



Tesis - KI142502

PENGEMBANGAN METODE DATA HIDING PADA AUDIO BERBASIS RANDOM REDUCE DIFFERENCE EXPANSION

Dwi Shinta Angreni

NRP: 5114201036

DOSEN PEMBIMBING

Dr. Tohari Ahmad, S.Kom, MIT.

PROGRAM MAGISTER

BIDANG KOMPUTASI BERBASIS JARINGAN

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2016



Thesis - KI142502

***AN IMPROVE OF AUDIO DATA HIDING
BASED ON RANDOM REDUCE DIFFERENCE
EXPANSION***

Dwi Shinta Angreni
NRP : 5114201036

SUPERVISOR

Dr. Tohari Ahmad, S.Kom, MIT.

MAGISTER PROGRAM

THE EXPERTISE FIELD OF NETWORK CENTRIC COMPUTING

DEPARTEMENT OF INFORMATICS

FACULTY OF INFORMATION TECHNOLOGY

INSTITUTE TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2016

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)

di

Institut Teknologi Sepuluh Nopember Surabaya

oleh:

DWI SHINTA ANGRENI

NRP. 5114201036

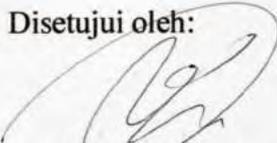
Dengan judul:

PENGEMBANGAN METODE DATA HIDING PADA AUDIO
BERBASIS RANDOM REDUCE DIFFERENCE EXPANSION

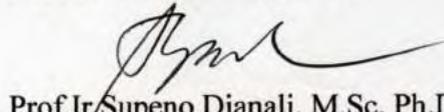
Tanggal Ujian : 27-6-2016

Periode Wisuda : 2015 Genap

Disetujui oleh:


Tohari Ahmad, S.Kom, MIT, Ph.D
NIP. 197505252003121002

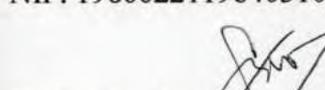
(Pembimbing 1)


Prof. Ir. Supeno Djanali, M.Sc, Ph.D
NIP. 194806191973011001

(Penguji 1)


Ir. Muchammad Husni, M.Kom
NIP. 196002211984031001

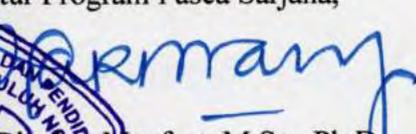
(Penguji 2)


Hudan Studiawan, S.Kom, M.Kom
NIP. 198705112012121003

(Penguji 3)

Direktur Program Pasca Sarjana,




Prof. Ir. Djuhanar Manfaat, M.Sc., Ph.D
NIP. 196012021987011001

(halaman ini sengaja dikosongkan)

PENGEMBANGAN METODE DATA HIDING PADA AUDIO BERBASIS RANDOM REDUCE DIFFERENCE EXPANSION

Nama mahasiswa : Dwi Shinta Angreni
NRP : 5114201036
Pembimbing : Dr. Tohari Ahmad, S.Kom, MIT.

ABSTRAK

Overflow dan *underflow* masih menjadi salah satu masalah yang timbul dalam proses data *hiding*. Hal ini menyebabkan tidak semua data yang ada pada *cover* media dapat disisipkan, sehingga kapasitas yang dihasilkan menjadi berkurang. Untuk dapat mengatasi masalah ini, diusulkan sebuah metode *data hiding* yang dapat menangani masalah *overflow* dan *underflow* sehingga semua nilai *cover* media dapat disisipkan. Metode ini merupakan pengembangan dari metode data *hiding* gambar yaitu DE dan RDE.

Agar dapat bekerja pada domain audio dimana adanya perbedaan antara struktur gambar dan audio, target gambar skema ini adalah dua dimensi 8-bit gambar *grayscale* dan target audio adalah satu dimensi 16-bit kuantisasi dari gelombang audio. Dengan metode *intelligent partitioning* (IP), yang membagi gelombang audio menjadi dua bagian, sehingga skema *data hiding* gambar dapat diterapkan pada audio. Untuk dapat mempertahankan kualitas audio hasil penyisipan maka diusulkan pengembangan metode partisi IP yang mendukung metode *data hiding* yang diusulkan.

Metode penyisipan yang diusulkan pada penelitian ini adalah *Random Reduce Difference Expansion* (RRDE). Berdasarkan uji coba yang dilakukan gabungan dua metode yang diusulkan yaitu RRDE dan IP usulan 2 dapat menghasilkan kapasitas terbaik dibanding metode lainnya. Sedangkan Nilai SNR terbaik dihasilkan metode RRDE tanpa menggunakan metode partisi. Dimana rata-rata nilai SNR yang dihasilkan diatas 40 dB.

Kata Kunci: *data hiding, reduce difference expansion, intelligent partitioning, audio*

(halaman ini sengaja dikosongkan)

AN IMPROVE OF AUDIO DATA HIDING BASED ON RANDOM REDUCE DIFFERENCE EXPANSION

Name : Dwi Shinta Angreni
Student Identity Number: 5114201036
Supervisor : Dr. Tohari Ahmad, S.Kom, MIT.

ABSTRACT

Overflow and underflow is one of the issues that arise in data hiding process. This causes all the existing data on the media cover can not be embedded, so that make the capacity result is reduced . In order to solve this problem, we propose a method of data hiding to overcome overflow and underflow so that we can embedd all values in the media cover . This method is the improvement of data hiding methods for images that is DE and RDE.

In order to make it work in the audio domain where there is differences between the structure of image and audio. We need to change one-dimensional 16-bit audio to 8-bit like in grayscale images. With intelligent partitioning (IP) method, it can divides the audio waves into two parts, so that the image data hiding scheme can be applied to the audio. In order to maintain audio quality after embedding we improve IP method so that it can support data hiding method that we propose.

The embedding method that we proposed is called Random Reduce Difference Expansion (RRDE). Based on experimental result, the combination of two methods that we proposed RRDE and proposed IP 2 is produce the higher capacity compared to other methods. The highest SNR value is generated from RRDE method without using partitioning method. Where the average value of the resulting SNR is above 40 dB

Key Words: *data hiding, reduce difference expansion, intelligent partitioning, audio.*

(halaman ini sengaja dikosongkan)

KATA PENGANTAR

Segala Puji ke hadirat Allah SWT atas Rahmat, Nikmat dan Taufiknya, sehingga dapat diselesaikannya proposal tesis yang berjudul “Pengembangan Metode Data Hiding Pada Audio Berbasis Random Reduce Difference Expansion”. Penyusunan tesis ini merupakan salah satu syarat untuk memperoleh gelar **Magister Komputer** (M.Kom) pada Program Studi Magister, Bidang Keahlian Komputasi Berbasis Jaringan, Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

Dalam penyelesaian tesis ini penulis banyak mendapat bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu dengan segala kerendahan hati, penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya atas segala bimbingan, bantuan serta dukungannya kepada:

1. Bapak Tohari Ahmad, sebagai dosen pembimbing yang telah mengarahkan, membimbing, dan memotivasi penulis selama penyelesaian tesis ini.
2. Bapak Prof. Supeno Djanali, Muchammad Husni, Hudan Studiawan selaku dosen Program Pascasarjana Teknik Informatika, dan penguji yang telah memberikan bimbingan dan masukan ilmunya selama perkuliahan.
3. Ayah dan Ibu yang selalu memberikan dukungan yang sangat besar, baik secara material maupun secara spiritual sehingga penulis dapat menyelesaikan perkuliahan dan tesis dengan baik.
4. Kakak dan adik, terima kasih selalu menjadi pendukung dan penyemangat bagi penulis.
5. Sahabat-sahabat seperjuangan S2, mbak rizqa, mbak ira, mbak devy dan mbak ajeng. Terima kasih untuk senyum dan tawa kalian semua yang selalu menghibur penulis di saat sedih dan senang.
6. Teman-teman S2 Teknik Informatika FTIf yang tidak dapat disebutkan satu persatu. Terima kasih banyak sudah banyak membagi ilmu dan pengalaman terhadap penulis.

Akhirnya penulis menyadari masih banyak kekurangan dan kelemahan. Untuk itu saran dan kritik yang konstruktif akan sangat membantu agar tesis ini dapat menjadi lebih baik.

Surabaya, Juli 2016

Penulis

Dwi Shinta Angreni

DAFTAR ISI

ABSTRAK	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Kontribusi Penelitian	4
1.7 Sistematika Penulisan	5
BAB 2	7
KAJIAN PUSTAKA DAN DASAR TEORI	7
2.1 Pemrosesan Berkas Audio	7
2.2 Intelligent Partitioning (IP)	8
2.3 <i>Difference Expansion</i>	11
2.4 <i>Reduce Difference Expansion</i>	12
2.5 <i>Signal to Noise Ratio (SNR)</i>	13
BAB 3	15
METODE PENELITIAN	15
3.1 Tahapan Penelitian	15
3.2 Studi Literatur	16
3.3 Perancangan Algoritma	16
3.3.1 Metode Partisi Usulan 1	16
3.3.2 Metode Partisi Usulan 2	21

3.3.3	Proses <i>Encoding</i> Dan <i>Decoding</i> Tanpa Partisi	25
3.3.4	Metode Pengembangan <i>Reduce Difference Expansion</i>	27
3.3.5	<i>Multi Layer</i> dan <i>Multi Level Embedding</i>	31
3.4	Implementasi Penelitian.....	32
3.5	Perancangan Uji Coba dan Analisis Hasil	32
3.6	Penyusunan Buku Tesis	34
BAB 4	35
HASIL DAN PEMBAHASAN	35
4.1	Lingkungan Uji Coba.....	35
4.2	Pelaksanaan Uji Coba	35
4.3	Hasil dan Evaluasi	36
4.3.1	Skema Penyisipan <i>Multi-Layer</i> Dengan Metode Partisi Usulan 1 ..	37
4.3.2	Skema Penyisipan <i>Multi-Layer</i> Dengan Metode Partisi Usulan 2 ..	42
4.3.3	Skema Penyisipan <i>Multi-Layer</i> Tanpa IP	46
4.3.4	Berdasarkan Instrumen dan Genre Musik Dengan Metode Partisi Usulan 1	47
4.3.5	Berdasarkan Instrumen dan Genre Musik Dengan Metode Partisi Usulan 2.....	49
4.3.6	Berdasarkan Instrumen dan Genre Musik Tanpa IP.....	50
4.3.7	Uji Coba Berdasarkan Ukuran Pesan	54
4.3.8	Uji <i>Reversibility</i> Dari Hasil Proses <i>Decoding</i>	56
4.3.9	Perbandingan Dengan Metode Penyisipan Lain.....	58
BAB 5	61
KESIMPULAN DAN SARAN	61
5.1	Kesimpulan	61
5.2	Saran	61
DAFTAR PUSTAKA	63
BIODATA PENULIS	71

DAFTAR GAMBAR

Gambar 2.1 Representasi 16 bit <i>Array</i> sampel Audio (Choi et al., 2015)	9
Gambar 2.2 Total <i>Varians</i> per Segmen.....	10
Gambar 2.3 Proses Partisi Berdasarkan <i>Array S</i>	11
Gambar 3.1 Tahap Metodologi Penelitian	15
Gambar 3.2 Diagram Alir Proses Partisi Audio.....	18
Gambar 3.3 Diagram Alir Sistem Pada Proses <i>Encoding</i>	20
Gambar 3.4 Diagram Alir Sistem Pada Proses <i>Decoding</i>	21
Gambar 3.5 Bagan Alir Metode Partisi 2.....	22
Gambar 3.6 Diagram Alir Proses <i>Encoding</i> Metode Partisi 2	23
Gambar 3.7 Diagram alir proses <i>Decoding</i> metode partisi 2.....	24
Gambar 3.8 Diagram alir proses <i>Encoding</i> Tanpa Partisi.....	26
Gambar 3.9 Diagram alir proses <i>Decoding</i> Tanpa Metode Partisi	27
Gambar 3.10 Diagram Alir Proses Penyisipan	29
Gambar 3.11 Diagram Alir Proses Ekstraksi dan <i>Recovery</i>	31

(halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 4.1 Spesifikasi Lingkungan Perancangan	35
Tabel 4.2 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 10	38
Tabel 4.3 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 20	39
Tabel 4.4 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 30	39
Tabel 4.5 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 40	40
Tabel 4.6 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 50	41
Tabel 4.7 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 10	42
Tabel 4.8 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 20	43
Tabel 4.9 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 30	44
Tabel 4.10 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 40	45
Tabel 4.11 Hasil Pengujian <i>Multi-Layer</i> dengan Besar Segmen = 50	45
Tabel 4.12 Hasil Pengujian <i>Multi-Layer</i> Tanpa IP	47
Tabel 4.13 Hasil Pengujian Dengan Metode Partisi Usulan 1	48
Tabel 4.14 Hasil Pengujian Dengan Metode Partisi Usulan 2	50
Tabel 4.15 Hasil Pengujian Berdasarkan Genre Musik Tanpa IP	51
Tabel 4.16 Perbedaan Hasil Pengujian Dengan dan Tanpa Metode Partisi 1	52
Tabel 4.17 Perbedaan Hasil Pengujian Dengan dan Tanpa Metode Partisi 2	53
Tabel 4.18 Perbedaan Hasil Pengujian dengan Metode Partisi 1 dan 2	54
Tabel 4.19 Hasil SNR Berdasarkan Ukuran Pesan Metode Partisi 1	55
Tabel 4.20 Hasil SNR Berdasarkan Ukuran Pesan Metode Partisi 2	55
Tabel 4.21 Hasil Kapasitas Berdasarkan Ukuran Pesan Metode Partisi 1 dan 2 ..	56
Tabel 4.22 Hasil Pengujian Reversibility	57
Tabel 4.23 Hasil Perbandingan nilai SNR dengan Metode <i>Existing</i>	58
Tabel 4.24 Hasil Perbandingan Kapasitas dengan Metode <i>Existing</i>	59

(halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Media digital dapat memberikan akses dalam meningkatkan portabilitas, efisiensi, dan keakuratan informasi yang disajikan. Namun ada efek yang tidak diinginkan yang dapat timbul dari akses data ini yaitu terjadinya pelanggaran hak cipta dan perusakan atau manipulasi pada konten. Steganografi atau *data hiding* merupakan salah satu teknik yang dapat menyediakan perlindungan hak kekayaan intelektual, dan indikasi manipulasi konten. Steganografi merupakan proses yang digunakan untuk menyembunyikan data, seperti informasi hak cipta, dalam berbagai bentuk media seperti gambar, audio, atau teks dengan jumlah minimum degradasi yang dipahami oleh "host" sinyal; yaitu, data tertanam harus terlihat dan tak terdengar oleh pengamatan manusia. Perlu diketahui bahwa steganografi, mirip dengan kompresi, namun berbeda dari enkripsi. Tujuannya bukan untuk membatasi atau mengatur akses untuk *host* sinyal, melainkan untuk memastikan bahwa data yang tersembunyi utuh dan dapat dipulihkan (Bender et al. 1996).

