Setelah digambarkan proses *encoding* secara umum, selanjutnya dijelaskan proses *encoding* secara spesifik. Tahapan dari proses *encoding* secara spesifik seperti Gambar 3.9. Ada dua berkas input dari proses ini antarlain berkas audio serta berkas data *stream binary payload*. Penjelasan mengenai penyisipan memiliki beberapa tahapan proses antarlain :

1. Input file audio, *input* dari penelitian ini berupa berkas audio, berkas yang digunakan berupa *waveform* audio format dengan berekstensi *.wav*. menggunakan jenis berkas ini karena berkas *wav* tidak mengalami kompresi dan murni hasil PCM sehingga tidak memunculkan kekuatiran untuk kehilangan jumlah *sample* data. Berkas audio yang menjadi input dan telah *disampling* dan memiliki magnitude diilustrasikan pada Gambar 3.10.
2. Normalisasi, setelah berkas audio dibaca selanjutnya proses normalisasi, proses ini merubah tipe data digunakan dari integer 16 (-32768 - 32767) menjadi unsigned integer 16 (0 - 65535).
3. *Sampling* audio signal merupakan hasil pembacaan berkas audio yang telah ternormalisasi. Sebagai contoh 3 buah *sampling* yang telah normalisassi sebagai berikut  $y_1 = 80, y_2 = 90, y_3 = 80$



Gambar 3.9 Proses *Encoding*



Gambar 3.10 Proses *Sampling*

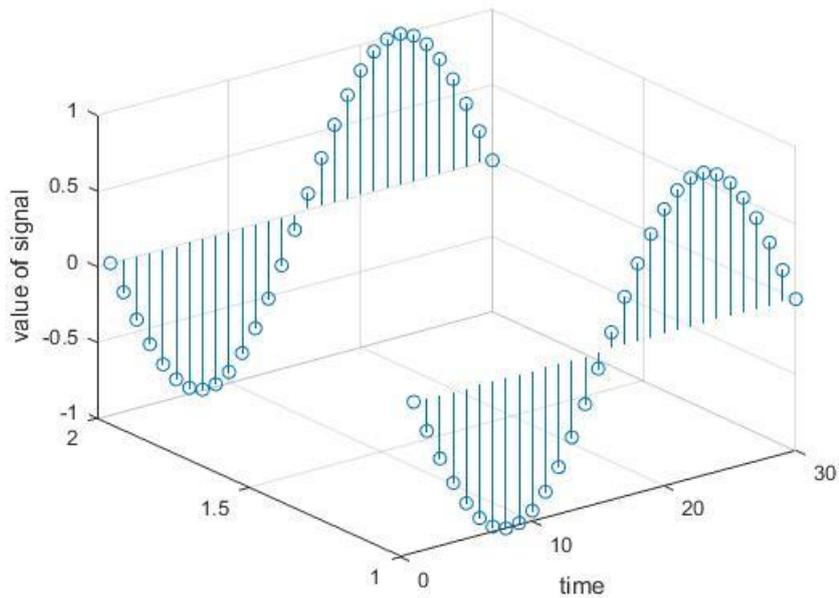
4. Tahap NDDI, implementasi persamaan (22) untuk memperoleh signal hasil NDDI. Signal ini yang selanjutnya menjadi *pivoting point* atau titik acuan penyisipan data sebagai pengganti peran *location maps*. Pada signal ini menjadi ruang menyisipan sehingga penyisipan tidak pada signal audio asli. Pada tahap ini signal audio asli akan mengalami pergeseran karena signal hasil NDDI akan berada pada kedua signal, sehingga input  $y_1$  tidak mengalami perubahan,  $y_2$  menjadi  $y_3$ , serta  $y_3$  menjadi  $y_4$ , dengan kata lain signal audio berada pada indeks ganjil dan signal NDDI berada pada indeks genap.

$$y_n = \frac{x_n - x_{n+1}}{x_{n-1} - x_{n+1}} y_{n-1} + \frac{x_n - x_{n-1}}{x_{n+1} - x_{n-1}} y_{n+1} \quad (22)$$

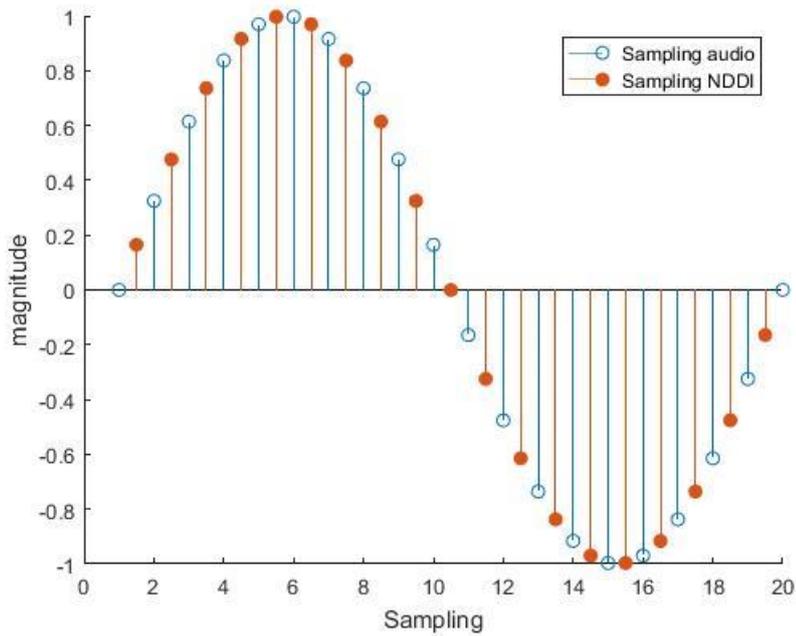
Notasi  $y_n$  merupakan besar nilai NDDI dan  $x_n$  indeks dari signal dan Ilustrasi proses pembentukan signal NDDI seperti pada Gambar 3.11, pada ilustrasi ini menunjukkan proses terpisah antara signal asli dan signal NDDI. Setelah di peroleh signal NDDI dan pergeseran indeks maka hasil menyusun signal seperti pada Gambar 3.12. Signal NDDI diperoleh sebagai berikut.

$$y_2 = \frac{2 - 3}{1 - 3} 80 + \frac{2 - 1}{3 - 1} 90 \rightarrow y_2 = 85$$

$$y_4 = \frac{4 - 5}{3 - 5} 84 + \frac{4 - 3}{5 - 3} 80 \rightarrow y_4 = 85$$



Gambar 3.11 Proses Pemisahan *Sampling*



Gambar 3.12 Hasil *Sampling* Audio dan *Sampling* NDDI

5. *Integration* signal NDDI & audio signal, proses ini proses penggabungan antara kedua signal. Proses ini menjadi sebuah vektor baru yang terdiri gabungan signal, sehingga vektor tersebut menjadi seperti formula (23).

$$[y_1, \bar{y}_2, y_3, \bar{y}_4, \dots, \dots, y_n, (y_{n+1})] \quad (23)$$

notasi  $y_n$  merupakan signal audio asli dan  $\bar{y}_{n+1}$  merupakan signal NDDI. Sehingga pada tahap ini signal baru menjadi [80, 85, 90, 85, 80].

6. *Difference* signal, tahapan selanjutnya mencari selisih antara kedua signal, tahap ini bertujuan untuk memperoleh parameter penentu besarnya jumlah bit pada tahapan selanjutnya. Persamaan yang digunakan untuk memperoleh nilai selisih dengan persamaan (24). Pengurangan ini menggunakan nilai mutlak sehingga diperoleh nilai yang bulat.

$$d_n = [|y_{n-1} - y_n|] \quad (24)$$

Dari persamaan di atas diperoleh

$$d_1 = [|85 - 80|] \rightarrow d_1 = 5$$

$$d_2 = [|85 - 90|] \rightarrow d_2 = 5$$

7. *Compute allocation number of bit*, pada tahap ini menjadi tahapan penentuan banyaknya bit yang dapat disisipkan pada signal NDDI, persamaan (25) merupakan persamaan dari metode Jung dan Yoo, penelitian ini juga mengembangkan beberapa persamaan antara lain persamaan (26), (27), (28), (29). Tahapan ini juga mengevaluasi kinerja yang dihasilkan oleh masing-masing persamaan seperti pada Gambar 3.12. Hasil dari Gambar 3.13 menunjukkan persamaan (29) menjadi persamaan yang memperoleh paling banyak dapat menyisipkan bit rahasia. Sehingga persamaan (29) menjadi usulan yang digunakan pada penelitian ini.