Secara umum, steganografi atau *data hiding* diklasifikasikan ke dalam dua kategori yaitu *irreversibel* dan *reversibel*. Ketika menyembunyikan data tanpa menyimpan informasi pemulihan, citra stego akan terdistorsi secara permanen. *Least significant bit (LSB)* salah satu tipe *irreversibel data hiding* yang mengganti LSB dari host citra dengan pesan yang akan disembunyikan. Di sisi lain, jika informasi pemulihan dapat diperoleh oleh *decoder*, data asli dapat benar-benar pulih. Dalam beberapa aplikasi yang menyembunyikan data ke gambar penting, distorsi permanen pada gambar tidak dapat diterima. Misalnya, ketika menyembunyikan data ke dalam Gambar X-ray atau lukisan seni, gambar terdistorsi dapat mengakibatkan diagnosis yang salah medis atau penilaian yang salah dari sebuah karya seni sebagai pemalsuan (C.-C. Lin, Yang, & Hsueh, 2008).

Sejumlah metode *reversible data hiding* berbasis *integer transform* telah diusulkan. Tian (2003) mengusulkan teknik *reversible data hiding* dengan karakteristik kapasitas tinggi dan rendah distorsi dengan memanfaatkan *difference*

perluasan pasangan piksel. Nilai-nilai piksel dibagi menjadi pasangan dan pasangan piksel yang dipilih tidak boleh menyebabkan *overflow* dan *underflow*. kemudian data dan *location map* diembed ke dalam *difference* dari pasang piksel. Metode *reversible data hiding* usulan Tian disebut sebagai metode *difference expansion* (DE). Skema DE Tian menyebabkan terjadinya ekspansi ganda nilai-nilai piksel setelah pengolahan nilai piksel. Untuk mencegah pengurangan dari kualitas gambar, proses dengan nilai lebih besar dari 255 akan ditinggalkan. Hal ini akan menyebabkan pengurangan kapasitas penyisipan. Lou dkk. (2009) mengusulkan sebuah metode *reduce difference expansion* atau yang disebut metode RDE untuk memecahkan masalah diatas. Metode ini menyesuaikan ekspansi *difference* dengan menambahkan formula *reduce* pada nilai *difference* sebelum dilakukan penyisipan (Lou et al., 2009). (Alattar, 2004a) mengusulkan sebuah metode yaitu quad *difference expansion* (QDE) dengan menambah algoritma Tian lebih lanjut untuk menanamkan dua bit dalam setiap quad piksel yang berdekatan. Alattar (2004b) mengembangkan algoritma Tian menggunakan perbedaan perluasan vektor, bukan pasangan yang disebut *generalized difference expansion* (GDE), untuk meningkatkan kapasitas penyisipan dan efisiensi komputasi algoritma. Pendekatan ini memungkinkan algoritma untuk menyisipkan beberapa bit dalam setiap vektor dalam data gambar. Ahmad dkk. (2013) mengusulkan metode *reversible data hiding* dengan menggabungkan metode RDE Lou dan QDE Alattar. Metode ini menghasilkan kapasitas yang tinggi dengan kualitas gambar yang lebih baik dari metode sebelumnya.

Metode *reversible data hiding* diatas tidak hanya dapat diimplementasikan pada gambar tetapi juga pada media lain seperti audio. Pada tahun 2015 Choi dkk mengaplikasikan metode GDE pada audio. Tantangan utama untuk menyesuaikan skema *data hiding* gambar agar dapat bekerja pada domain audio dimana adanya perbedaan antara struktur gambar dan audio, target gambar skema ini adalah dua dimensi 8-bit gambar *grayscale* dan target audio adalah satu dimensi 16-bit kuantisasi dari gelombang audio. Isu dimensi dapat diselesaikan dengan modifikasi kecil, sementara perbedaan antara ukuran gambar piksel dan sampel gelombang ditangani oleh algoritma *intelligent partitioning* (IP) yang diusulkan, yang

membagi gelombang audio menjadi dua bagian, sehingga skema *data hiding* gambar dapat diterapkan pada audio (Choi, Pun, & Chen, 2015).

Secara umum metode-metode *data hiding* yang telah diusulkan diatas telah mampu merekonstruksi kembali data citra dan meningkatkan kapasitas penyisipan. Namun masih terdapat beberapa masalah yang belum dapat diatasi yaitu *overflow* dan *underflow*. *Overflow* adalah keadaan yang yang ditimbulkan jika setelah penyisipan dilakukan nilai piksel baru yang dihasilkan melebihi 255. Sedangkan *underflow* adalah keadaan yang ditimbulkan jika nilai piksel baru yang didapatkan setelah penyisipan kurang dari 0. Hal ini mengakibatkan proses penyisipan hanya dapat dilakukan pada piksel tertentu yang tidak menimbulkan *overflow* dan *underflow*, yang pada akhirnya akan membatasi tempat penyisipan pada *cover*. Oleh karena itu diusulkan sebuah metode penyisipan yang dapat mengatasi *underflow* dan *overflow* dimana setiap nilai piksel yang ada pada *cover* data adalah *expandable* atau dapat diekspansi. Metode ini juga menggunakan *difference* dari pasangan piksel *cover* dan nilai random sebagai tempat ekspansi sehingga dapat meningkatkan kapasitas penyisipan dari *cover* data sebanyak dua kali dari metode sebelumnya. Dalam penelitian ini juga diusulkan pengembangan metode *intelligent partition* yang nantinya akan menjadi pendukung metode penyisipan diterapkan pada audio. Metode ini diharapkan dapat mempertahankan kualitas berkas audio setelah dilakukan penyisipan.

1.2 Perumusan Masalah

Dalam penelitian ini, masalah-masalah yang akan diselesaikan dirumuskan sebagai berikut:

1. Bagaimana cara mengatasi masalah *overflow* dan *underflow* dengan menggunakan pengembangan metode RDE pada berkas audio?
2. Bagaimana mempertahankan kualitas berkas audio setelah dilakukan partisi dengan menggunakan pengembangan metode *intelligent partitioning* (IP) pada audio?
3. Bagaimana pengaruh skema Multi-layer terhadap kualitas dan kapasitas berkas audio?

1.3 Batasan Masalah

Permasalahan yang dibahas pada penelitian ini memiliki beberapa batasan sebagai berikut:

1. File audio yang digunakan berformat wav dan terdiri dari dua channel atau stereo.
2. Pesan rahasia yang digunakan berupa teks.
3. *Location map* terpisah dari *cover* medianya.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengatasi masalah *overflow* dan *underflow* dengan menggunakan pengembangan metode RDE pada berkas audio.
2. Mempertahankan kualitas data audio setelah dilakukan partisi dengan menggunakan pengembangan metode IP pada audio.
3. Melihat pengaruh skema Multi-layer terhadap kualitas dan kapasitas berkas audio.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat menerapkan metode penyisipan yang digunakan pada domain citra agar dapat digunakan pada domain audio. Selain itu juga diharapkan dapat meningkatkan kualitas hasil berkas audio yang telah disisipi data sehingga tidak menimbulkan kecurigaan bahwa berkas audio telah dimodifikasi. Kemudian meningkatkan kapasitas data pada berkas audio sehingga lebih banyak informasi penting yang dapat disisipkan. Serta dapat mengembalikan berkas audio ke bentuk originalnya tanpa merusak informasi yang terdapat di dalam berkas audio tersebut.

1.6 Kontribusi Penelitian

Kontribusi dari penelitian ini diantaranya adalah pengembangan metode RDE dengan menggunakan *difference* dari pasangan nilai *cover* media dan nilai

random sebagai tempat penyisipan. Selain itu penelitian ini juga mengembangkan metode *Intelligent partitioning* (IP) pada audio yang memfokuskan penempatan dua *bigit* terakhir dari 16 *bigit* audio yang masing-masing menjadi LSB dari dua partisi yang dihasilkan.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ditunjukkan untuk memberikan gambaran dan uraian dari penelitian, secara garis besar yang meliputi beberapa bab, sebagai berikut:

BAB 1 : PENDAHULUAN

Bagian pendahuluan menguraikan tentang latar belakang masalah yang kemudian dirumuskan kedalam rumusan masalah dalam bentuk uraian terstruktur dan dilengkapi dengan, batasan masalah, tujuan, manfaat, kontribusi serta sistematika penulisan.

BAB 2 : TINJAUAN PUSTAKA

Bagian tinjauan pustaka melakukan pengkajian mengenai teori-teori dan referensi yang berkaitan dan menunjang penelitian.

BAB 3: METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan mengumpulkan data permasalahan. Data selanjutnya diolah dengan menggunakan metode yang diajukan untuk mendapatkan pengetahuan serta informasi yang bisa dimanfaatkan oleh pengguna.

BAB 4 : HASIL DAN PEMBAHASAN

Bab ini akan dijabarkan dan dijelaskan hasil yang diperoleh dari hasil penelitian berupa analisa terhadap pengetahuan yang dihasilkan dari pengguna menggunakan metode yang diusulkan

BAB 5 : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diambil berdasarkan analisa setelah pengujian serta saran untuk pengembangan lebih lanjut.

(halaman ini sengaja dikosongkan)

BAB 2

KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini akan dibahas beberapa teori dasar yang menunjang dalam pembuatan Tesis.

2.1 Pemrosesan Berkas Audio

Pemrosesan sinyal audio digital sering digunakan dalam rekaman dan penyimpanan musik dan *speech* sinyal, untuk audio mixing dan produksi program digital, dalam transmisi digital untuk disiarkan kepada penerima dan juga dalam produk untuk konsumen seperti CD, DATs dan PC. Dalam kedua kasus, sinyal audio dalam bentuk digital dari mikrofon sampai ke pengeras suara, yang memungkinkan pengolahan real-time prosesor sinyal digital dengan cepat (Zölzer, 2008).

Basik operasi untuk sinyal kontinu pada audio digital adalah *sampling* dan *quantization*. Proses ini dilakukan untuk mengubah sinyal audio yang kontinu menjadi diskrit. Nilai diskrit inilah yang akan digunakan untuk kemudian diolah pada proses selanjutnya.

Tahap pertama yang dilakukan adalah *sampling* berkas audio. *Sampling* adalah konversi fungsi kontinu terhadap waktu $x(t)$ menjadi urutan angka-angka $x(n)$ atau juga sering disebut konversi dari analog ke digital (AD konversi). Sedangkan proses sebaliknya disebut konversi digital ke analog (DA konversi). *Sampling* fungsi $x(t)$ dijelaskan oleh Shannon *sampling* teorema menyatakan bahwa sinyal kontinu dengan *bandwidth* fB dapat *disampling* dengan *sampling rate* $fs > 2fB$ tanpa mengubah isi informasi dalam sinyal. Sinyal analog asli direkonstruksi oleh *low-pass filtering* dengan *bandwidth* fB .

Tahap selanjutnya adalah *quantization* dari nilai *sampling* yang telah didapatkan sebelumnya. *Quantization* adalah digitalisasi sinyal sampel terhadap amplitudo secara terus menerus. Nilai amplitudo yang didapatkan akan direpresentasikan menjadi bilangan bulat atau desimal.

Pulse Code Modulation (PCM) adalah salah satu metode yang menggunakan *sampling* dan *quantization* dalam mengubah sinyal analog menjadi

format digital (Faruque, 2015). Ini adalah bentuk standar dari audio digital pada komputer, *Compact Discs*, telepon digital dan aplikasi digital audio lainnya. Pada PCM, amplitudo dari sinyal analog di *sampling* secara regular pada interval yang seragam, dan setiap sampel dikuantisasi ke nilai terdekat dalam sebuah tingkat dari tangga digital.

PCM adalah format audio *uncompressed* atau *lossless* yang sering dikenal dan dipakai. WAV dan AIFF adalah dua contoh *lossless* audio dengan format berbasis PCM. Keduanya adalah format berkas audio fleksibel yang didisain untuk menyimpan kurang lebih setiap kombinasi dari *sampling rates* atau *bitrates*. Hal ini membuat berkas audio ini sangat cocok untuk penyimpanan rekaman original. PCM audio biasanya juga berupa berkas PCM audio stereo (*2-channel*), dengan sampel 44.1 kHz atau 44100 kali per detik, dengan bit *depth* 16 dengan besar mencapai 10 MB per menit.

Penelitian ini menggunakan salah satu berkas audio PCM yaitu WAV. Dengan menggunakan berkas audio ini, detail pada audio tidak akan hilang jika audio analog diubah dan disimpan menjadi digital audio. Ini membuat format WAV menjadi pilihan untuk mengedit audio dengan *high-fidelity*.