$$j_n = \log_2[d_n] \quad (25)$$

$$j_n = \log_2[d_n] + \log_2[d_n] \quad (26)$$

$$j_n = \log_2^{[d_n]} + \log_2[\log_2[d_n]] \quad (27)$$

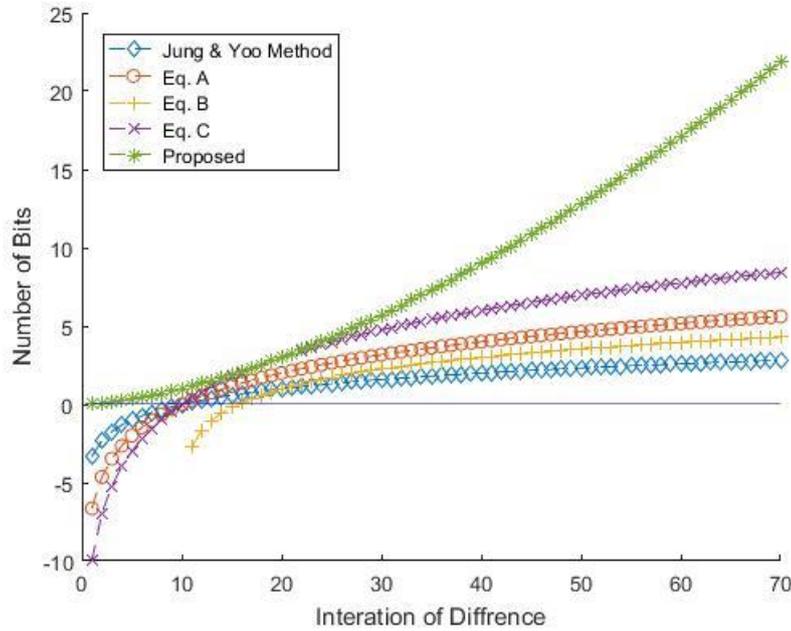
$$j_n = x * \log_2[d_n], \{x|x = 3\} \quad (28)$$

$$j_n = x^{\log_2[d_n]}, \{x|x = 2\} \quad (29)$$

Pada tahapan ini diperoleh banyaknya bit pada setiap signal NDDI adalah

$$j_1 = 2^{\log_2[5]} \rightarrow j_1 = 4$$

$$j_2 = 2^{\log_2[5]} \rightarrow j_2 = 4$$



Gambar 3.13 Evaluasi Kinerja Fungsi Parameter Jumlah Bit

8. *Input binary secret*, pada bagian ini merupakan input dari proses *preparation payload*, dimisalkan input bit rahasia  $sec = (00100111)_2$
9. *Segmen number of bits*, bagian ini melakukan pemotongan bit data rahasia dari *bit stream payload*. Pemotongan bit menggunakan formula (30)

$$b_n = j_n (\text{bit secret}) \quad (30)$$

diperoleh sebagai berikut.

$$b_1 = (0010)_2$$

$$b_2 = (0111)_2$$

10. *Convert binary to decimal*, setelah diperoleh segmen dari data rahasia selanjutnya melakukan konversi dari *binary* menjadi desimal dengan menggunakan formula 31.

$$b_n' = \text{bit2dec}(j_n) \quad (31)$$

Berturut-turut diperoleh hasil seperti berikut.

$$b_1' = \text{bit2dec}(0010)_2 \rightarrow b_1' = 2$$

$$b_2' = \text{bit2dec}(0111)_2 \rightarrow b_2' = 7$$

11. *Embedding signal NDDI & binary secret*, pada tahapan ini merupakan inti dari penyisipan antara signal NDDI dengan bit rahasia yang telah diubah menjadi desimal, keluaran dari signal ini menghasilkan signal baru  $\overline{y_{n+1}}'$ , penyisipan

ini memiliki 3 kondisi, syarat kondisi ini disesuaikan dengan pola trend yang dihasilkan antara signal audio asli ( $y_n$ ), dan signal NDDI ( $\overline{y_{n+1}}$ ), apabila kedua signal ini memiliki trend menurun maka menggunakan persamaan (33), sedangkan ketika tren yang diperoleh menaik maka signal sisipan menggunakan persamaan (32)

$$\overline{y_n}' = \overline{y_n} - b, \text{ jika } y_n > \overline{y_{n+1}} \quad (32)$$

$$\overline{y_n}' = \overline{y_n} + b, \text{ jika } y_n < \overline{y_{n+1}} \quad (33)$$

$$\overline{y_n}' = \overline{y_n}, \text{ jika } y_n = \overline{y_{n+1}} \quad (34)$$

Persamaan (34) digunakan ketika magnitude signal audio sama dengan signal NDDI. Penyisipan signal yang diperoleh dari contoh seperti berikut,

$$\overline{y_1}' = 85 + 2 \rightarrow \overline{y_1}' = 87$$

Pada signal sisipan yang pertama menggunakan persamaan (33) karena  $y_n < \overline{y_{n+1}} \rightarrow 80 < 85$ .

$$\overline{y_2}' = 85 - 7 \rightarrow \overline{y_1}' = 78$$

Pada signal sisipan yang kedua menggunakan persamaan (32) karena  $y_n > \overline{y_{n+1}} \rightarrow 90 > 85$ .

12. *Integration embedded* signal & audio signal, hasil dari tahapan sebelumnya digabung menjadi satu *vector* dengan menggunakan persamaan (23), berikut ini hasil dari signal yang telah disisipkan dan digabungkan.

$$[80, 85, 90, 85, 80] \rightarrow [80 \ 87 \ 90 \ 78 \ 80]$$

13. *Smoothing* signal, karena hasil dari signal sisipan masih memiliki selisih yang besar maka diperlukan proses *smoothing* signal, tujuan dari proses ini untuk memperbaiki kualitas stego yang diperoleh. Ide dari *smoothing* proses ini dikembangkan dari persamaan garis. Ada beberapa persamaan yang dikembangkan antara lain persamaan (35), (36), (37), (38) dan (39). Proses *smoothing* ini tidak merubah makna dari data sisipan yang telah disisipkan. Persamaan (35) dikembangkan untuk mereduksi signal yang dihasilkan setengah dari signal NDDI dan signal sisipan. Persamaan (36) dikembangkan untuk mereduksi signal yang di hasilkan 75% dari signal NDDI dan signal sisipan. Persamaan (37) dikembangkan untuk mereduksi signal yang dihasilkan 87.5% dari signal NDDI dan signal sisipan. Persamaan (38) dikembangkan untuk mereduksi

signal yang dihasilkan 93.75% dari signal NDDI dan signal sisipan. Ilustrasi proses reduksi signal sisipian (*smoothing* proses) ditunjukkan pada Gambar 3.14.

$$\gamma_{1,n} = \frac{\alpha_{1,n} + \beta_n}{2} \quad (35)$$

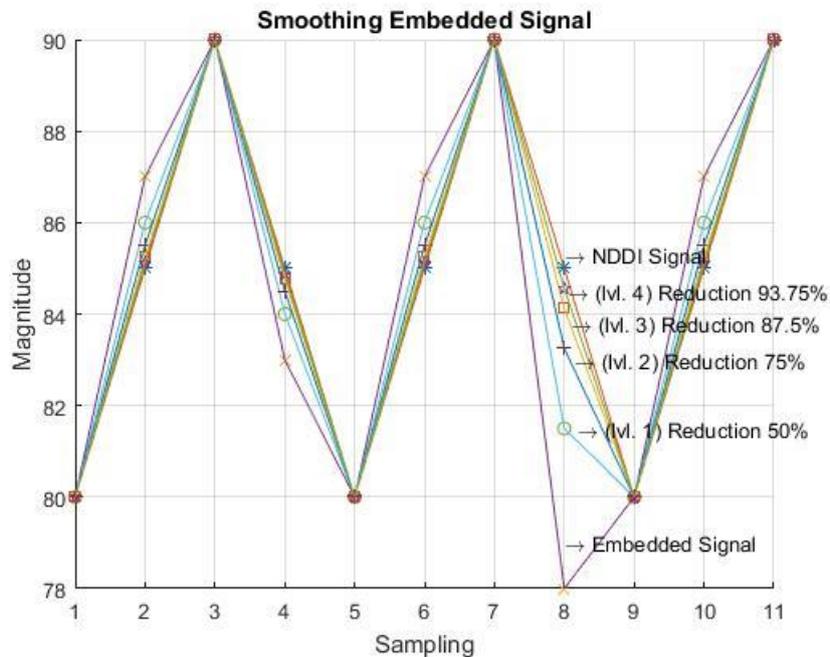
$$\gamma_{2,n} = \frac{\gamma_{1,n} + \beta_n}{2} \quad (36)$$

$$\gamma_{3,n} = \frac{\gamma_{2,n} + \beta_n}{2} \quad (37)$$

$$\gamma_{4,n} = \frac{\gamma_{3,n} + \beta_n}{2} \quad (38)$$

$$\gamma_{j,n} = \frac{\alpha_{j,n} + (2^j - 1) * \beta_n}{2^j} \quad (39)$$

Indeks( $j$ ) yang digunakan pada persamaan (35) – persamaan (39) merupakan indeks level reduksi. Indeks( $n$ ) yang merupakan indeks sample. Semakin tinggi levelnya maka semakin banyak signal sisipan yang direduksi. Notasi  $\gamma_{j,n}$  signal baru hasil reduksi,  $\alpha_{j,n}$  merupakan signal hasil sisipan dalam hal ini  $\overline{y}_n'$ , dan  $\beta_n$  merupakan signal NDDI ( $\overline{y}_{n+1}$ ).



Gambar 3.14 Proses *Smoothing*

Persamaan (35) diperoleh dari penjumlahaan antara signal sisipan dan signal NDDI kemudian dibagi dua. Hasil dari persamaan (35) menjadi nilai masukan

pada persamaan (36) dengan proses yang sama dengan persamaan (35). Proses ini berlangsung sesuai dengan indeks level reduksi yang diinginkan. Oleh karena itu persamaan (39) memiliki fungsi dan tujuan yang sama dengan persamaan (35) hingga persamaan (38), hanya saja persamaan (39) diperoleh dari pemodelan kembali antara persamaan (35) hingga persamaan (28).