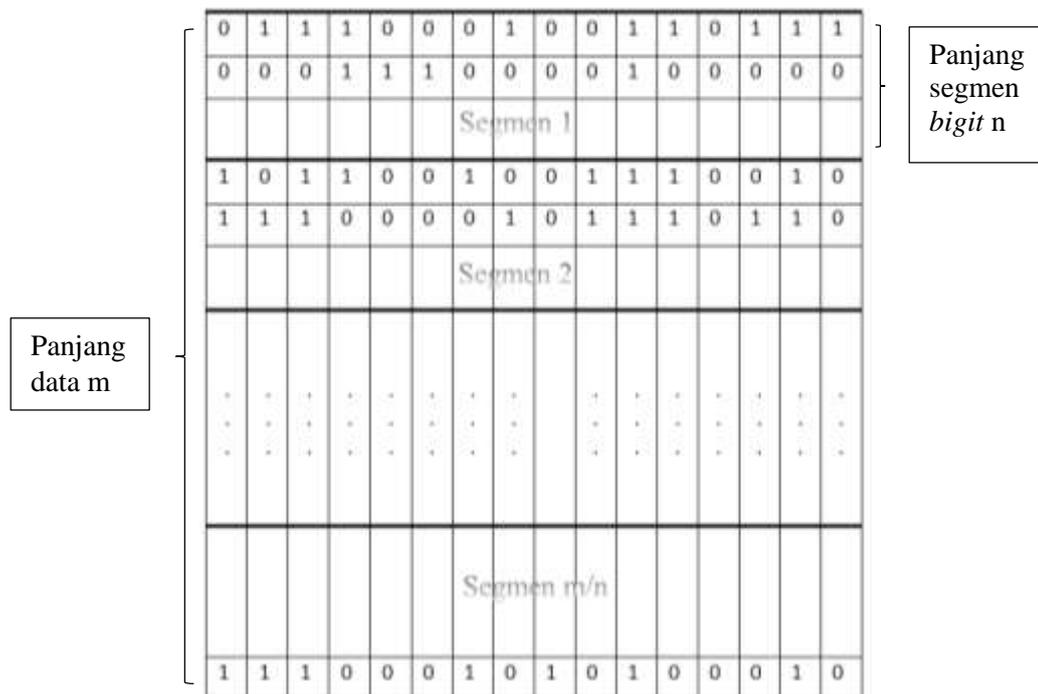
2.2 Intelligent Partitioning (IP)

Pada subbab ini membahas tentang *intelligent partitioning* (Choi et al., 2015) metode ini adalah metode partisi yang digunakan mempartisi berkas data menjadi dua atau lebih dengan ukuran bit yang semakin kecil. Karena adanya perbedaan besar sampel pada berkas citra dan audio, berkas audio memiliki besar sampel sebesar 16 bit sedangkan pada citra 8 bit, untuk dapat mengaplikasikan metode data *hiding* pada berkas audio maka dilakukan tahap *pre-processing* pada berkas audio dengan mengubah besar sampel audio menjadi 8 bit. Dengan metode IP, berkas audio yang awalnya memiliki besar 16 bit dipartisi menjadi dua bagian yang masing-masing memiliki besar 8 bit. Kemudian setelah terbentuk dua partisi 8 bit penyisipan data *payloads* dapat dilakukan pada masing-masing bagian tersebut. Partisi dilakukan dengan cara merepresentasikan gelombang audio sebagai

16 buah *array* seperti pada Gambar 2.1 dimana masing-masing *array* ini disebut *bigit*. Untuk menentukan pembagian kelompok dari tiap *bigit* maka kita perlu menentukan berapa besar segmen tiap *bigit*, kemudian menghitung nilai *variance* dari tiap segmen *bigit* pada sampel audio. Untuk sebuah *array* sampel M , nilai *variance* V didapatkan dengan menggunakan persamaan 2.1.

$$v(x) = \sum_{j=1}^n (x_j - a(x))^2 \quad (2.1)$$

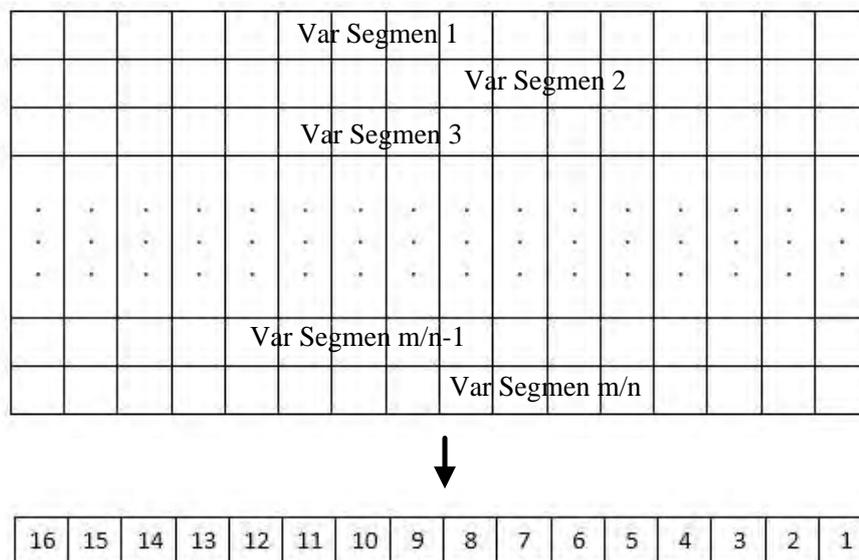
Dimana x adalah vektor $(x_1, x_2, x_3 \dots x_n)$ yang merepresentasikan satu buah segmen *bigit* dan $a(x)$ adalah rata-rata dari x yang dibulatkan. Untuk mencari V , *variance* dari semua segmen untuk setiap *bigit* harus dihitung terlebih dahulu, kemudian semua *variance* tersebut dijumlahkan untuk menghasilkan V . Nilai *varians* didapatkan dari setiap *bigit* berikut tahapan-tahapan yang dilakukan selanjutnya.



Gambar 2.1 Representasi 16 bit *Array* sampel Audio (Choi et al., 2015)

1. Ambil indeks dari array S yang disusun dari indeks 1-16 berdasarkan nilai $V(i)$ dari besar ke kecil. Tahap ini dapat dilihat pada Gambar 2.1 dan 2.2.

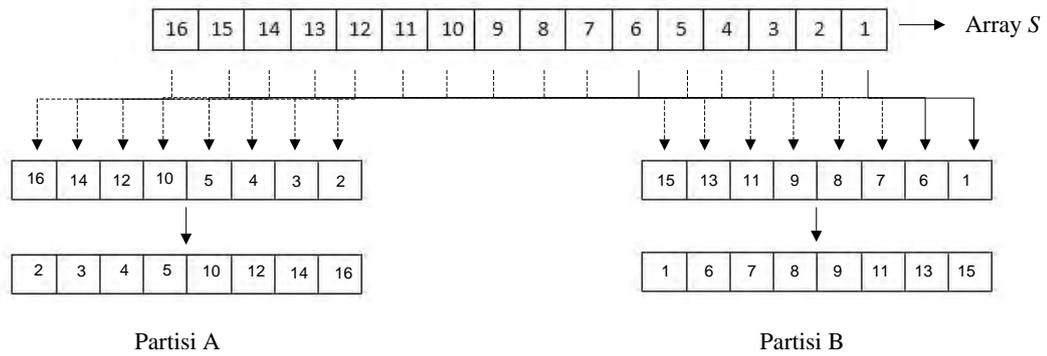
2. Ambil delapan tempat paling signifikan yaitu 9-16 kedalam dua grup A dan B secara berturut-turut.
3. Kemudian masukkan elemen pertama dan ketiga dari S ke grup A , dan elemen kedua dan keempat dari S ke grup B .
4. Masukkan empat elemen S selanjutnya kedalam grup A dan grup B , dengan menematkan dua elemen masing-masing pada setiap grup. Dengan tujuan setelah diurutkan grup A dan grup B memiliki *bigit* dengan nilai *varians* $V(i)$ yang tinggi ada pada urutan pertama atau kedua terkecil. Tahap ini dapat dilihat pada Gambar 2.3.



Gambar 2. 2 Total *Varians* Setiap *Bit*

5. Partisi M , berdasarkan urutan A dan B dengan elemennya menjadi dua partisi yaitu M_1 dan M_1 .

A dan B direpresentasikan sebagai array partisi 16 bit P yang akan digunakan kembali pada tahap *decoding*. Tahap *decoding* dilakukan dengan menyusun kembali sampel audio yang telah dipartisi sesuai dengan bit P .



Gambar 2. 3 Proses Partisi Berdasarkan Array S

2.3 Difference Expansion

Difference Expansion (DE) adalah sebuah teknik *reversible data hiding* yang diperkenalkan oleh (Tian, 2003) dimana teknik ini memberikan ruang penyimpanan tambahan pada *difference* dari nilai piksel citra. Metode DE diawali dengan *reversible integer transform* dengan menggunakan sebuah citra *grayscale* 8 bit (x, y) , $0 \leq x, y \leq 255$, menghitung nilai rata-rata integer l dan selisih h dengan persamaan (2.2) berikut

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor, \quad h = x - y \quad (2.2)$$

dengan *Inverse transform* dari persamaan (2.2) dapat dilihat pada persamaan (2.3).

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor, \quad y = l - \left\lfloor \frac{h}{2} \right\rfloor \quad (2.3)$$

Overflow dan *underflow* adalah tahapan masalah yang sering muncul paada saat penyisipan dimana *overflow* adalah kondisi piksel hasil modifikasi bernilai lebih dari 255, sedangkan *underflow* adalah kondisi ketika piksel hasil modifikasi bernilai kurang dari 0. Hal ini mengakibatkan penyisipan bit pesan b tidak mungkin dilakukan, maka untuk membuat penyisipan tetap dapat dilakukan nilai data hasil penyisipan h harus memenuhi persamaan (2.5)

$$|h| \leq 2\lfloor 255 - l \rfloor, \text{ and } |h| \leq 2l + 1 \quad (2.4)$$

$$\begin{cases} |h| \leq 2(255 - l), \text{ jika } 128 \leq l \leq 255 \\ |h| \leq 2l + 1, \text{ jika } 0 \leq l \leq 127 \end{cases} \quad (2.5)$$

Penyisipan bit data b pada nilai *difference* h dengan DE, maka *difference* expanded h' akan menjadi seperti pada persamaan (2.6), berdasarkan nilai data hasil penyisipan h' dan nilai rata-rata l maka dapat nilai x' dan y' dapat dihitung dengan persamaan (2.7).

$$h' = 2h + b \quad (2.6)$$

$$x' = l + \left\lfloor \frac{h'+1}{2} \right\rfloor, \quad y' = l - \left\lfloor \frac{h'}{2} \right\rfloor \quad (2.7)$$

Sedangkan pada proses ekstraksi data hampir memiliki langkah yang sama dengan proses penyisipan yaitu mengitung kembali l' dan h' dengan persamaan (2), namun untuk mendapat nilai b dan nilai h proses perhitungan menggunakan persamaan (2.8)

$$b = LSB(h'), \text{ dan } h = \left\lfloor \frac{h'}{2} \right\rfloor \quad (2.8)$$

Maka dengan nilai l' dan h yang telah didapatkan kita dapat menghitung kembali nilai pasangan (x, y) .

2.4 Reduce Difference Expansion

Reduce difference expansion atau RDE adalah metode reversibel *data hiding* yang disulkan (Lou et al., 2009) yang mengembangkan metode DE Tian dengan menambahkan formua reduksi pada nilai *difference* dari pasangan piksel sebelum dilakukan proses penyisipan.

Metode ini menggunakan fungsi transformasi untuk mengurangi nilai dari *difference*. Fungsi transformasi yang digunakan direpresentasikan sebagai:

$$h' = \begin{cases} h & \text{if } h' < 2 \\ h - 2^{\lfloor \log_2 h \rfloor - 1} & \text{otherwise} \end{cases} \quad (2.9)$$

Dimana h' menotasikan nilai *difference* yang asli.

Untuk dapat mengembalikan nilai *difference* yang asli, dibentuk sebuah *location map* diwaktu nilai *difference* yang asli diubah. Ukuran dari *location map* adalah sama dengan banyak pasangan nilai piksel. Jika nilai *difference* h adalah 0 atau 1, maka nilai piksel tidak diubah dan *location map* yang disimpan adalah 0. Ketika $h = 2$ dan $h' = 1$, *location map* yang disimpan adalah 1.

Nilai dari *location map* ditentukan menurut :

$$location\ Map = \begin{cases} 0 & \text{if } 2^{\lfloor \log_2 h' \rfloor} = 2^{\lfloor \log_2 h \rfloor} \text{ atau } h' = h \\ 1 & \text{if } 2^{\lfloor \log_2 h' \rfloor} \neq 2^{\lfloor \log_2 h \rfloor} \end{cases} \quad (2.10)$$

Sebagai contoh, perhitungan pada pasangan piksel yang menghasilkan $h=5$, nilai rata-rata $l=203$, dan $h'(h' = h - 2^{\lfloor \log_2 h \rfloor - 1} = 5 - 2^1 = 3)$. Kemudian bit data $b=1$ disisikan untuk menghasilkan $h'' = 2 \times h' + b = 2 \times 3 + 1 = 7$ dan *location map* = 1. Selanjutnya pasangan piksel baru bisa didapatkan dengan menggunakan persamaan berikut.

$$\begin{aligned} x' &= l + \left\lfloor \frac{h''+1}{2} \right\rfloor = 203 + \left\lfloor \frac{7+1}{2} \right\rfloor = 207, \\ y' &= l - \left\lfloor \frac{h''}{2} \right\rfloor = 203 - \left\lfloor \frac{7}{2} \right\rfloor = 200 \end{aligned} \quad (2.11)$$

Metode RDE jika dibandingkan dengan DE, dapat mengurangi nilai *differencenya* sehingga masalah *overflow* dan *underflow* dapat dikontrol dan meningkatkan kualitas dari citranya (Lou et al., 2009).