Pada penelitian ini menggunakan indeks level  $j = 1$ , artinya penelitian ini berusaha memperbaiki kualitas dengan mereduksi signal sebesar 50%. Hasil signal NDDI dan hasil signal sisipan berturut-turut seperti berikut [80, **85**, 90, **85**, 80]  $\rightarrow$  [80 **87** 90 **78** 80] dengan persamaan (39) diperoleh

$$\bar{y}_1' = 87 \rightarrow \gamma_{1,1} = \frac{87 + (2^1 - 1) * 85}{2^1} \rightarrow \gamma_{1,1} = 86$$

$$\bar{y}_2' = 78 \rightarrow \gamma_{1,1} = \frac{78 + (2^1 - 1) * 85}{2^1} \rightarrow \gamma_{1,1} = 81.5$$

sehingga besar signal akhir yang telah sisipi dan telah direduksi seperti berikut [80 **87** 90 **78** 80]  $\rightarrow$  [80 **86** 90 **81.5** 80]. Hasil reduksi ini memiliki jarak yang lebih dekat dengan signal NDDI. Tanapa dilakuakn reduksi maka jarak antar signal NDDI dan signal sisipan seperti berikut.

$$\Delta_1 = |87 - 85| \rightarrow \Delta_1 = 2$$

$$\Delta_2 = |78 - 85| \rightarrow \Delta_2 = 7$$

Sedangkan dengan melakukan reduksi maka akan diperoleh

$$\Delta_1 = |87 - 86| \rightarrow \Delta_1 = 1$$

$$\Delta_2 = |81.5 - 85| \rightarrow \Delta_2 = 3.5$$

Notasi  $\Delta_n$  merupakan nilai selisih signal sisipan reduksi dan signal sisipan. Semakin kecil nilai  $\Delta_n$  maka kualitas semakin baik karena jarak dengan signal audio asli semakin mirip.

14. Denormalisasi, tahapan akhir diperlukan denormalisasi, proses ini perlu agar signal sisipan yang baru diperoleh dapat dibaca oleh *encoder* audio.
15. *Change sampling rate*, karena jumlah *sampling* signal sisipan baru yang dihasilkan hampir mendekati dua kali dari jumlah *sampling* audio awal sebesar  $2 \times (n - 1)$  jumlah sample maka *sampling rate* baru yang digunakan sebesar 2 kali dari *sampling rate* awal. *Sampling rate* awal sebesar 44100 Hz menjadi 88200 Hz.

16. Stego-audio, tahap terakhir berkas stego telah siap digunakan dengan durasi waktu yang relatif sama karena *sampling rate* telah di tingkatkan.

### 3.1.3 Ekstraksi Pesan dan *Recovery* Berkas Audio

Pada subbab ini menjelaskan tentang rancangan metode ekstraksi pesan yang diusulkan dan proses *recovery* berkas audio. Input dari proses ini adalah stego-audio. Keluaran dari proses *decoding* ada dua jenis berkas yaitu berkas audio dan berkas *data stream payload* yang berbentuk *binary*, sehingga untuk berkas *binary* diolah pada *post processing payload* yang mana prosesnya menjelaskan perubahan data hingga kembali menjadi berkas *txt*. Penjelasan tahap *decoding* ini dijelaskan sesuai dengan Gambar 3.15. Tahapan-tahapan ekstraksi memiliki beberapa proses antarlain :

1. Input stego-audio, tahap awal yaitu *load* dan pembacaan berkas audio-stego, pada skema *decoding* input yang digunakan hanya berkas stego-audio saja.
2. Normalisasi, tahap kedua normalisasi masing-masing sample audio, proses normalisasi *decoding* sama dengan proses normalisasi pada proses *encoding*. Pada contoh ini berkas signal stego-audio yang terbaca [80 86 90 81.5 80].
3. *Splitting embedded signal*, proses pemisahan dilakukan untuk mendapatkan dua signal terdapat pada signal stego-audio yaitu signal audio awal dan signal sisipan data rahasia. Tahapan ini akan digunakan pada proses selanjutnya untuk dilakukan ekstraksi data *payload*. Proses ini memperoleh 2 vektor signal yang pertama signal audio asli [80 90 80] dan vektor signal sisipan [86 81.5]. Proses pemisahan ini menggunakan pola indeks genap dan pola indeks ganjil, seperti ketentuan proses *encoding* signal berindeks ganjil merupakan signal audio asli. Sedangkan pada indeks signal genap adalah signal yang tersisipi data rahasia.
4. *Sampling* audio signal, setelah terpisah maka *sample* signal audio [80 90 80] akan digunakan pada proses selanjutnya, yaitu proses pengembalian beras audio, dan akan dilakukan NDDI kembali untuk keperluan ekstraksi data rahasia.
5. Denormalisasi, tahapan denormalisasi diperlukan agar signal audio yang dipisahkan dapat dibaca oleh *encoder* audio.
6. *Change sampling rate*, karena sample audio asli sudah dipisah maka jumlah *sample* audio pun berubah, perubahan jumlah ini menjadi setengah dari *sample*



7. File audio, setelah mengalami proses denormalisasi dan perubahan *sampling rate* maka berkas audio telah kembali menjadi berkas audio awal (*cover*).
8. NDDI, proses ini dilakukan sama seperti proses NDDI pada proses *encoding*, proses ini menggunakan persamaan (22). Keluaran signal NDDI yang dihasilkan selanjutnya akan digunakan pada proses *resmoothering* dan ekstraksi pesan. Signal NDDI ini menjadi *pivoting point* sebagai penanda signal sisipan. Hasil dari pemisahan diperoleh [80 90 80], selanjutnya akan diinterpolasi dengan NDDI sehingga diperoleh [80 **85** 90 **85** 80].
9. *Resmoothing signal*, karena signal stego-audio mengalami reduksi signal agar hasil kualitasnya lebih baik, maka pada proses ekstraksi dibutuhkan proses pengembalian signal yang direduksi menjadi signal sisipan. *Proses resmoothing* ini menggunakan persamaan (40). Penurunan persamaan (40) diperoleh dari persamaan (35) hingga persamaan (38).

$$\alpha_{j,n} = (2^j * \gamma_{j,n}) - ((2^j - 1) * \beta_n) \quad (40)$$

Notasi yang digunakan pada persamaan (40) sama dengan persamaan (39), nilai indeks yang digunakan pada proses *encoding* harus sama dengan indeks *resmoothing decoding*, apabila indeks yang digunakan berbeda maka akan berdampak pada perubahan nilai data sisipan sehingga secara langsung merubah makna data rahasia tersebut. Hasil dari proses pembacaan awal [80 **86** 90 **81.5** 80]. Pada proses ini persamaan (40) diperoleh sebagai berikut.

$$\begin{aligned} \alpha_{1,1} &= (2^1 * 86) - ((2^1 - 1) * 85) \rightarrow \alpha_{1,1} = 87 \\ \alpha_{1,2} &= (2^1 * 81.5) - ((2^1 - 1) * 85) \rightarrow \alpha_{1,2} = 78 \end{aligned}$$

Hasil *resmoothing* seperti berikut [80 **87** 90 **78** 80].

10. *Sampling embedded* signal digunakan sebagai acuan bahwa suatu signal telah tersisip data dan perlu dilakukan ekstraksi. Tahapan ini diperoleh dari tahapan sebelumnya.
11. *Difference embedded* signal & NDDI signal, tahapan ini merupakan bagian penting dari ekstraksi, karena proses ini proses pemisahan antara signal *cover* dan data rahasia. Ada 2 kondisi dalam penentuan ekstraksi data rahasia,

$$b_n = \left( \frac{x_n - x_{n+1}}{x_{n-1} - x_{n+1}} y_{n-1} + \frac{x_n - x_{n-1}}{x_{n+1} - x_{n-1}} y_{n+1} \right) - \overline{y_n'}, \text{ jika } y_n < \overline{y_{n+1}} \quad (41)$$

$$b_n = \overline{y_n}' - \left( \frac{x_n - x_{n+1}}{x_{n-1} - x_{n+1}} y_{n-1} + \frac{x_n - x_{n-1}}{x_{n+1} - x_{n-1}} y_2 \right), \text{ jika } y_n > \overline{y_{n+1}} \quad (42)$$

Dengan menggunakan persamaan (41) dan (42) diperoleh,

$$b_1 = 87 - 85 \rightarrow b_1 = 2$$

$$b_2 = 85 - 78 \rightarrow b_2 = 7$$

Data rahasia tersebut telah mapu dikembalikan mejadi 2 dan 7.

12. *Compute allocation number of bit*, proses ini adalah untuk mengetahui pengalokasian jumlah bit yang telah disisipkan sebelumnya pada tahapan *encoding*. Proses pengalokasian jumlah bit ini menggunakan persamaan (24) dan (29), serta proses yang sama dengan cara pada *encoding* proses. Dari persamaan (24) dan (29) diperoleh.

$$j_1 = 2^{\lceil \log_2 |5| \rceil} \rightarrow j_1 = 4$$

$$j_2 = 2^{\lceil \log_2 |5| \rceil} \rightarrow j_2 = 4$$

Proses tersebut menghasilkan alokasi 4 bit pada sisipan signal pertama dan kedua.

13. *Convert decimal to binary*, melakukan konversi nilai desimal menjadi *binary* menggunakan formula (43) sesuai dengan jumlah bit yang telah ditentukan pada tahap sebelumnya.

$$b_n' = \text{dec2bin}(b_n) \quad (43)$$

Sehingga diperoleh

$$b_1' = \text{dec2bin}(2) \rightarrow b_1' = (0010)_2$$

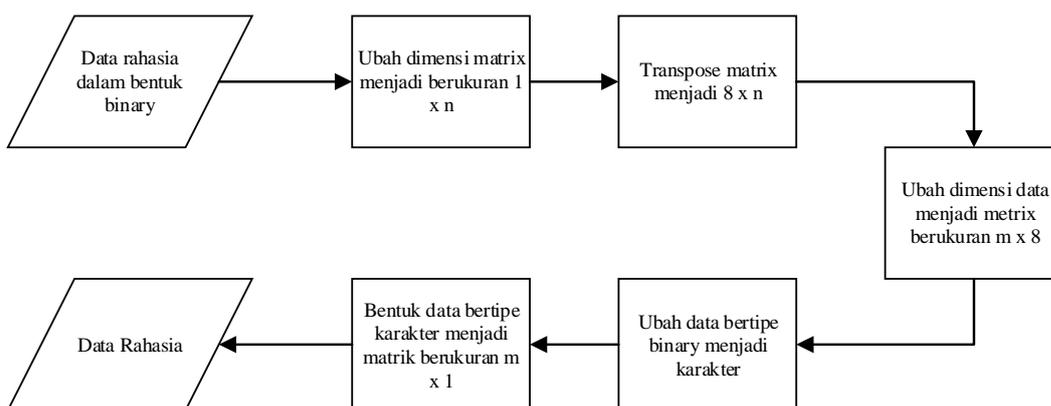
$$b_2' = \text{dec2bin}(7) \rightarrow b_2' = (0110)_2$$

14. *Temporary number of bits*, masing bit yang telah menjadi *binary* dikumpulkan menjadi bagian dalam sebuah *temporary*, bagian ini bertujuan agar bit data tersusun sesuai dengan indeks sisipan.
15. *Binary secret message*, bagian akhir data rahasia menjadi bit *stream* untuk diolah kembali menjadi teks.

### 3.1.4 Post Processing Payload

Pada subbab ini menjelaskan tentang rancangan metode ekstrasi pesan yang diusulkan dan proses *recovery* berkas audio. Skema ini merupakan kelanjutan dari

skema ekstraksi pesan dan skema proses *recovery*. Pada tahapan awal dijelaskan mengenai skema pengembalian data ekstraksi *payload* yang digunakan. Sekama ini menjelaskan perubahan data *payload* dari *binary* menjadi teks. Adapun prosesnya seperti Gambar 3.16.



Gambar 3.16 Bagan Pengembalian Data *Payload*

Pada tahap pertama menjelaskan tentang data rahasia yang digunakan, data rahasia atau *payload* yang digunakan tidak hanya melalui proses perubahan dari karakter menjadi biner, tetapi ada beberapa tahapan yang dilalui. Tahapan-tahapan ini bertujuan agar meningkatkan keamanan dari data tersebut agar tidak mudah diketahui oleh pihak yang tidak berkepentingan. Tahapan ini digambarkan sesuai dengan Gambar 3.16. Adapun tahapannya antara lain:

1. Data rahasia atau *payload* berbentuk berkas *binary*. Berisi sekumpulan bit-bit data rahasia. Langkah pertama ini melakukan *load* data dengan membuka berkas tersebut. Pada kasus ini dimisalkan data rahasia yang digunakan berupa kata “informatika” dan memiliki biner “0110100101101110011001100110111101110010011011010110000101110100011010010110101101100001”.
2. Tahapan kedua melakukan perubah ukuran matriks menjadi  $1 \times n$ . Sehingga dari contoh akan diperoleh matriks berukuran  $1 \times 88$ , sebelumnya melakuakn perhitungan jumlah biner yang digunakan sebagai data rahasia. Pada contoh karakter yang digunakan sebanyak 88 biner.

3. Tahapan ketiga yaitu melakukan *transpose* matrik dari data rahasia sebelumnya menjadi matrik  $8 \times n$ . Sehingga diperoleh matrik baru berukuran  $8 \times 11$ .
4. Selanjutnya merubah matrik menjadi menjadi berukuran  $m \times 8$ , angka 8 diperoleh dari binary 8 bit ASCII, sehingga diperoleh matrik baru sebesar  $11 \times 8$ .
5. Tahapan lima merubah seluruh data biner data rahasia menjadi karakter.
6. Tahapan keenam, merubah ukuran matrik menjadi berukuran  $m \times 1$ . Sehingga diperoleh matriks baru menjadi  $11 \times 1$ .
7. Data rahasia dari biner telah kembali menjadi karakter semula seperti saat proses penyisipan.

### 3.4 Implementasi Algoritma

Tahapan ini digunakan untuk mengimplementasikan metode yang usulkan kedalam perangkat lunak. Proses implementasi ini meliputi menulis kode program, *compilling*, *debugging*, sampai dihasilkan kode program yang siap dieksekusi.

### 3.5 Perancangan Uji Coba Metode

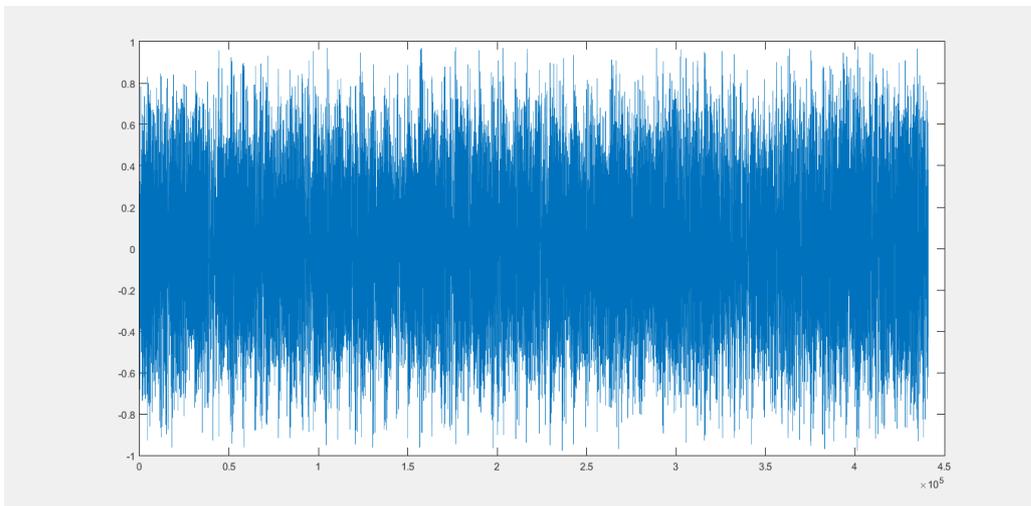
Untuk mendapatkan metode yang optimal, maka diperlukan serangkaian skenario pengujian pada metode yang diusulkan. Skenario pengujian dilakukan dengan mencocokkan data hasil ekstraksi dengan data asli. Parameter yang digunakan dalam proses skenario pengujian ini adalah:

#### 1. Ukuran pesan

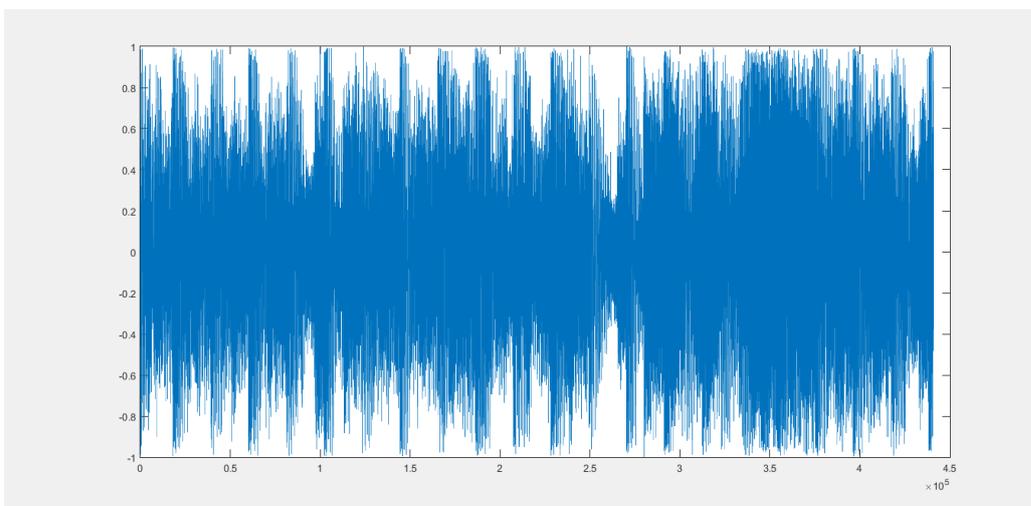
Pada skenario pertama dilakukan pengujian terhadap ukuran pesan *payload* yang mampu ditampung oleh berkas *cover* audio yang digunakan. Pada skenario pengujian ini bertujuan untuk melihat kapasitas yang paling optimal dari metode yang diusulkan. Proses skenario uji coba dilakukan dengan menyisipkan pesan dengan ukuran tertentu. Skenario penyisipan kapasitas pesan *payload* ini akan dilakukan sebanyak 12 kali. Pada setiap skenario penyisipan kapasitas pesan *payload* ini akan menggunakan pesan teks yang berukuran 5kB, sehingga pengujian ini akan dilakukan penyisipan data teks sebesar 1 kB, 5 kB, 10 kB, 15 kB, 20 kB, 25 kB, 30 kB, 35 kB, 40 kB, 45 kB, 50 kB dan 55 kB.

## 2. Genre berkas audio

Skenario pengujian kedua yaitu melihat pengaruh *genre* dari berkas audio terhadap kualitas dari hasil sisipan dan kuantitas berkas data *payload* yang mampu ditampung pada berkas audio. Media pembawa berkas audio yang digunakan pada skenario ini diujicoba dengan beberapa *genre* antara lain: *country-folk*, klasik, musik pop-rock. Skenario pengujian pada *genre* berkas audio dikarenakan pada jenis musik yang berbeda-beda memiliki nilai kuantitasi yang berbeda pada signal yang dihasilkan meskipun durasi yang digunakan sama panjangnya. Gambar 3.17 dan Gambar 3.18 merupakan contoh berkas audio yang digunakan dengan *genre* yang berbeda satu dengan yang lainnya.