2.5 Signal to Noise Ratio (SNR)

Pengukuran menggunakan SNR tetap banyak digunakan untuk penilaian kualitas persepsi pada watermarking audio, hal ini dikarenakan kesederhanaannya (Liu, Jellyman, Mason, Evans, & others, 2006). Menurut (Quackenbush, Barnwell, & Clements, 1988) definisi SNR adalah dimana $\sum_n [s_o(n)]^2$ adalah jumlah dari *host signal* s_o yang dipangkatkan dua dan $\sum_n [s_w(n) - s_o(n)]^2$ adalah jumlah dari *noise* yang disebabkan setelah proses penyisipan (Y. Lin & Abdulla, 2011). Formula SNR dapat dilihat pada persamaan (2.12).

$$SNR(s_o, s_w) = 10 \log_{10} \frac{\sum_n [s_o(n)]^2}{\sum_n [s_w(n) - s_o(n)]^2} \quad (2.12)$$

Semakin besar nilai SNR maka kemiripan dari *host signal* dan *signal* yang telah dimodifikasi atau diberi *noise* juga semakin besar. Sedangkan jika nilai SNR yang dihasilkan infinitif maka *host signal* dan *signal* modifikasi sama atau identik.

(halaman ini sengaja dikosongkan)

BAB 3 METODE PENELITIAN

Bab ini berisi pembahasan mengenai langkah penelitian dan jadwal kegiatan penelitian. Langkah penelitian dibagi menjadi enam tahap yakni studi literatur, perancangan algoritma, implementasi penelitian, perancangan uji coba dan evaluasi, analisis hasil dan penyusunan buku tesis. Selanjutnya jadwal kegiatan penelitian memuat garis waktu dari semua langkah penelitian selama empat bulan.

3.1 Tahapan Penelitian

Dalam rangka mencapai tujuan yang diharapkan dalam penelitian tesis ini akan dilakukan beberapa langkah. Langkah-langkah tersebut dijelaskan pada Gambar 3.1. Langkah pertama adalah studi literatur untuk mempelajari permasalahan dan penelitian terkini. Kemudian perancangan algoritma yang diusulkan yaitu pengembangan metode RDE dan IP. Dilanjutkan dengan uji coba metode yang diusulkan. Kemudian hasil dari uji coba tersebut dilakukan analisis. Dan diakhiri dengan penyusunan buku tesis.



Gambar 3.1 Tahap Metodologi Penelitian

Secara umum, luaran yang diharapkan dari penelitian ini adalah sistem yang mampu melakukan data *reversible data hiding* pada audio yang dapat meningkatkan kualitas dan kapasitas hasil citra stego. Sistem ini menggunakan

metode penyisipan pengembangan RDE dan metode partisi pengembangan IP dengan penyisipan multi *layer* dan multi level.

3.2 Studi Literatur

Studi literatur merupakan tahap untuk mempelajari konsep, teori, fakta dan informasi serta mencari referensi yang diperlukan untuk mendukung penelitian. Referensi yang digunakan dapat berasal dari jurnal, artikel, buku dan lainnya. Informasi dan referensi yang didapatkan kemudian akan dikaji dan diterapkan untuk perancangan algoritma dalam menyelesaikan penelitian ini. Topik-topik literatur yang dikumpulkan adalah sebagai berikut:

1. Metode RDE dan perkembangannya.
2. Metode partisi berkas audio yaitu IP dan perkembangannya.
3. Pemrosesan dan pengambilan informasi pada berkas audio

3.3 Perancangan Algoritma

Pada sub bab ini akan dijelaskan mengenai perancangan algoritma dari proses-proses yang akan dilakukan untuk mengimplementasikan penelitian. Tahapan proses yang dijelaskan untuk memberikan informasi detail mengenai alur-alur algoritma pada sistem *data hiding* audio.

3.3.1 Metode Partisi Usulan 1

Metode partisi pertama yang diusulkan adalah pengembangan metode IP (Choi et al., 2015). Metode ini digunakan untuk mempartisi berkas audio yang memiliki panjang 16 bit menjadi 2 partisi yang memiliki panjang 8 bit. Partisi ini dilakukan agar metode penyisipan gambar dapat diimplementasikan pada berkas audio. Metode IP membagi sampel audio berdasarkan nilai *varians* per segmen yang setiap bigitnya akan ditotal. Nilai total *varians* setiap bigit ini yang nantinya

akan diurutkan dari besar kekecil. Nilai indeks dari data urutan *varians* ini kemudian dibagi menjadi dua partisi berdasarkan aturan yang ada pada metode IP.

Metode partisi yang diusulkan ini menempatkan dua bit terakhir dari sampel data menjadi LSB dari masing masing partisi yang dihasilkan. Hal ini dilakukan untuk membagi rata penyebaran bit akhir sampel audio. Karena bit akhir sampel audio merupakan bit yang akan selalu berubah jika dilakukan penyisipan. Sehingga jika dijadikan LSB dari setiap partisi maka dapat mencegah terjadi perubahan yang besar jika dilakukan penyisipan.

Langkah awal partisi adalah dengan mengubah data sampel audio yang awalnya berbentuk *float* menjadi nilai *integer*. Data *sampling* awal yang didapatkan berada dalam *range* -1 sampai 1 karena data ini tidak dapat diubah kebentuk *binary* maka terlebih dahulu akan dilakukan proses normalisasi untuk dapat mengubahnya kedalam bentuk *integer*. Proses normalisasi dilakukan dengan menggunakan persamaan (3.1).

$$y = y_0 \left(\frac{2^{nbit}}{2} \right) + \left(\frac{2^{nbit}}{2} \right) \quad (3.1)$$

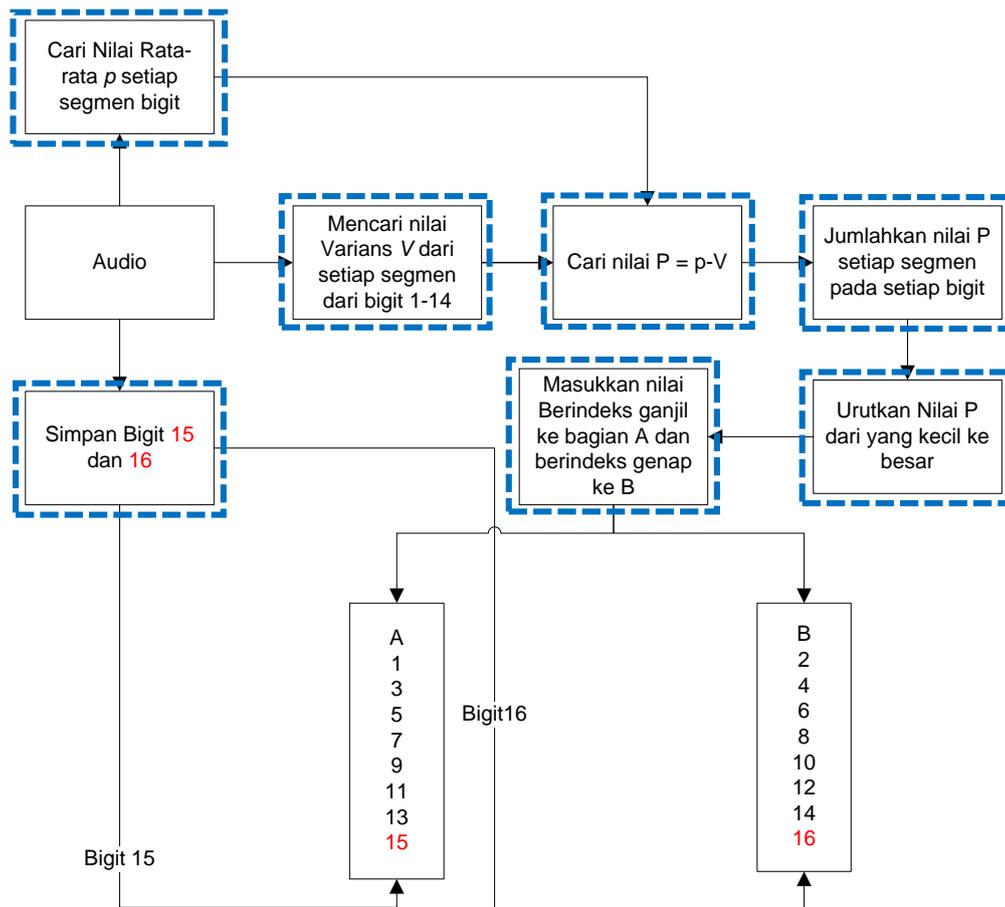
Dimana y adalah data sampel audio hasil normalisasi yang berbentuk *integer*, y_0 adalah data sampel audio dan $nbit$ adalah besar *bit depth* dari sampel audio. Persamaan (3.2) digunakan untuk mengembalikan data bentuk *integer* kebentuk semula.

$$y_0 = \frac{y - \left(\frac{2^{nbit}}{2} \right)}{\left(\frac{2^{nbit}}{2} \right)} \quad (3.2)$$

Setelah didapatkan nilai *integer* setelah proses normalisasi maka selanjutnya data akan diubah kebentuk *binary* sehingga dihasilkan 16 array *bigit* audio. Proses partisi dapat dilihat pada Gambar 3.2. Pertama, hal yang dilakukan pada tahap partisi ini adalah menentukan panjang segmen per *bigit*. Hitung *varians* dari setiap segmen pada *bigit* 1 sampai 14 menggunakan persamaan (3.3)

$$s^2(x) = \frac{\sum x^2 - \frac{(\sum x)^2}{n}}{n-1} \quad (3.3)$$

dimana x adalah vektor (x_1, x_2, \dots, x_n) yang merupakan nilai-nilai yang ada pada satu segmen pada *bigit*. Sedangkan n adalah panjang segmen per *bigit*. Pada gambar, yang ditunjukkan dengan kotak garis biru merupakan kontribusi yang diusulkan pada metode partisi ini.



Gambar 3.2 Diagram Alir Proses Partisi Audio

Kemudian hitung nilai rata-rata x yang ada pada setiap segmen lalu kurangi dengan nilai *variansnya* seperti yang ditunjukkan pada persamaan (3.4).

$$p = \frac{\sum x}{n} - s^2(x) \quad (3.4)$$

Dimana p adalah nilai perbedaan dari nilai rata-rata dan nilai *varians*. Proses ini ditambahkan untuk mencari nilai dengan persebaran angka satu yang lebih besar. Dengan mengetahui persebaran angka satu yang lebih besar pada setiap bigit dapat mempermudah untuk menyusun partisi sehingga nantinya akan menghasilkan

nilai partisi yang kecil. Jika nilai partisi yang dihasilkan kecil maka kemampuan ekspansi dari partisi tersebut akan menjadi besar.

Selanjutnya jumlahkan nilai p pada setiap segmen *bigit* dengan menggunakan persamaan (3.5).

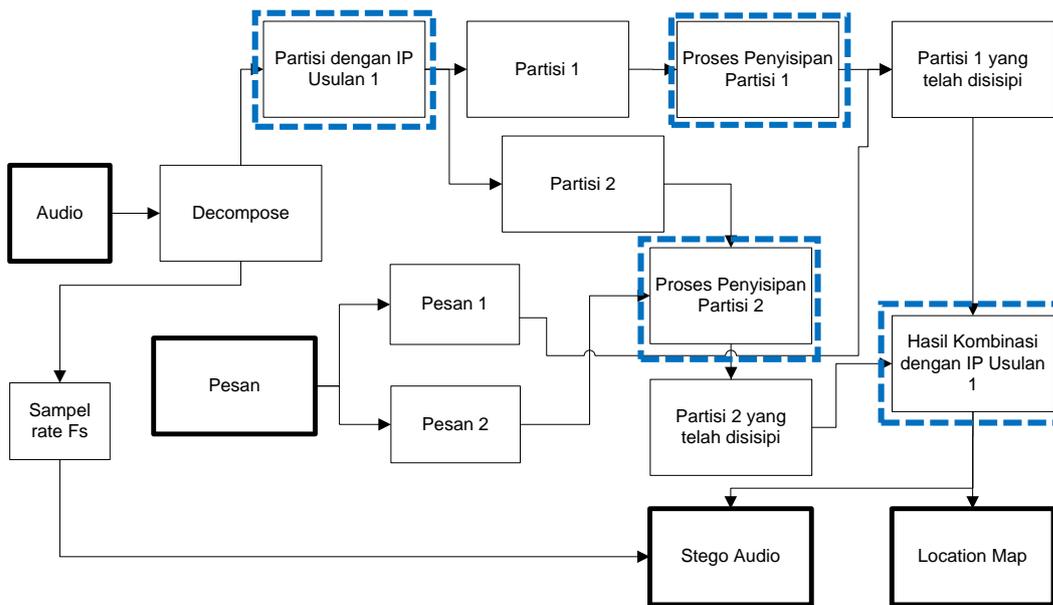
$$P = \sum p \quad (3.5)$$

Nilai akhir yang akan didapatkan adalah jumlah nilai p pada setiap *bigit*. Urutkan nilai rata-rata p dari nilai yang terendah ke yang tertinggi. Tempatkan *bigit* berindeks ganjil yaitu 1, 3, 5, 7, 9, 11, dan 13 pada partisi *A* dan *bigit* berindeks genap yaitu 2, 4, 6, 8, 10, 12, dan 14 pada partisi *B*. Kemudian dua *bigit* terakhir yang kita sisakan yaitu *bigit* ke 15 dan 16 ditempatkan berturut-turut pada partisi *A* dan *B*. Penentuan pembagian partisi berdasarkan ganjil genap bertujuan untuk membagi persebaran data pada partisi lebih merata sehingga dihasilkan nilai partisi yang baik.