Gambar 3.17. Signal Berkas Audio *Genre* Pop-Rock



Gambar 3.18. Signal berkas audio yang *genre* klasik

### 3. Instrumen berkas audio

Skenario pengujian ketiga yaitu melihat pengaruh instrumen dari berkas audio terhadap kualitas dari hasil sisipan dan kuantitas berkas data *payload* yang mampu ditampung pada berkas audio. Media pembawa berkas audio yang digunakan pada skenario ini diujicoba dengan beberapa 5 instrumen antara lain: cello, gitar akustik, piano, saxophone, dan *human voice*. Skenario pengujian pada instrumen berkas audio dikarenakan pada jenis musik yang berbeda-beda memiliki nilai kuantitasi yang berbeda pada signal yang dihasilkan meskipun durasi yang digunakan sama panjangnya.

### 4. Kapitas maksimum

Skenario pengujian keempat yaitu pengujian kapasitas maksimum yang digunakan pada skema penyisipan. Pengujian ini hanya mengukur kapasitas maksimal yang dapat disisipkan, selanjutnya dilihat dampak pada kualitas stego-audio.

### 5. *Smoothing*

Skenario pengujian dengan melakukan pengujian terhadap signal audio yang dihasilkan antara signal *embedded* dan signal *embedded* yang telah direduksi. Pada pengujian ini akan dilihat dari kinerja fungsi yang kembangkan, apakah sudah sesuai dengan hipotesis penelitian yaitu dapat memperbaiki kualitas stego-audio.

## 3.6 Evaluasi

*Signal Noise to ratio* adalah metode yang dapat digunakan untuk mengukur kualitas stego audio. Metode ini berkerja dengan cara mengukur kemiripan antara berkas audio yang telah disisipkan data rahasia dan berkas audio yang belum sisipkan data rahasia. Semakin tinggi nilai dari *Signal Noise to ratio* maka semakin baik hasil dari penyisipan data tersebut, yang mana kedua berkas tersebut menunjukkan kemiripan data yang tinggi. Satuan dari *Signal Noise to ratio* adalah *decibel* (Db). *Signal Noise to ratio* memiliki batas ambang bawah yang dianggap relevan, jika hasil berkas audio penyisipan berada dibawah batas ambang maka dianggap metode yang digunakan kurang baik karena rendahnya kemiripan anara kedua berkas. Nilai batas ambang bawah *Signal noise to ratio* yang digunakan sebesar 30 Db. Ada satu cara untuk menghitung *signal noise to ratio* pertama yaitu

dengan masing-masing data sebelum dan sesudah sisipan akan dihitung kemiripannya, untuk formula *signal noise to ratio* yang digunakan seperti pada persamaan 44. Bentuk persamaannya seperti berikut.

$$SNR(s_o, s_w) = 10 \log_{10} \frac{\sum_n [s_o(n)]^2}{\sum_n [s_w(n) - s_o(n)]^2} \quad (44)$$

Notasi  $SNR(s_o, s_w)$  merupakan *signal noise to ratio* dengan  $s_o$  adalah signal original dari *cover* berkas audio,  $s_w$  adalah signal dari hasil sisipan berkas audio.  $\sum_n [s_o(n)]^2$  merupakan total nilai kuantisasi dari seluruh data signal original sebanyak  $n$  data yang dikuadratkan.  $\sum_n [s_w(n) - s_o(n)]^2$  merupakan jumlah selisih dari masing-masing data berkas audio yang telah disisipkan dan berkas audio original yang dikuadratkan.

$$kapasitas = \sum_n k(n) \quad (45)$$

Notasi  $k(n)$  merupakan fungsi parameter dari penentuan besarnya kapsitas, masing-masing metode memiliki fungsi persamaan matematika yang berbeda-beda. Pada metode usulan ini fungsi matematika  $k(n)$  sama dengan persamaan (29), sedangkan metode pembandingan menggunakan persamaan (25).

Pada tahap akhir akan dilakukan analisis hasil terhadap semua parameter dari skenario pengujian ini, selanjutnya keluaran dari analisis ini akan dievaluasi untuk mendapatkan simpulan akhir yang menyatakan apakah metode ini lebih baik dari metode yang lain atau kurang baik dari metode yang lain.

### 3.7 Penyusunan Laporan Akhir

Pada tahap ini dilakukan pendokumentasian dan laporan dari seluruh konsep penelitian, dasar teori yang digunakan penelitian, implementasi dan penerapan hasil penelitian, serta proses yang telah dilakukan dan hasil-hasil yang telah diperoleh selama pengerjaan penelitian. Laporan akhir yang akan disusun bertujuan untuk memberikan gambaran dari pengerjaan penelitian, diharapkan hasil laporan akhir dapat berguna sebagai literatur bagi pembaca yang tertarik melakukan pengembangan lebih mendalam mengenai topik penelitian ini.

### 3.7 Jadwal Kegiatan Penelitian

Pada bagian ini menunjukkan penjadwalan penelitian tesis yang direncanakan hingga penyelesaian akhir. Tabel 3.1 menunjukkan jadwal kegiatan penelitian yang ditargetkan selama empat bulan. Penelitian dimulai pertengahan bulan Maret 2017 dan diakhiri pada pertengahan Juni 2017.

Tabel 3.2 Jadwal Kegiatan Penelitian

No.	Kegiatan	Bulan ke-1				Bulan ke-2				Bulan ke-3				Bulan ke-4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Studi literatur	■	■	■	■												
2.	Analisis dan desain			■	■	■	■	■	■								
3.	Implementasi Perangkat Lunak									■	■	■	■	■			
4.	Uji coba dan analisis														■	■	■
5.	Penulisan naskah		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

*[Halaman ini sengaja dikosongkan]*

## BAB 4

### HASIL DAN PEMBAHASAN

Bab 4 berisi pembahasan mengenai hasil penelitian dan pembahasan penelitian. Pada bab ini berisi lingkungan uji coba, pelaksanaan uji coba yang berisi *dataset* yang digunakan, hasil serta evaluasi kinerja dari metode usulan penelitian.

#### 5.1 Lingkungan Uji Coba

Lingkungan perancangan pada proses implementasi algoritma ini meliputi perangkat keras dan perangkat lunak. Spesifikasi perangkat keras dan perangkat lunak yang digunakan pada penelitian ini seperti ditunjukkan pada Tabel 4.1 berikut:

Tabel 4.1 Lingkungan Implementasi Penelitian

Lingkungan	Spesifikasi	
Perangkat Keras	<i>Processor</i>	AMD A8, 2 GHz
	<i>RAM</i>	8192 Mb
	<i>VGA</i>	4574 Mb
Perangkat Lunak	Sistem Operasi	Windows
	<i>Tools</i>	Matlab 2015

#### 4.2 Pelaksanaan Uji Coba

Penelitian ini menggunakan *input* berkas audio dengan jenis *wave audio format* dan berekstensi *.wav*. Pemilihan berkas audio *wav* dikarenakan berkas ini tidak mengalami proses kompresi sehingga diharapkan tidak ada berkas yang hilang baik sebelum maupun sesudah dilakukan penyisipan berkas.

*Dataset* yang digunakan pada penelitian ini yaitu IRMAS (*Dataset For Instrument Recognition In Musical Audio Signals*) dan berkas audio dapat diunduh pada alamat url [www.mtg.upf.edu/download/dataset](http://www.mtg.upf.edu/download/dataset).

Berkas audio yang digunakan sebanyak 15 data, terdiri dari 5 instrumen dan masing-masing instrumen terdiri dari 3 *genre* seperti ditampilkan pada Tabel 4.2. Durasi berkas audio yang digunakan pada masing-masing jenis musik selama 3 detik. Karakteristik audio seperti pada Tabel 4.3. Sedangkan Berkas yang digunakan untuk data *payload* merupakan berkas teks dengan berekstensi *.txt*. Data

ini diperoleh dari teks generator, [www.id.lipsum.com](http://www.id.lipsum.com) merupakan penyedia dari teks *generator*.

Data *payload* yang digunakan bervariasi sebanyak 12 jenis. Masing-masing jenis berbeda ukurannya dari 1kB, 5kB, 10kB, 15kB, 20kB, 25kB, 30kB, 35kB, 40kB, 45kB, 50kB dan 55kB. Berkas yang digunakan untuk data *payload* merupakan berkas teks dengan berekstensi .txt.

Tabel 4.2 Dataset Berkas Audio

<b>Instrumen</b>	<b>Genre</b>	<b>Data</b>	<b>Nama</b>
Cello (cel)	Country-Folk (cou_fol)	[cel][jaz_blu]0011__2	Data 1
	Classical (cla)	[cel][cla]0007_1	Data 2
	Pop-Rock (pop-roc)	[cel][pop_roc]0060_2	Data 3
Acoustic guitar (gac)	Country-Folk (cou_fol)	[gac][jaz_blu]0549_2	Data 4
	Classical (cla)	[gac][cla]0530_1	Data 5
	Pop-Rock (pop-roc)	[gac][pop_roc]0560_3	Data 6
Piano (pia)	Country-Folk (cou_fol)	[pia][jaz_blu]1348_1	Data 7
	Classical (cla)	[pia][cla]1291_1	Data 8
	Pop-Rock (pop-roc)	[pia][pop_roc]1306_1	Data 9
Saxophone (sax)	Country-Folk (cou_fol)	[sax][jaz_blu]1605_2	Data 10
	Classical (cla)	[sax][cla]1598_1	Data 11
	Pop-Rock (pop-roc)	[sax][pop_roc]1588_1	Data 12
Human singing voice (voi)	Country-Folk (cou_fol)	[voi][jaz_blu]2358_1	Data 13
	Classical (cla)	[voi][nod][cou_fol]2442_1	Data 14
	Pop-Rock (pop-roc)	[pop_roc]2547_3	Data 15

Tabel 4.3 Karakteristik Audio

<b>No</b>	<b>Keterangan</b>	<b>Karakteristik</b>
1	Data Pembawa	Data 1 ~ Data 15
2	Jumlah Kanal	1 Buah
3	<i>Sample Rate</i>	44100
4	Jumlah <i>Sampling</i>	132299
5	Durasi Waktu	~ 3 Detik
6	Bit Per Sample	16 bit

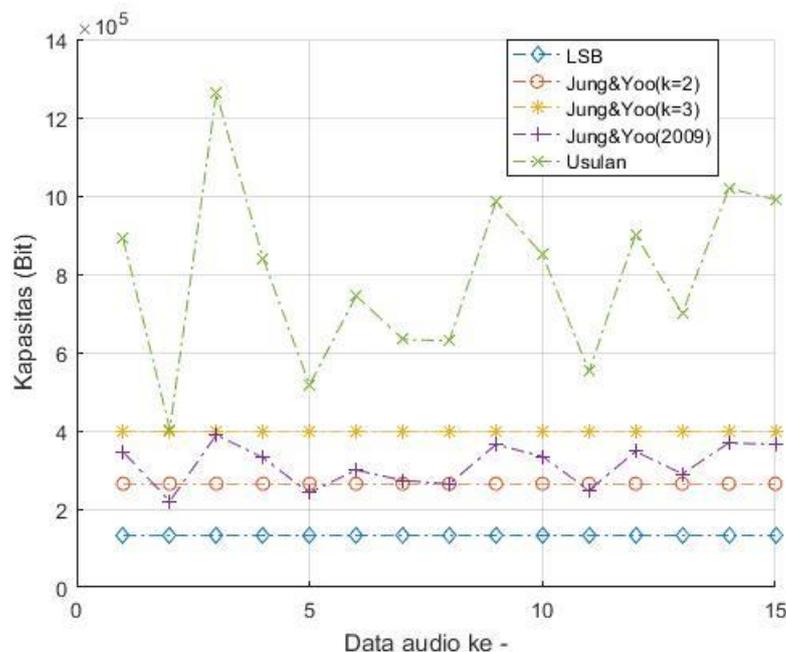
### 4.3 Hasil dan Evaluasi

Pada bab Pembahasan penelitian disajikan dua sub bab. Pertama pembahasan mengenai kapasitas yang dihasilkan dari metode usulan dan metode pembandingan, kedua pembahasan mengenai kualitas yang diperoleh dari stego-audio.

#### 4.3.1 Skenario 1

Pada pengujian ini dilakukan evaluasi dengan menggunakan persamaan 45. Pengujian kapasitas ini bertujuan untuk mengetahui seberapa banyak dari masing-masing audio mampu menampung bits data rahasia. Selain untuk mengetahui kapasitas yang dapat disisipkan pengujian ini juga bertujuan untuk melihat kinerja metode yang diusulkan dibandingkan dengan metode sebelumnya. Metode yang digunakan sebagai pembandingan antara lain *least significant bit* (LSB), usulan metode Jung dan Yoo pada tahun 2009, usulan metode Jung dan Yoo pada tahun 2015 dengan menggunakan nilai parameter ( $k=2$ ) dan ( $k=3$ ).

Secara umum metode yang diusulkan memiliki kemampuan menyisipkan jauh lebih banyak dibandingkan dengan metode lain. Hasil dari pengujian kapasitas ini disajikan pada Gambar 4.1.



Gambar 4.1 Hasil Perbandingan Kapasitas

Pada metode pembandingan LSB kapasitas maksimum yang dapat ditampung sebesar 132299 bit. Hasil penyisipan metode memiliki nilai yang konstan pada audio 1 hingga audio 15. Pada metode ini instrumen dan *genre* lagu tidak berpengaruh pada penyisipan, hal ini disebabkan karena kinerja dari LSB menyisipkan bit-bit rahasia pada masing-masing *sample* hanya sebanyak 1 bit. Metode ini hanya berpengaruh pada jumlah *sample* dari signal audio, semakin banyak *sample* dari berkas audio maka semakin banyak data *payload* yang dapat ditampung.

Tabel 4.4 Kapasitas Maksimum Metode Usulan dan Pembandingan

Audio	Kapasitas (Bits)				Usulan	Keterangan
	LSB	Jung & yoo (k = 2) (2015)	Jung & yoo (k = 3) (2015)	Jung & yoo (2009)		
1	132299	264596	396894	345238	<b>889954</b>	Meningkat
2	132299	264596	396894	219731	<b>403850</b>	Meningkat
3	132299	264596	396894	389733	<b>1262296</b>	Meningkat
4	132299	264596	396894	330847	<b>840109</b>	Meningkat
5	132299	264596	396894	240945	<b>516739</b>	Meningkat
6	132299	264596	396894	299593	<b>744691</b>	Meningkat
7	132299	264596	396894	272783	<b>633483</b>	Meningkat
8	132299	264596	396894	263823	<b>629116</b>	Meningkat
9	132299	264596	396894	364703	<b>985074</b>	Meningkat
10	132299	264596	396894	334146	<b>849049</b>	Meningkat
11	132299	264596	396894	247738	<b>551105</b>	Meningkat
12	132299	264596	396894	347158	<b>901643</b>	Meningkat
13	132299	264596	396894	289733	<b>698944</b>	Meningkat
14	132299	264596	396894	368453	<b>1018824</b>	Meningkat
15	132299	264596	396894	365114	<b>990378</b>	Meningkat

Pada metode pembandingan Jung & Yoo (2015) kapasitas maksimum yang dapat ditampung sebesar 264596 bits dengan menggunakan parameter ( $k = 2$ ), dan kapasitas maksimum yang dapat ditampung dengan parameter ( $k = 3$ ) sebesar 396894 bits. Hasil penyisipan metode memiliki nilai yang konstan pada audio 1 hingga audio 15. Pada metode ini instrumen dan *genre* lagu tidak berpengaruh pada penyisipan, hal ini disebabkan karena kinerja dari usulan Jung & Yoo pada tahun 2015 mengadopsi dan memodifikasi metode LSB. Metode usulan Jung dan Yoo berkerja dengan menyisipkan bit-bit rahasia pada masing-masing *sample* hanya sebanyak  $k$  bit, dimana nilai  $k$  tersebut telah ditentukan pada penelitian. Jika nilai parameter ( $k = 2$ ) maka banyak bit yang disisipkan pada *sample* audio sebanyak dua bit, begitu pula pada nilai parameter ( $k = 3$ ), setelah ditentukan jumlah bit dan nilai bit yang disisipkan maka selanjutnya bit tersebut diubah menjadi nilai integer. Penyisipan yang dilakukan pada domain tipe data integer. Metode ini juga hanya berpengaruh pada jumlah *sample* dari signal audio, semakin banyak *sample* dari berkas audio maka semakin banyak data *payload* yang dapat ditampung.

Pembahasan selanjutnya membahas 2 metode yaitu metode usulan penelitian, dan metode Jung yoo pada tahun 2009. Metode usulan dan metode Jung & Yoo memiliki kesamaan, kesamaan itu terletak pada penyisipan yang dilakukan memiliki kemampuan yang adaptif, dimana jumlah kapasitas yang disisipkan tidak konstan. Metode usulan sangat dipengaruhi oleh jumlah *sample* yang dimiliki berkas audio, semakin banyak *sample* dari berkas audio maka semakin banyak bit *payload* yang dapat disisipkan. Instrumen dan *genre* musik berkas audio sangat berpengaruh dan berdampak pada banyaknya penyisipan, hal ini dikarenakan proses penentuan banyaknya jumlah bit yang akan disisipkan dipengaruhi oleh besarnya *magnitude* dari *sample* audio, semakin besar jarak atau selisih antara dua *sample* maka semakin banyak data *payload* yang dapat disisipkan. Pada *genre* audio seperti *pop-rock* memiliki perubahan *magnitude* yang besar antara signal sehingga *genre* ini memiliki kemampuan penyisipan yang sangat banyak. Berbeda dengan *genre country-folk* yang mana jarak atau selisih antar signal relatif kecil sehingga penyisipann yang dilakukan sangat terbatas.

Pada audio 1 (instrumen cello dan *genre country folk*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 34523 bits dan metode usulan sebanyak 889954 bits. Peningkatan kapasitas sebesar 544716 bits atau sebesar 157%.

Pada audio 2 (instrumen cello dan *genre klasik*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 219731 bits dan metode usulan sebanyak 403850 bits. Peningkatan kapasitas sebesar 184119 bits atau sebesar 83%.

Pada audio 3 (instrumen cello dan *genre pop rock*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 389733 bits dan metode usulan sebanyak 1262296 bits. Peningkatan kapasitas sebesar 872563 bits atau sebesar 223%.

Pada audio 4 (instrumen akustik gitar dan *genre country folk*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 330847 bits dan metode usulan sebanyak 840109 bits. Peningkatan kapasitas sebesar 509262 bits atau sebesar 153%.

Pada audio 5 (instrumen akustik gitar dan *genre klasik*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 240945 bits dan metode usulan sebanyak 516739 bits. Peningkatan kapasitas sebesar 275794 bits atau sebesar 114%.

Pada audio 6 (instrumen akustik gitar dan *genre pop rock*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 299593 bits dan metode usulan sebanyak 744691 bits. Peningkatan kapasitas sebesar 445098 bits atau sebesar 148%.

Pada audio 7 (instrumen piano dan *genre country folk*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 272783 bits dan metode usulan sebanyak 633483 bits. Peningkatan kapasitas sebesar 360700 bits atau sebesar 132%.

Pada audio 8 (instrumen piano dan *genre klasik*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 263823 bits dan metode usulan sebanyak 629116 bits. Peningkatan kapasitas sebesar 365293 bits atau sebesar 138%.

Pada audio 9 (instrumen piano dan *genre pop rock*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 364703 bits dan metode usulan sebanyak 985074 bits. Peningkatan kapasitas sebesar 620371 bits atau sebesar 170%.