Secara garis besar sistem yang akan dibangun terdiri dari dua proses yaitu *encoding* dan *decoding*. *Encoding* merupakan proses yang terdiri proses partisi data *sampling* audio dan proses penyisipan. Sedangkan *decoding* adalah proses ekstraksi dan *recovery*. Kedua tahap ini akan dijelaskan lebih detail pada subbab selanjutnya.

3.3.1.1 Proses *Encoding*

Terdapat dua proses utama yang ada pada sistem ini yaitu proses *encoding* dan *decoding*. Pada proses *encoding* yang ditunjukkan pada Gambar 3.3, tahap yang dikotak biru dengan garis putus-putus menunjukkan kontribusi yang diusulkan pada metode ini. Pada gambar terdapat beberapa tahap yang dilakukan yaitu pertama, meliputi tahap *decompose* membaca sampel pada berkas audio dan merepresentasikannya dalam bentuk *float* atau *integer*. Mempartisi sampel dengan IP menghasilkan dua buah partisi yang memiliki besar 8 bit. Pesan yang akan disisipkan juga akan dibagi dua menyesuaikan dengan hasil partisi sampel audio yang dihasilkan. Kemudian pada masing-masing bagian sampel disisipkan pesan yang sudah dibagi menjadi dua dengan menggunakan pengembangan metode RDE.



Gambar 3.3 Diagram Alir Sistem Pada Proses *Encoding*

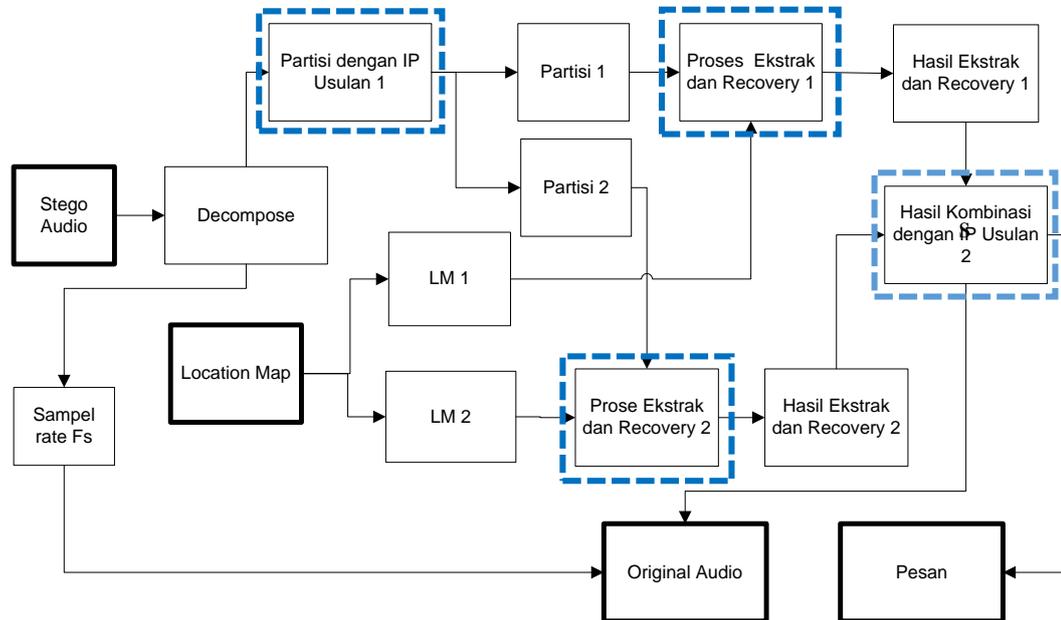
Setelah proses penyisipan, sistem akan menyusun kembali dua bagian sampel menjadi satu bagian utuh sesuai dengan indeks awal data sampel. Sistem kemudian menulis kembali berkas audio hasil *encoding* dengan sampel *rate Fs* kedalam bentuk .wav bersama dengan berkas tambahan yang berisikan *location map* yang akan digunakan oleh sistem dalam melakukan proses *decoding*.

3.3.1.2 Proses *Decoding*

Proses *decoding* juga terdiri dari beberapa tahap, diagram alir proses *decoding* dapat dilihat pada Gambar 3.4. Tahap awal yang dilakukan pada proses *decoding* yaitu membaca sampel dari berkas audio hasil *encode*. Kemudian sistem akan membagi kembali sampel menjadi dua bagian yang memiliki besar 8 bit sesuai dengan *array* partisi yang telah dibaca. Setelah itu data *payload* akan diekstrak dengan mengambil nilai LSB setiap datanya.

Kemudian dilakukan *recovery* berkas audio dengan melihat *location map*. Data *payload* yang telah didapatkan akan direkonstruksi kembali menjadi seperti semula. Setelah itu dua partisi ini akan digabungkan membentuk berkas audio yang sama seperti semula. Proses penggabungan dilakukan sesuai dengan susunan *array*

partisi sehingga data yang dipartisi akan terekonstruksi sesuai dengan data asli audio.



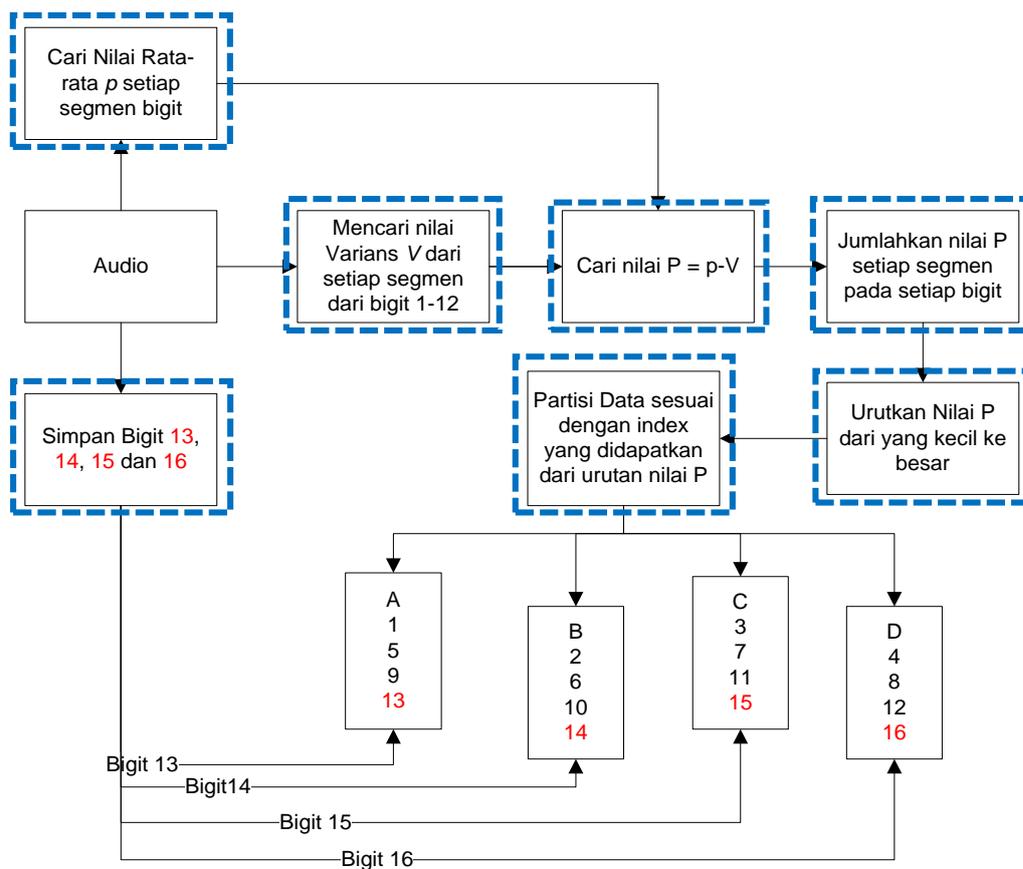
Gambar 3.4 Diagram Alir Sistem Pada Proses *Decoding*

3.3.2 Metode Partisi Usulan 2

Metode ini merupakan usulan kedua dari penelitian ini yang juga merupakan pengembangan metode IP (Choi et al., 2015). Pada metode usulan ini berkas audio yang memiliki panjang 16 bit akan dipartisi menjadi empat dengan panjang 4 bit per sampel. Data berdimensi 4 bit adalah $(x, y) \in \mathbb{Z}, \exists 0 \leq x, y \leq 15$. Penggunaan metode partisi ini hanya dapat dilakukan pada metode penyisipan dengan perubahan yang terjadi pada proses penyisipan hanya terbatas pada data berdimensi 4 bit. Jika perubahan yang terjadi melebihi 15 atau dibawah 0 maka penyisipan tidak dapat dilakukan. Oleh karena itu untuk dapat mengimplementasikan DE dan RDE pada metode ini harus ada aturan-aturan tambahan yang harus ditambahkan sehingga penyisipan tetap dapat dilakukan.

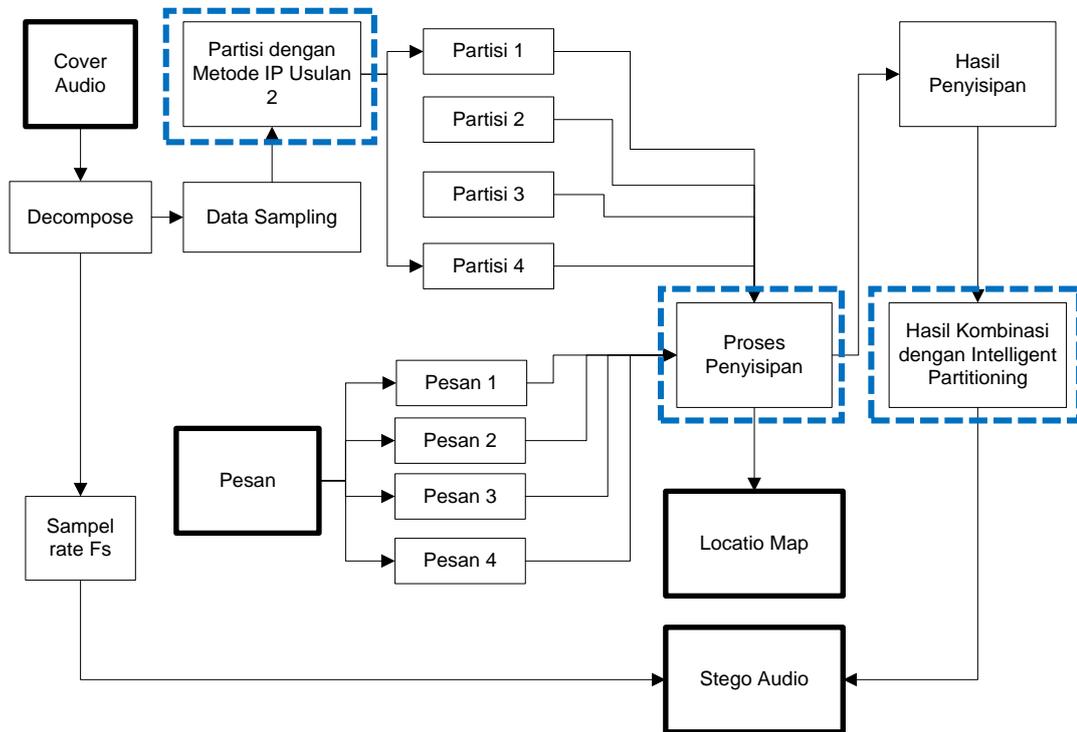
Langkah-langkah metode partisi kedua yang diusulkan dapat dilihat pada Gambar 3.5. Pertama, baca file audio M dan mengubahnya menjadi 16 bits array. Kemudian tentukan panjang segmen per *bigit*. Hitung *varians* dari setiap segmen

pada *bit* 1 sampai 12 menggunakan persamaan (3.3) dimana x adalah vektor (x_1, x_2, \dots, x_n) yang merupakan nilai-nilai yang ada pada satu segmen pada *bit*. Sedangkan n adalah panjang segmen per *bit*. Sisihkan *bit* 13 sampai 16 untuk digunakan pada tahap terakhir. Hitung nilai penyebarannya dengan menjumlahkan nilai x yang ada pada setiap segmen kemudian kurangi dengan nilai *variansnya* seperti yang ditunjukkan pada persamaan (3.4). Jumlahkan nilai p pada setiap segmen *bit* sesuai dengan persamaan (3.5). Nilai akhir yang akan didapatkan adalah jumlah nilai p pada setiap *bit*. Urutkan nilai rata-rata p dari nilai yang terendah ke yang tertinggi. Tempatkan *bit* berindeks 1, 5, dan 9 pada partisi A, 2, 6, dan 10 pada partisi B, 3, 7, dan 11 pada partisi C, dan 4, 8, dan 12 pada partisi D. Empat *bit* terakhir yang kita sisakan yaitu *bit* ke 13, 14, 15 dan 16 ditempatkan berturut-turut pada partisi A, B, C dan D.



Gambar 3.5 Bagan Alir Metode Partisi 2

Pada metode ini tahapan penggabungan memiliki cara yang sama dengan tahap partisi sehingga sama seperti metode usulan pertama, metode ini tidak perlu menyimpan array indeks partisi A , B , C dan D dalam proses penggabungan kembali. Sehingga metode partisi ini dapat dilakukan lagi pada tahap *decoding*. Akan tetapi jika pada proses penyisipan perubahan yang terjadi tidak hanya pada LSB partisi maka, indeks partisi harus tetap disimpan dan dipakai pada proses *decoding*.