Pada audio 10 (instrumen saxophone dan *genre country folk*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 334146 bits dan metode usulan

sebanyak 849049 bits. Peningkatan kapasitas sebesar 514903 bits atau sebesar 154%.

Tabel 4.5 Selisih Kapasitas Antara Metode Usulan Terhadap Metode Pembanding

Audio	Kapasitas (Bits)			
	LSB	Jung & yoo (k = 2)	Jung & yoo ( k = 3)	Jung & yoo
		(2015)	(2015)	(2009)
1	757655	625358	<b>493060</b>	544716
2	271551	139254	<b>6956</b>	184119
3	1129997	997700	<b>865402</b>	872563
4	707810	575513	<b>443215</b>	509262
5	384440	252143	<b>119845</b>	275794
6	612392	480095	<b>347797</b>	445098
7	501184	368887	<b>236589</b>	360700
8	496817	364520	<b>232222</b>	365293
9	852775	720478	<b>588180</b>	620371
10	716750	584453	<b>452155</b>	514903
11	418806	286509	<b>154211</b>	303367
12	769344	637047	<b>504749</b>	554485
13	566645	434348	<b>302050</b>	409211
14	886525	754228	<b>621930</b>	650371
15	858079	725782	<b>593484</b>	625264

Pada audio 11 (instrumen saxophone dan *genre* klasik) metode Jung & Yoo mampu melakukan penyisipan sebanyak 247738 bits dan metode usulan sebanyak 551105 bits. Peningkatan kapasitas sebesar 303367 bits atau sebesar 122%.

Pada audio 12 (instrumen saxophone dan *genre pop rock*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 347158 bits dan metode usulan sebanyak 901643 bits. Peningkatan kapasitas sebesar 554485 bits atau sebesar 159%.

Pada audio 13 (instrumen *human voice* dan *genre country folk*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 289733 bits dan metode usulan sebanyak 698944 bits. Peningkatan kapasitas sebesar 409211 bits atau sebesar 141%.

Pada audio 14 (instrumen *human voice* dan *genre* klasik) metode Jung & Yoo mampu melakukan penyisipan sebanyak 368453 bits dan metode usulan sebanyak 1018824 bits. Peningkatan kapasitas sebesar 650371 bits atau sebesar 176%.

Pada audio 15 (instrumen *human voice* dan *genre pop rock*) metode Jung & Yoo mampu melakukan penyisipan sebanyak 365144 bits dan metode usulan sebanyak 990378 bits. Peningkatan kapasitas sebesar 625264 bits atau sebesar 171%.

Hasil tersebut menunjukkan bahwa kapasitas penyisipan yang paling sedikit pada audio 5 (instrumen akustik gitar dan *genre* klasik), sedangkan kapasitas penyisipan yang paling banyak pada audio 3 (instrumen cello dan *genre pop rock*). Dari hasil tersebut dapat disimpulkan bahwa metode usulan penelitian ini telah mampu meningkatkan kapasitas penyisipan dibandingkan dengan metode lainnya.

Tabel 4.5 menunjukkan hasil selisih kemampuan kapasitas antara metode usulan terhadap metode pembanding, semakin kecil nilai kapasitas yang diperoleh maka metode pembanding tersebut mendekati dengan metode usulan, hal ini menunjukkan bahwa metode tersebut menjadi metode alternatif untuk proses penyisipan data rahasia pada media audio.

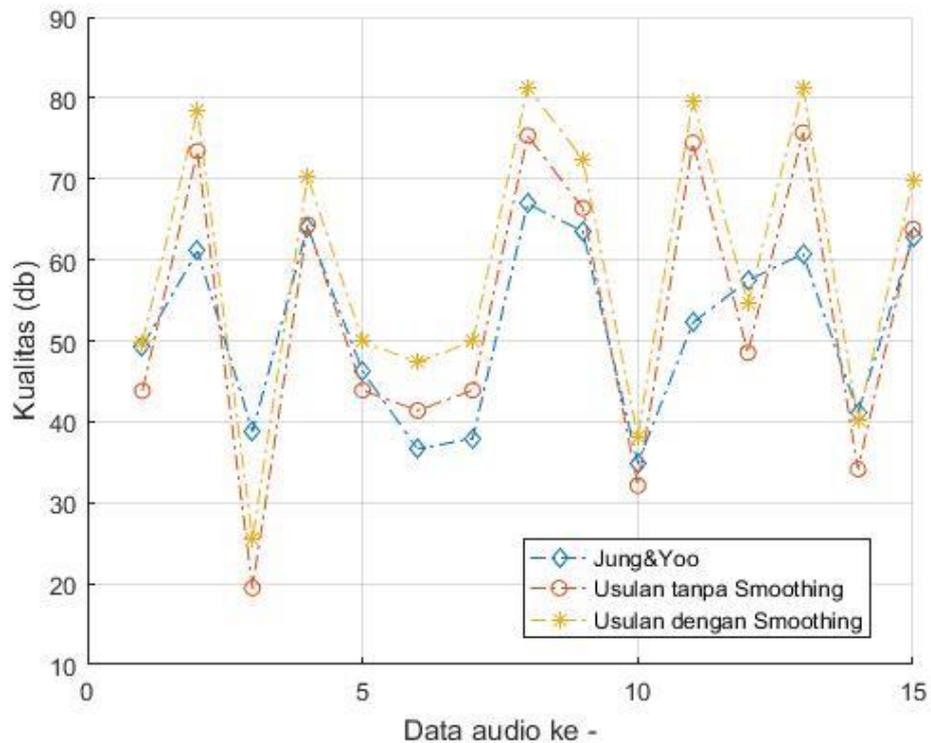
#### **4.3.2 Skenario 2**

Pada pengujian ini dilakukan evaluasi dengan menggunakan persamaan 44. Pengujian kapasitas ini bertujuan untuk mengetahui kualitas hasil sisipan stego-audio. Pengujian ini juga bertujuan mengetahui pengaruh data *payload* yang disisip serta *genre* yang digunakan. Pengujian ini dibandingkan dengan metode lain, metode yang digunakan sebagai pembanding yaitu metode Jung & Yoo.

Gambar 4.2 menyajikan hasil kualitas yang diperoleh dari stego-audio. Pengujian ini juga menyajikan hasil evaluasi dari metode usulan dengan proses

reduksi signal, sehingga dapat dibandingkan antara signal sisipan sebelum mendapatkan proses *smoothing* dan sesudah mendapatkan proses *smoothing*.

Pengujian ini melakukan evaluasi sebanyak 96 kali, jumlah pengujian ini diperoleh dari jumlah data audio yang digunakan dikalikan dengan jumlah data *payload*.



Gambar 4.2 Hasil Perbandingan Kualitas

Hasil yang diperoleh Gambar 4.2 merupakan hasil rata-rata kualitas yang dilakukan dengan menggunakan 12 jenis data *payload* (1kB – 55kB).

Hasil pada audio 1, kualitas metode Jung & Yoo sebesar 49.3 dB, usulan tanpa *smoothing* sebesar 43.8 dB dan metode usulan dengan proses *smoothing* sebesar 49.8 dB. Metode usulan meningkat sebesar 1.09% dari metode pembandingan.

Hasil pada audio 2, kualitas metode Jung & Yoo sebesar 61.1 dB, usulan tanpa *smoothing* sebesar 73.4 dB dan metode usulan dengan proses *smoothing* sebesar 78.4.8 dB. Metode usulan meningkat sebesar 28.3% dari metode pembandingan.

Hasil pada audio 3, kualitas metode Jung & Yoo sebesar 38.8 dB, usulan tanpa *smoothing* sebesar 19.4 dB dan metode usulan dengan proses *smoothing* sebesar 25.5 dB. Metode usulan menurun sebesar -34.3% dari metode pembandingan.

Hasil pada audio 4, kualitas metode Jung & Yoo sebesar 64.1 dB, usulan tanpa *smoothing* sebesar 64.3 dB dan metode usulan dengan proses *smoothing* sebesar 70.3 dB. Metode usulan meningkat sebesar 9.79% dari metode pembandingan.

Tabel 4.6 Hasil Rerata Kualitas Stego-audio

Audio	Rerata (DR 1 - 12) PSNR (dB)			Keterangan
	Jung & Hoo	Usulan Tanpa Smooting	Usulan dengan Smooting	
1	49.29438	43.81411	<b>49.83451</b>	Lebih baik
2	61.1182	73.44021	<b>78.43474</b>	Lebih baik
3	<b>38.79525</b>	19.46299	25.48287	Penurunan kualitas
4	64.07636	64.32995	<b>70.35011</b>	Lebih baik
5	46.24282	43.89501	<b>49.91348</b>	Lebih baik
6	36.62736	41.37906	<b>47.36915</b>	Lebih baik
7	37.96585	43.90299	<b>49.88902</b>	Lebih baik
8	66.95735	75.32739	<b>81.34393</b>	Lebih baik
9	63.49767	66.39761	<b>72.41816</b>	Lebih baik
10	34.87374	32.12237	<b>38.14476</b>	Lebih baik
11	52.23167	74.47474	<b>79.7325</b>	Lebih baik
12	<b>57.43312</b>	48.53977	54.55991	Penurunan kualitas
13	60.75326	75.72546	<b>81.29412</b>	Lebih baik
14	<b>41.1379</b>	34.13539	40.15465	Penurunan kualitas
15	62.7137	63.80433	<b>69.82415</b>	Lebih baik

Hasil pada audio 5, kualitas metode Jung & Yoo sebesar 46.2 dB, usulan tanpa *smoothing* sebesar 43.9 dB dan metode usulan dengan proses *smoothing* sebesar 49.9 dB. Metode usulan meningkat sebesar 7.93% dari metode pembandingan.