Gambar 3.6 Diagram Alir Proses *Encoding* Metode Partisi 2

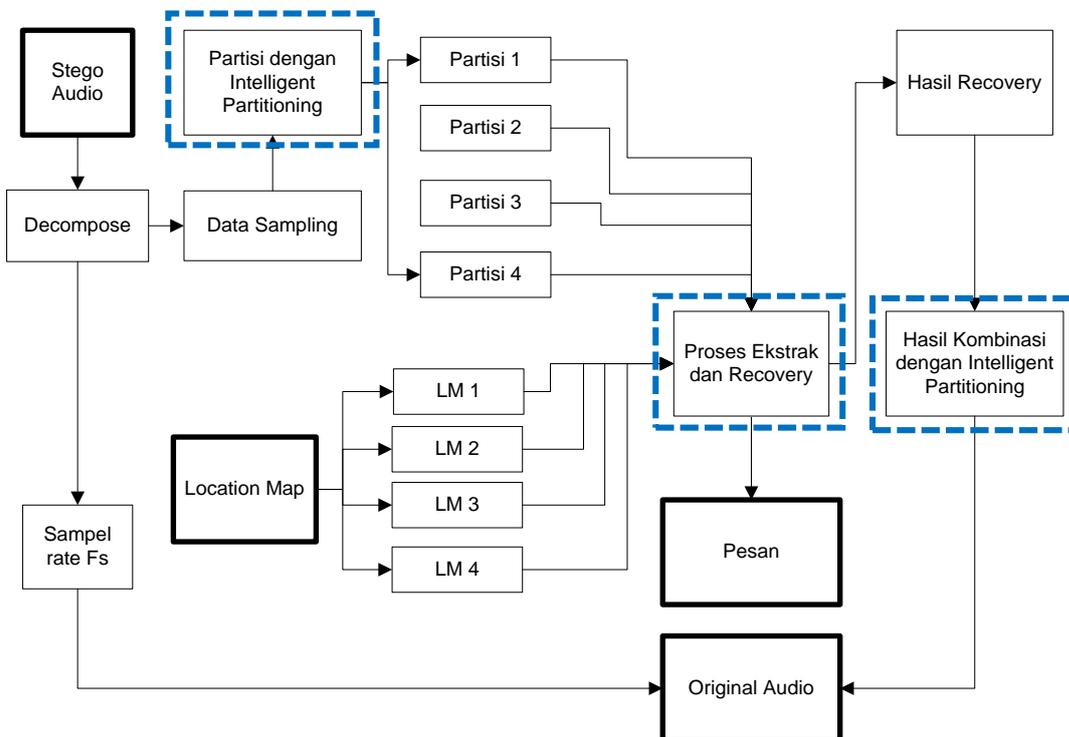
3.3.2.1 Proses *Encoding*

Sama seperti proses *encoding* sebelumnya tahap ini terdiri dari dua proses yaitu proses partisi dan penyisipan. Bagan alir tahap *encoding* dapat dilihat pada Gambar 3.6. Tahap awal yang dilakukan adalah mendecompose audio yang kemudian akan didapatkan data *sampling* dan *sampel rate Fs*. Data *sampling* selanjutnya akan diproses menjadi data dalam bentuk *integer*. Agar dapat dipartisi data *sampling* terlebih dahulu diubah ke bentuk *binary*. Untuk dapat diubah

kebinary data *sampling* yang diolah harus dalam bentuk *integer* sehingga dibutuhkan proses normalisasi dengan menggunakan persamaan (3.1).

Jika pada metode sebelumnya sampel dipartisi menjadi dua bagian yang memiliki besar 8 bit, pada metode ini sampel dibagi menjadi empat bagian yang memiliki besar 4 bit. Pesan dalam bentuk *binary* yang akan disisipkan juga dibagi empat. Kemudian pada masing- masing bagian sampel akan disisipkan dengan pesan dengan menggunakan pengembangan RDE.

Setelah proses penyisipan selesai, sistem akan menyusun kembali empat bagian sampel menjadi satu bagian utuh. Hasil penyisipan kemudian dinormalisasi kembali dengan menggunakan persamaan (3.2). Kemudian sistem menulis kembali berkas audio hasil *encoding* dalam bentuk .wav bersama dengan berkas tambahan yang berisikan *location map* dalam bentuk .txt yang nanti akan digunakan oleh sistem dalam melakukan proses *decoding*.



Gambar 3.7 Diagram alir proses *Decoding* metode partisi 2

3.3.2.2 Proses *Decoding*

Proses *decoding* juga dilakukan sama dengan pada proses *encoding*. Namun pada tahap ini proses penyisipan diganti dengan proses ekstraksi dan *recovery* pada sampel yang telah dipartisi. Proses ekstraksi akan menghasilkan pesan yang telah diambil, sedangkan proses *recovery* akan menghasilkan sampel data yang sudah dikembalikan menjadi nilai aslinya. Data hasil *recover* ini kemudian akan digabungkan kembali dengan informasi indeks awal yang kemudian akan disusun kembali menjadi Audio.

Diagram alir proses *decoding* dapat dilihat pada Gambar 3.7. Dimana pada tahap ini audio yang *didecompose* menghasilkan data *sampling* dan sampel *rate Fs*. Proses penyisipan, ekstraksi, dan *recovery* yang digunakan pada tahap ini sama dengan yang digunakan pada metode partisi satu adalah metode penyisipan pengembangan RDE.

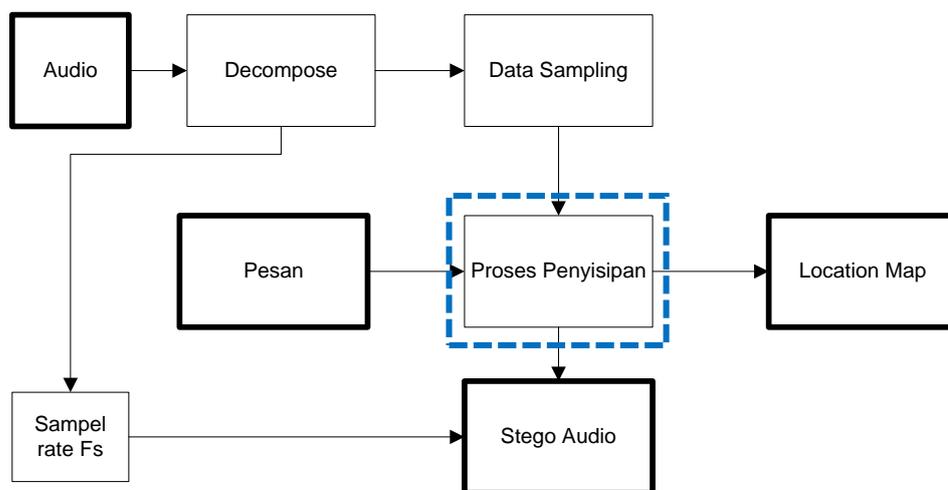
Data *sampling* akan dipartisi kembali menjadi empat menggunakan indeks partisi yang telah didapatkan dengan metode partisi. Kemudian *location map* juga akan dibagi menjadi empat. Proses ekstraksi dilakukan dengan mengambil nilai LSB dari data hasil partisi kemudian akan direkonstruksi kembali menjadi pesan. Untuk *merecovery* data *sampling*, *location map* digunakan pada masing-masing partisi sehingga nilai asli data dapat dibentuk kembali. Data hasil *recovery* kemudian akan digabungkan kembali sesuai dengan indeks awal audio sehingga didapatkan data dengan dimensi 16 bit. Data ini kemudian akan direkonstruksi kembali menjadi audio dengan tambahan informasi sampel *rate* kedalam format .wav. Dalam proses ini dihasilkan dua file yaitu audio *recover* dan pesan.

3.3.3 Proses *Encoding* Dan *Decoding* Tanpa Partisi

Pada tahap ini proses *encoding* pada metode ini tahap pertama yang dilakukan sama dengan metode sebelumnya yaitu membaca sampel pada berkas audio dan merepresentasikannya dalam bentuk *integer* dengan menggunakan persamaan (3.1). Data *integer* memiliki *range* 0 sampai 65535 ini akan langsung disisipkan dengan pesan dengan menggunakan pengembangan RDE. Proses ini

dilakukan berkali-kali pada satu berkas audio menggunakan skema *multi-layer* sehingga kapasitas penyimpanan pada berkas dapat dimaksimalkan. Setelah proses penyisipan selesai, sistem kemudian menulis kembali berkas audio hasil *encoding* bersama dengan berkas tambahan yang berisikan *location map* yang akan digunakan oleh sistem dalam melakukan proses *decoding*.

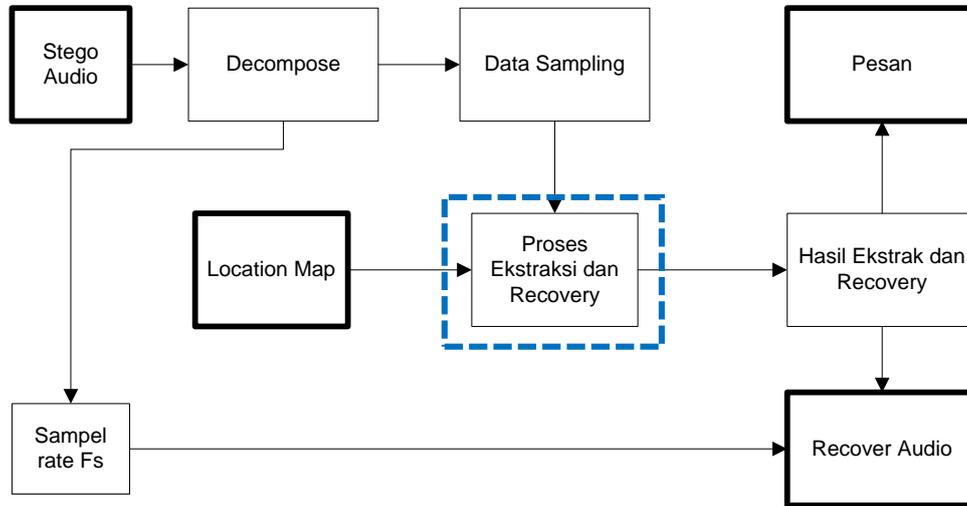
Diagram alir proses *encoding* tanpa metode partisi dapat dilihat pada Gambar 3.8. Tahap awal yang dilakukan adalah *decompose* audio yang kemudian akan didapatkan data *sampling* dan sampel rate F_s . Data *sampling* yang dihasilkan kemudian disisipkan dengan pesan. *Location map* yang dihasilkan pada proses penyisipan akan disimpan kedalam format *.txt*. Kemudian data *sampling* yang telah disisipkan akan dinormalisasi kembali menggunakan persamaan (3.2). Kemudian hasil normalisasi direkonstruksi kembali kedalam bentuk *.wav* dengan menggunakan sampel *rate*.



Gambar 3.8 Diagram alir proses *Encoding* Tanpa Partisi

Proses *decoding* juga dilakukan sama dengan pada proses *encoding*. Dimana setelah *sampling* dan mengubah data menjadi *integer* dengan menggunakan persamaan (3.1). Pesan akan diekstrak dengan mengambil nilai LSB dari data *sampling* dan stego datanya akan *recovery* kembali dengan menggunakan nilai pada *location map*. Diagram alir proses *decoding* dapat dilihat pada Gambar 3.9.

Metode penyisipan, ekstraksi, dan *recovery* yang digunakan pada tahap ini adalah metode pengembangan RDE.



Gambar 3.9 Diagram alir proses *Decoding* Tanpa Metode Partisi

3.3.4 Metode Pengembangan *Reduce Difference Expansion*

Metode penyisipan yang diusulkan merupakan pengembangan dari metode RDE (Lou et al., 2009) dimana metode ini menggunakan fungsi transformasi untuk mengurangi nilai *difference* dari dua piksel pasangan. Namun pada metode usulan ini digunakan nilai *difference* dari nilai piksel *cover* data dan nilai random sehingga dapat meningkatkan kapasitas penyisipannya dua kali metode sebelumnya.

3.3.4.1 Proses Penyisipan

Pertama, seperti pada metode penyisipan sebelumnya akan dihitung nilai *difference*nya, seperti yang ditunjukkan pada persamaan (3.6).

$$h = x - R \quad (3.6)$$

Dimana h adalah nilai *difference* dari nilai piksel x dan nilai random R , $x, R \in Z, \exists 0 \leq x \leq 255$. Setelah didapatkan nilai *difference* kemudian akan dicari nilai *reduce difference* dengan menggunakan persamaan (3.7) berikut:

$$h' = \begin{cases} |h| & , \text{jika } h = 0 \text{ atau } h = 1 \\ \left\lceil \frac{|h|-1}{2} \right\rceil & , \text{jika } h \text{ lain} \end{cases} \quad (3.7)$$

formula *reduce difference* hanya digunakan pada nilai *difference* h yang bukan 0 atau 1. Jika nilai $h = 0$ atau $h = 1$ maka nilai $h' = |h|$. Aturan pada penghitungan nilai *reduce difference* ini untuk menghindari terjadinya hasil mines pada kondisi tertentu, sehingga nilai *difference*nya masih tetap dapat dikontrol.

Pencarian nilai rata-rata l menggunakan persamaan (3.8) berikut

$$l = \begin{cases} \left\lfloor \frac{x+R}{2} \right\rfloor & \text{jika } h \leq 1 \\ \left\lceil \frac{x+R}{2} \right\rceil & \text{jika } h > 1 \end{cases} \quad (3.8)$$

Penghitungan nilai l dibagi menjadi dua kondisi dimana jika $h \leq 1$ maka nilai l yang dihasilkan akan dibulatkan kebawah sedangkan jika $h > 1$ maka nilai l digunakan merupakan hasil pembulatan keatas. Pembagian kondisi ini dibuat untuk menyeimbangi nilai l terhadap perubahan yang terjadi pada nilai h' .