Hasil pada audio 6, kualitas metode Jung & Yoo sebesar 36.6 dB, usulan tanpa *smoothing* sebesar 41.3 dB dan metode usulan dengan proses *smoothing* sebesar 47.3 dB. Metode usulan meningkat sebesar 29.3% dari metode pembandingan.

Hasil pada audio 7, kualitas metode Jung & Yoo sebesar 37.9 dB, usulan tanpa *smoothing* sebesar 43.9 dB dan metode usulan dengan proses *smoothing* sebesar 49.8 dB. Metode usulan meningkat sebesar 31.4% dari metode pembandingan.

Hasil pada audio 8, kualitas metode Jung & Yoo sebesar 66.9 dB, usulan tanpa *smoothing* sebesar 75.3 dB dan metode usulan dengan proses *smoothing* sebesar 81.34 dB. Metode usulan meningkat sebesar 21.5% dari metode pembandingan.

Hasil pada audio 9, kualitas metode Jung & Yoo sebesar 63.5 dB, usulan tanpa *smoothing* sebesar 66.4 dB dan metode usulan dengan proses *smoothing* sebesar 72.4 dB. Metode usulan meningkat sebesar 14.0% dari metode pembandingan.

Hasil pada audio 10, kualitas metode Jung & Yoo sebesar 34.8 dB, usulan tanpa *smoothing* sebesar 32.1 dB dan metode usulan dengan proses *smoothing* sebesar 38.1 dB. Metode usulan meningkat sebesar 9.37% dari metode pembandingan.

Hasil pada audio 11, kualitas metode Jung & Yoo sebesar 52.2 dB, usulan tanpa *smoothing* sebesar 74.5 dB dan metode usulan dengan proses *smoothing* sebesar 79.7 dB. Metode usulan meningkat sebesar 52.6% dari metode pembandingan.

Hasil pada audio 12, kualitas metode Jung & Yoo sebesar 57.4 dB, usulan tanpa *smoothing* sebesar 48.5 dB dan metode usulan dengan proses *smoothing* sebesar 54.4 dB. Metode usulan menurun sebesar -5.00% dari metode pembandingan.

Hasil pada audio 13, kualitas metode Jung & Yoo sebesar 60.7 dB, usulan tanpa *smoothing* sebesar 75.7 dB dan metode usulan dengan proses *smoothing* sebesar 81.3 dB. Metode usulan meningkat sebesar 33.8% dari metode pembandingan.

Hasil pada audio 14, kualitas metode Jung & Yoo sebesar 41.1 dB, usulan tanpa *smoothing* sebesar 34.1 dB dan metode usulan dengan proses *smoothing* sebesar 40.1 dB. Metode usulan menurun sebesar -2.3% dari metode pembandingan.

Hasil pada audio 15, kualitas metode Jung & Yoo sebesar 62.7 dB, usulan tanpa *smoothing* sebesar 63.8 dB dan metode usulan dengan proses *smoothing* sebesar 69.8 dB. Metode usulan meningkat sebesar 11.3% dari metode pembandingan.

Hasil pengujian kualitas menunjukkan bahwa secara umum metode usulan dengan proses *smoothing* signal telah mampu memperbaiki kualitas stego-audio pada beberapa jenis audio. Pada pengujian audio 3, audio 12, dan audio 14 metode terdapat anomali, metode yang diusulkan memiliki kualitas dibawah dari metode pembandingan. Setelah diidentifikasi terjadinya anomali kualitas dari stego-audio disebabkan oleh kapasitas penyisipan. Ruang kapasitas penyisipan pada audio 3, audio 12, dan audio 14 dengan metode pembandingan lebih sedikit dibandingkan dengan metode usulan, sehingga kualitas stego-audio metode pembandingan lebih baik.

Kesimpulan dari pengujian kualitas yaitu kualitas stego-audio akan berbanding terbalik dengan kapasitas sisipan, semakin banyak data *payload* yang sisip maka kualitas semakin buruk, berlaku sebaliknya, semakin sedikit kapasitas sisipan maka semakin baik kualitas yang dihasilkan stego-audio.

## BAB 5

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

1. Penyisipan yang dikembangkan metode usulan dengan menggunakan metode *newton's divided-difference interpolating polynomials* telah mampu menyembunyikan data *payload* tanpa mempengaruhi dan merusak *sampling* data audio asli, karena penyisipan *payload* dilakukan pada signal *newton's divided-difference interpolating polynomials* bukan pada signal audio asli.
2. Konsep *pivoting point* digunakan sebagai pengganti *location maps*, karena *pivoting point* pada *newton's divided-difference interpolating polynomials* menghasilkan ruang penyisipan yang berpola, pola inilah yang menjadi penanda bahwa signal yang telah disisipkan.
3. Kapasitas penyisipan *payload* pada berkas audio kanal tunggal dengan *newton's divided-difference interpolating polynomials* dapat ditingkatkan dengan menggunakan fungsi *nonlinear* berpangkat logaritmik, dengan fungsi ini menghasilkan jumlah kapasitas yang disisipkan bertambah secara *eksponensial*, fungsi ini menjadi salah satu parameter untuk menghasilkan jumlah kapasitas, parameter lain yang menjadi faktor penentu besar kapasitas yaitu parameter selisih dua signal, semakin besar nilai selisih antara dua signal yang bersebelahan maka semakin banyak data *payload* yang dapat disisipkan.
4. Kualitas penyisipan *payload* pada berkas audio kanal tunggal dengan *newton's divided-difference interpolating polynomials* dapat diperbaiki dengan proses *smoothing* signal, proses ini dikembangkan dengan mengadopsi model matematika dari fungsi persamaan garis. Fungsi *smoothing* dapat mereduksi jarak signal sisipan yang berdampak pada kualitas sisipan, semakin kecil jarak signal sisipan dengan jarak signal *newton's divided-difference interpolating polynomials* maka kualitas sisipan semakin baik.
5. *Genre Pop-Rock* dengan instrumen *cello* merupakan *genre* dan instrumen yang memiliki jumlah kapasitas maksimum ditunjukkan dari hasil penyisipan

pada audio 3 sebesar **1262296 bits**. *Genre Country-folk* dengan instrumen *Human singing voice* merupakan *genre* dan instrumen yang memiliki kualitas sisipan terbaik ditunjukkan hasil penyisipan pada audio 13 sebesar **81.29412 dB**.

## **5.2 Saran**

1. Pengembangan metode dengan DE, RDE, maupun GRDE.
2. Mereduksi size berkas audio.
3. Meningkatkan kapasitas dengan model interpolasi polynomial orde tinggi.
4. Melakukan penyisipan pada berkas audio berkanal banyak.

## DAFTAR PUSTAKA

- Bilal, I., Roj, M.S., Kumar, R., Mishra, P.K. (2014). Recent advancement in audio steganography. Paralel, Distributed and Grid Computing.
- Feng, J. B. Lin, L, C., Tsui, C. S., & Chu, Y., (2006). Reversible Watermarking: current Status and Key Issues. International Journal of Network Security, 161-171
- Ingemar, J. Cox. (2008). Digital watermarking and steganography, Burlington: Morgan Kaufmann.
- Li, L. Fang, X. (2010). Adaptive detection for spread spectrum audio watermarking. Wireless Communications, Networking and Information Security (WCNIS). IEEE International Conference on pp: 58-62.
- Lin, C., C., Yang., S., P., & Hsueh, N., L., (2008). Lossless Data Hiding Based on Difference Expansion without a Location Map. Congress on Image and Signal Processing. 8-12
- Natgunanathan, I. Xiang, Y. (2010). A novel bipolar time-spread echo hiding based watermarking method for stereo audio signals. 4th International Conference on Signal Processing and Communication Systems. ICSPCS'2010.
- Patel, K. Rora, K. Singh, K. Verma, S. (2013). Lazy Wavelet Transform Based Steganography in Video. International Conference on Communication Systems and Network Technologies. pp: 497-500
- Tian, Jun. (2003). Reversible data embedding using a difference expansion. Circuit and system for video Technology. Vol 13. No 8. Pp 890
- Tohari, Ahmad. Holil, M. (2014). Increasing the performance of difference expansion-based steganography for securing Medical Data.
- Zhou, M. Li, S. Luo. X, Wang, D. (2013). A New Echo Hiding Algorithm with High Robustness. International Journal of Automation and Power Engineering (IJAPE). vol: 2 (4) pp: 240-246

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Tegar palyus fiqar lahir di Samarinda pada tanggal 7 September 1990. Sejak SD hingga SMA pendidikannya ditempuh di kota Samarinda. Ia menyelesaikan pendidikan SD nya di SD Muhammadiyah 1 Samarinda pada tahun 2002, kemudian melanjutkan menempuh pendidikan SMP di SMP Negeri 1 Samarinda dan menyelesaikan pendidikan SMP pada tahun 2005, pendidikan SMA dilanjutkan di SMA Negeri 10 “Melati” Samarinda dan diselesaikan pada tahun 2008. Setelah lulus SMA, ia memilih untuk melanjutkan pendidikan sarjana di Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan jurusan Teknik Elektro dan mengambil konsentrasi bidang studi teknik komputer dan telematika. Hal ini tidak lepas dari ketertakiannya dalam bidang teknologi. Setelah lulus pada tahun 2013 membangun start-up “niltava-labs” bersama teman satu lab, kurang dari setahun mendapat peluang beasiswa untuk melanjutkan pendidikan pasca sarjana di kampus yang sama saat tingkat sarjana, prodi pasca sarjana yang dituju yaitu teknik informatika. Apabila kedepannya bagi pembaca ingin mendiskusikan baik kritik dan saran tentang isi dari penelitian ini dapat menghubungi melalui email [tegar.pf@gmail.com](mailto:tegar.pf@gmail.com)