Pencarian nilai hasil penyisipan ditetapkan seperti pada persamaan (3.9) berikut

$$h'' = \begin{cases} 2h' - b & \text{jika } h < 0 \\ 2h' + b & \text{jika } h \geq 0 \end{cases} \quad (3.9)$$

tanpa mempertimbangkan *overflow* dan *underflow* dimana h'' adalah hasil dari penyisipan dan b bit pesan yang disisipkan, $b \in \{0,1\}$. Dua aturan yang dibentuk pada proses ini dilakukan untuk mengontrol perubahan *difference*nya sehingga nilai hasil penyisipan yang dihasilkan tidak akan menyebabkan terjadinya masalah *overflow* dan *underflow*.

Integer transform dinyatakan seperti pada persamaan (3.10)

$$x' = \begin{cases} \begin{cases} l - \left\lfloor \frac{h''+1}{2} \right\rfloor, & \text{jika } x = 2n + 1 \\ l - \left\lfloor \frac{h''}{2} \right\rfloor, & \text{jika } x = 2n \end{cases} & \text{jika } h < 0 \\ \begin{cases} l + \left\lfloor \frac{h''-1}{2} \right\rfloor, & \text{jika } x = 2n + 1, \\ l + \left\lfloor \frac{h''}{2} \right\rfloor, & \text{jika } x = 2n, \end{cases} & \text{jika } h \geq 0 \end{cases} \quad (3.10)$$

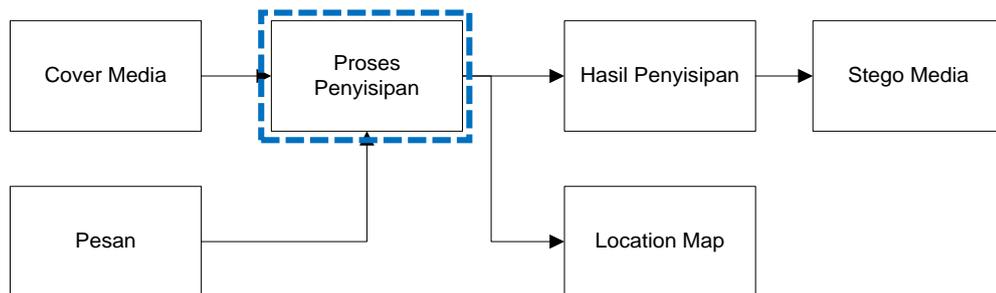
dimana nilai Integer transformnya bergantung terhadap nilai asli *cover* x , nilai vektor $n (0,1, \dots, n), n \in Z$. Hasil penyisipan ini akan menghasilkan nilai perubahan yang mendekati nilai aslinya diman jika nilai $x = 2n + 1$ atau ganjil disisipkan pesan 1 maka nilai perubahannya akan tetap sedangkan jika disisipi dengan 0 maka nilai perubahannya akan berkurang 1. Sebaliknya jika $x = 2n$ atau

genap jika disisipkan pesan 1 maka nilai perubahannya akan bertambah 1 sedangkan jika disisipi dengan 0 maka nilai perubahannya akan tetap.

Penentuan *location map* akan ditentukan berdasarkan nilai *cover* aslinya, jika nilai $x = 2n + 1$ atau ganjil maka *location map* yang disimpan adalah 1 dan jika $x = 2n$ atau genap maka *location map* yang disimpan adalah 0. Berikut penentuan *location map* berdasarkan persamaan (3.11).

$$Location Map = \begin{cases} 1, & \text{jika } x = 2n + 1 \\ 0, & \text{jika } x = 2n \end{cases} \quad (3.11)$$

Diagram alir dari proses penyisipannya dapat dilihat pada Gambar 3.10. Pada tahap ini pesan akan disisipkan pada *cover* media sesuai dengan metode penyisipan yang diusulkan. Kemudian akan dihasilkan dua *file output* yaitu stego media dan *location map*. *Location map* akan disimpan dengan format file .txt yang selanjutnya akan digunakan pada proses ekstraksi dan *recovery*.



Gambar 3.10 Diagram Alir Proses Penyisipan

3.3.4.2 Proses Ekstraksi dan *Recovery*

Ekstraksi adalah proses dimana informasi yang sebelumnya telah disisipi pada *cover* media akan diambil atau diekstrak kembali. Proses ekstraksi dilakukan dengan mengambil nilai LSB dari data *cover* media yang telah disisipi seperti pada persamaan (3.12).

$$b = LSB(x') \quad (3.12)$$

Selanjutnya adalah melakukan proses *recovery*. *Recovery* adalah proses mengembalikan stego-data ke bentuk awal atau aslinya. Dalam hal ini adalah untuk mengembalikan nilai x' menjadi x . Pada proses *recovery* dibutuhkan *location map*,

untuk *recovery* stego-data dengan benar. Dimana ukuran dari *location map* sama dengan ukuran data yang ada pada *cover* yang menunjukkan informasi penyisipan pada setiap data yang ada. Setelah mendapatkan *location map* maka proses pengembalian nilai *cover* dapat dilakukan sama dengan tahap penyisipan yaitu dengan menggunakan persamaan (3.13).

$$h = x' - R \quad (3.13)$$

Persamaan ini digunakan untuk mencari nilai *difference* dimana h adalah nilai *difference* dari nilai piksel $x' \in Z, \exists 0 \leq x \leq 255$ dan nilai random R , dimana nilai random yang dipilih pada tahap ini tidak harus sama dengan pada tahap penyisipan $R \in Z$.

Setelah didapatkan nilai *difference* kemudian akan dicari nilai *reduce difference* dengan menggunakan persamaan (3.14) berikut:

$$h' = \begin{cases} |h| & , \text{jika } h = 0 \text{ atau } h = 1 \\ \left\lfloor \frac{|h|-1}{2} \right\rfloor & , \text{jika } h \text{ lain} \end{cases} \quad (3.14)$$

Formula *reduce difference* hanya digunakan pada nilai *difference* h yang bukan 0 atau 1. Jika nilai $h = 0$ atau $h = 1$ maka nilai $h' = |h|$. Pencariann nilai rata-rata l menggunakan persamaan (3.15) berikut :

$$l = \begin{cases} \left\lfloor \frac{x'+R}{2} \right\rfloor & \text{jika } h \leq 1 \\ \left\lceil \frac{x'+R}{2} \right\rceil & \text{jika } h > 1 \end{cases} \quad (3.15)$$

Sedangkan yang membedakan proses ini dengan proses penyisipan adalah tahap menghitung nilai h'' yang menggunakan nilai LM , yang dapat dilihat pada persamaan (3.16).

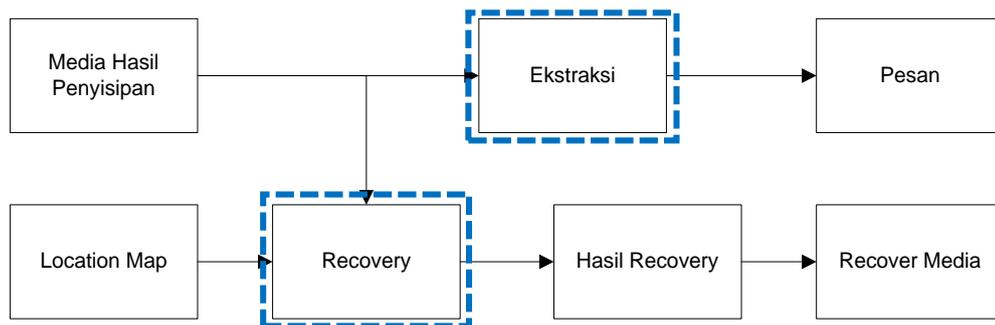
$$h'' = \begin{cases} 2h' - LM & \text{jika } h < 0 \\ 2h' + LM & \text{jika } h \geq 0 \end{cases} \quad (3.16)$$

kemudian tahap selanjutnya juga sama dengan pada proses penyisipan yaitu dengan menggunakan persamaan (3.17) untuk mendapatkan nilai *cover* x kembali. Metode ini kami sebut dengan metode *Random Reduce Difference Expansion* (RRDE).

$$x = \begin{cases} l - \left\lfloor \frac{h''+1}{2} \right\rfloor, & \text{jika } x' = 2n + 1 \\ l - \left\lfloor \frac{h''}{2} \right\rfloor, & \text{jika } x' = 2n \end{cases} \quad \text{jika } h < 0$$

$$x = \begin{cases} l + \left\lfloor \frac{h''-1}{2} \right\rfloor, & \text{jika } x' = 2n + 1, \\ l + \left\lfloor \frac{h''}{2} \right\rfloor, & \text{jika } x' = 2n, \end{cases} \quad \text{jika } h \geq 0$$
(3.17)

Diagram alir dari proses penyisipannya dapat dilihat pada Gambar 3.10. Dimana pada gambar ditunjukkan proses awal yang dilakukan adalah proses ekstraksi yaitu mengambil dan merekonstruksi pesan dengan mengambil LSB dari setiap data. Kemudian dengan mengambil informasi dari *location map* proses *recovery* dilakukan untuk mengembalikan data ke nilai aslinya. Dalam tahap ini akan dihasilkan dua file data yaitu *recover-media* dan pesan.



Gambar 3.11 Diagram Alir Proses Ekstraksi dan *Recovery*

3.3.5 Multi Layer dan Multi Level Embedding

Untuk meningkatkan kapasitas dari *cover* data diusulkan penyisipan multi *layer* dan multi level. Penyisipan multi *layer* dilakukan dengan menyisipkan sebanyak mungkin data pada sebanyak mungkin *layer* dengan yang ditentukan. Sedangkan penyisipan multi level diterapkan dengan melakukan penyisipan pada audio yang memiliki dua *channel* sinyal yang independen yaitu audio stereo. Dengan menggunakan audio kapasitas audio dapat ditingkatkan sebanyak dua kali dibanding audio yang memiliki satu *channel* sinyal atau yang sering disebut mono.

3.4 Implementasi Penelitian

Implementasi pada penelitian ini menggunakan *hardware* berupa PC dengan spesifikasi Intel Core i3 2330 (2.20 GHz) dan 2 GB DDR3 SDRAM. Sedangkan bahasa pemrograman yang digunakan adalah MATLAB 2012b. Tahapan ini digunakan untuk mengimplementasikan metode yang diusulkan ke dalam perangkat lunak. Proses implementasi ini meliputi menulis kode program, *compiling*, *debugging*, sampai dihasilkan kode program yang siap dieksekusi.

3.5 Perancangan Uji Coba dan Analisis Hasil

Sub bab ini menjelaskan mengenai skenario uji coba yang akan dilakukan pada penelitian ini. Terdapat tiga variabel yang akan diuji pada tahap ini yaitu:

Kualitas : Pengujian kualitas audio akan dilakukan dengan menggunakan metode *Signal To Noise Ratio* (SNR). SNR adalah metode yang mengukur kemiripan dari suatu data dengan data yang lain. SNR sering digunakan untuk mengukur kualitas dari berkas audio, dengan membandingkan stego audio dan *cover* audio. Semakin tinggi nilai SNR maka semakin tinggi pula kemiripan dari kedua berkas yang diuji tersebut.

Kapasitas : Pengujian kapasitas ditentukan dari kemampuan suatu *cover* untuk dapat menampung data. Dalam hal *data hiding* ukuran maksimum *payloads* digunakan untuk mengukur berapa kapasitas dari sebuah *cover*. Uji kapasitas akan dilakukan dengan membandingkan kapasitas penyisipan setiap *layer* di tambahkan hingga mencapai *layer* yang optimal.

Reversibility : *Reversibility* adalah pengujian terhadap karakteristik *reversible* dari metode yang digunakan. Hasil *recovery* media *cover* akan dibandingkan kemiripannya dengan media *cover* aslinya. Perbandingan dilakukan dengan alat bantu *diff checking tool* untuk mengetahui apakah ada perubahan pada berkas.

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang akan dilakukan. Terdapat beberapa skenario uji coba yang akan dilakukan, diantaranya yaitu:

1. Perbandingan kapasitas penyisipan dan nilai SNR pada skema penyisipan *multi-layer* dengan metode IP usulan 1. Pada skenario pengujian ini dilakukan proses *encoding* sebanyak 5 kali dimana pada setiap proses *encoding* dilakukan sebanyak 5 *layer* dengan nilai parameter besar segmen yang bervariasi pada rentang 10 sampai 50.
2. Perbandingan kapasitas penyisipan dan nilai SNR pada skema penyisipan *multi-layer* dengan metode IP usulan 2. Pada skenario pengujian ini dilakukan proses *encoding* sebanyak 5 kali dimana pada setiap proses *encoding* dilakukan sebanyak 5 *layer* dengan nilai parameter besar segmen yang bervariasi pada rentang 10 sampai 50.
3. Perbandingan kapasitas penyisipan dan nilai SNR pada skema penyisipan *multi-layer* tanpa metode IP. Pada skenario pengujian ini dilakukan proses *encoding* sebanyak 5 kali dimana pada setiap proses *encoding* dilakukan sebanyak 5 *layer*.
4. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan instrumen dan genre musik dengan metode IP usulan 1. Pada skenario pengujian ini akan dipilih beberapa genre musik yang berbeda untuk melihat hubungan antara genre musik terhadap hasil penyisipan. Genre musik yang akan diuji adalah klasik, *jazz*, *pop*, dan *rock*. Penyisipan dilakukan sebanyak satu *layer* dengan besar segmen adalah 50.
5. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan instrumen dan genre musik dengan metode IP usulan 2.
6. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan instrumen dan genre musik tanpa metode IP.
7. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan ukuran pesan. Pada skenario pengujian ini dilakukan proses *encoding* sebanyak 4 kali dimana pada setiap proses *encoding* ukuran pesan yang disisipkan bervariasi pada rentang 150k sampai 300k.
8. Uji *reversibility* dari hasil proses *decoding*. Pada pengujian ini akan dilakukan uji coba terhadap karakteristik *reversible* dari metode yang digunakan di dalam sistem *data hiding*. Pada uji coba ini akan dilakukan proses *encoding* pada

semua dataset audio. Kemudian hasil dari proses tersebut akan di *decode* kembali. Hasil *decode* akan dibandingkan kemiripannya dengan berkas asli.

9. Perbandingan kapasitas penyisipan dan nilai SNR pada audio dengan metode penyisipan lain. Pada pengujian ini akan dilakukan perbandingan hasil penyisipan dari dari setiap metode yang diusulkan dengan metode penyisipan yang sudah ada yaitu DE dan RDE. Pengujian akan pada ke duabelas audio yang ada. Banyak layer penyisipan adalah 1 dan segmen yang digunakan adalah 50.

Pelaksanaan uji coba akan dilakukan pada audio .wav yang merupakan audio musik instrument dengan durasi 3 detik, ukuran 517 *Kilobyte*, bit *rate* 1411, bit *depth* 16 dan frame *rate* 44100. Terdapat duabelas audio yang digunakan pada uji coba ini yaitu audio musik dengan bermacam instrumen dan genre. Instrumen yang ada pada audio ini adalah Selo, Gitar Akustik, Piano, dan Saxofon. Dataset audio ini sering digunakan dalam beberapa pengujian terhadap proses steganografi, watermarking, audio recognition dan berbagai audio processing lainnya. Selain itu dataset ini memiliki keragaman dan variasi musik instrumen dan genre.

Pesan yang akan digunakan adalah berbentuk .txt, dengan ukuran 2579 bytes dan jumlah karakter sebanyak 3971 karakter dan terdiri dari 5 paragraf yang bersumber dari teks *lorem ipsum* (Lorem Ipsum). Teks ini sering dipakai pengganti sebagai data sementara, rancangan aplikasi, seperti pada *template website*, dan berbagai keperluan IT lainnya. Oleh karena itu, penggunaan teks ini dianggap dapat mewakili karakteristik pesan rahasia.

3.6 Penyusunan Buku Tesis

Pada tahap ini melakukan pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tesis. Buku Tesis yang akan disusun bertujuan untuk memberikan gambaran dari pengerjaan tesis dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

BAB 4

HASIL DAN PEMBAHASAN

Bab ini membahas tentang hasil uji coba dan pembahasan metode *data hiding* yang diusulkan. Lingkungan uji coba akan dijelaskan pada sub bab 4.1. Pelaksanaan uji coba dijelaskan pada sub bab 4.2, dan yang terakhir adalah hasil dan evaluasi pada sub bab 4.3.

4.1 Lingkungan Uji Coba

Lingkungan perancangan perangkat lunak yang dibangun meliputi perangkat keras dan perangkat lunak. Spesifikasi lingkungan perancangan secara lebih jelas bisa dilihat pada Tabel 4.1 berikut ini:

Tabel 4.1 Spesifikasi Lingkungan Perancangan

Lingkungan	Spesifikasi	
Perangkat Keras	Processor	Intel Core i3 2330 M (2.20 GHz)
	RAM	2 GB
	VGA	802 MB
Perangkat Lunak	Sistem Operasi	MsWindows 7
	Tools	MATLAB R2012b

4.2 Pelaksanaan Uji Coba

Implementasi dilakukan pada beberapa metode yang diusulkan, dan beberapa metode *existing*. Metode *existing* yang diimplementasikan adalah metode penyisipan *Reduce Difference Expansion* dan *Difference Expansion* sebagai dan metode partisi *Intelligent Partitioning*.

Audio yang digunakan pada uji coba ini bersumber dari *Irmas* dataset (“*Irmas Dataset*”, 2012). Gambaran gelombang audio dari setiap dataset audio yang digunakan dapat dilihat pada Lampiran 1.1 sampai 1.12.

Hasil uji coba akan dibandingkan dengan metode sebelumnya berdasarkan nilai kapasitas dan SNRnya. Nilai kapasitasnya dihitung berdasarkan berapa bit yang dapat disisipkan pada suatu *cover* audio. SNR disajikan dalam satuan dB, semakin tinggi nilai SNRnya maka semakin tinggi pula kualitas dari stego audionya. Jika nilai SNRnya kurang dari 30dB maka dapat dikatakan bahwa stego audio tersebut memiliki kualitas yang rendah. Sedangkan jika nilai SNRnya lebih dari 40 dB maka dapat dikatakan bahwa stego audio tersebut memiliki kualitas yang tinggi .

Hasil pengujian akan ditunjukkan dengan berbagai macam skenario. Terdapat enam skenario yang akan dilakukan pada tahap ini yaitu :

1. Perbandingan kapasitas penyisipan dan nilai SNR pada skema penyisipan *multi-layer* dengan metode IP usulan 1.
2. Perbandingan kapasitas penyisipan dan nilai SNR pada skema penyisipan *multi-layer* dengan metode IP usulan 2.
3. Perbandingan kapasitas penyisipan dan nilai SNR pada skema penyisipan *multi-layer* tanpa metode IP.
4. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan instrumen dan genre musik dengan metode IP usulan 1.
5. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan instrumen dan genre musik dengan metode IP usulan 2.
6. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan instrumen dan genre musik tanpa metode IP.
7. Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan
8. Uji *reversibility* dari hasil proses *decoding*.
9. Perbandingan kapasitas penyisipan dan nilai snr pada audio dengan metode penyisipan lain.

4.3 Hasil dan Evaluasi

Ada beberapa parameter pengujian yang akan dibandingkan yaitu berdasarkan kualitas stego-audio akan dilakukan dengan menggunakan metode *Signal To Noise Ratio* (SNR). Semakin tinggi nilai SNR maka semakin tinggi pula

kemiripan dari kedua berkas yang diuji tersebut. Kemudian terhadap kapasitas yang ditentukan dari banyaknya pesan yang dapat disisipkan dalam suatu *cover*.

4.3.1 Skema Penyisipan *Multi-Layer* Dengan Metode Partisi Usulan 1

Pada skenario pengujian ini dilakukan proses *encoding* sebanyak 5 kali dimana setiap proses *encoding* dilakukan sebanyak 5 *layer* dengan nilai parameter besar segmen yang bervariasi pada rentang 10 sampai 50. Proses pengujian ini dilakukan untuk melihat pengaruh penyisipan *multi-layer* dan segmen terhadap nilai kapasitas dan kualitas stego-audio.

Audio stereo yang *disampling* dan dikuantisasi berdurasi 3 detik dengan banyak data 132,299 per *channel* atau 264,598 data per audio. Data yang *disampling* berada pada *range* -1 sampai 1 yang kemudian diubah menjadi *integer* yang berada pada *range* 0 sampai 65535. Data *integer* ini akan diproses diubah menjadi biner dengan ukuran data 16 bit yang kemudian akan diproses menggunakan metode partisi usulan. Setiap *channel* dipartisi menjadi dua sehingga dihasilkan 529,196 data berukuran 8 bit dengan *range* nilai 0 sampai 255. Data ini selanjutnya akan disisipkan dengan pesan dengan menggunakan metode RRDE. Proses penyisipan menghasilkan dua file yaitu stego-data dan *location map*. *Location map* akan disimpan dalam file berbentuk .txt. Sedangkan stego-data akan digabungkan kembali dengan metode balikan partisi usulan hingga dapat kembali disimpan dalam bentuk audio .wav.

Tabel 4.2 Menunjukkan hasil pengukuran kapasitas dan kualitas terhadap skema *Multi-layer* terhadap gabungan metode RRDE dan IP usulan dengan segmen = 10. Dari hasil uji coba dapat dilihat bahwa setiap *layer* memiliki kapasitas yang sama yaitu sebesar 529,196 bit dengan total kapasitas adalah sebesar 2,645,980 dan nilai SNR sebesar 67.3390dB dengan 5 *layer encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 294.93164 detik sedangkan waktu *Decoding* adalah 303.53554 detik.

Tabel 4.2 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 10

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	58.83451	60.105673	529,196	67.3493	529,196
	2	58.71055	60.463805	529,196	67.3277	529,196
	3	60.1034	62.69801	529,196	67.3425	529,196
	4	58.56298	59.778134	529,196	67.3345	529,196
	5	58.7202	60.489918	529,196	67.3390	529,196
Total Kapasitas						2,645,980
SNR Akhir						67.3390
Total Waktu <i>Encoding</i>						294.93164
Total Waktu <i>Decoding</i>						303.53554
Total <i>Location Map</i>						2,645,980

Berikutnya dilakukan pengujian dengan besar segmen = 20. Dari hasil uji coba dapat dilihat bahwa setiap layer memiliki kapasitas yang sama yaitu sebesar 529,196 bit dengan total kapasitas adalah sebesar 2,645,980 dan total nilai SNR sebesar 67.3390 dB dengan 5 layer *encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 130.50047 detik sedangkan waktu *Decoding* adalah 137.668156 detik. Hasil diatas dapat dilihat pada Tabel 4.3.

Uji coba selanjutnya dilakukan dengan menggunakan segmen sebesar 30. Kapasitas totalnya adalah sebesar 2,645,980 dan total nilai SNR sebesar 67.3390 dB dengan 5 layer *encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 95.6255 detik sedangkan waktu *Decoding* adalah 99.59868 detik. Hasil detail dari pengujian ini dapat dilihat pada tabel 4.4.

Uji coba pada Tabel 4.5 dilakukan dengan menggunakan segmen sebesar 40. Total kapasitas adalah sebesar 2,645,980 dan nilai SNR sebesar 67.3390 dB dengan 5 layer *encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 80.53919 detik sedangkan waktu *Decoding* adalah 88.047171 detik.

Tabel 4.3 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 20

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	26.15226	26.89251	529,196	67.3493	529,196
	2	25.71388	27.329878	529,196	67.3277	529,196
	3	25.94109	27.617413	529,196	67.3425	529,196
	4	26.98536	27.806303	529,196	67.3345	529,196
	5	25.70788	28.022052	529,196	67.3390	529,196
Total Kapasitas						2,645,980
SNR Akhir						67.3390
Total Waktu Encoding						130.50047
Total Waktu Decoding						137.668156
Total Location Map						2,645,980

Tabel 4.4 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 30

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	19.78819	20.248577	529,196	67.3493	529,196
	2	19.15544	19.905503	529,196	67.3277	529,196
	3	19.02947	19.76671	529,196	67.3425	529,196
	4	19.19344	19.898138	529,196	67.3345	529,196
	5	18.45896	19.779752	529,196	67.3390	529,196
Total Kapasitas						2,645,980
SNR Akhir						67.3390
Total Waktu Encoding						95.6255
Total Waktu Decoding						99.59868
Total Location Map						2,645,980

Tabel 4.5 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 40

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	16.43013	17.426301	529,196	67.3493	529,196
	2	15.91856	17.783412	529,196	67.3277	529,196
	3	16.17572	17.316908	529,196	67.3425	529,196
	4	15.95819	17.87785	529,196	67.3345	529,196
	5	16.05659	17.6427	529,196	67.3390	529,196
Total Kapasitas						2,645,980
SNR Akhir						67.3390
Total Waktu Encoding						80.53919
Total Waktu Decoding						88.047171
Total Location Map						2,645,980

Hasil pengujian terakhir dilakukan dengan besar segmen = 50. Kapasitas dan kualitas hasil pengujian dapat dilihat pada Tabel 4.6. Hasil SNR yang didapatkan adalah sebesar 67.3390 dB dan kapasitas totalnya adalah 2,645,980. Total waktu *encoding* dibutuhkan adalah sebesar 72.96226 detik sedangkan waktu *Decoding* adalah 79.42875 detik.

Jika dilihat dari nilai kapasitasnya metode RRDE memiliki kapasitas yang sama dengan banyak data audio setelah dipartisi. Hal ini menunjukkan bahwa metode RRDE dapat mengatasi masalah *overflow* dan *underflow* dimana semua data yang ada pada audio dapat sisipkan. Metode partisi yang digunakan juga menyebabkan kapasitas data yang sebelumnya $nx2$ untuk data stereo menjadi $nx4$ yaitu meningkat dua kali lipat dari data asli sebelum dipartisi.

Hasil uji coba juga menunjukkan bahwa besar segmen tidak berpengaruh terhadap kapasitas dan nilai SNR audio hasil penyisipan. Namun lebih berpengaruh terhadap waktu *encoding* dimana dari hasil yang didapatkan, semakin besar nilai segmen maka semakin kecil waktu *encoding* yang dihasilkan.

Tabel 4.6 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 50

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	14.99492	15.837023	529,196	67.3493	529,196
	2	14.50362	15.995022	529,196	67.3277	529,196
	3	14.46865	15.64055	529,196	67.3425	529,196
	4	14.4817	16.229802	529,196	67.3345	529,196
	5	14.51337	15.726353	529,196	67.3390	529,196
Total Kapasitas						2,645,980
SNR Akhir						67.3390
Total Waktu Encoding						72.96226
Total Waktu Decoding						79.42875
Total Location Map						2,645,980

Penggunaan skema *Multi-Layer* pada metode RRDE tidak mempengaruhi nilai kapasitas dari audio hasil penyisipan. Besar kapasitas yang tidak berubah ini dikarenakan *location map* yang digenerate tidak disisipkan pada *cover* audio sehingga tidak ada perubahan kapasitas pada setiap *layer* penyisipan. *Location map* tidak disisipkan pada *cover* audio karena redundansi dari datanya sangat rendah. Hal ini menyebabkan *location map* tidak dapat dicompres dan disisipkan pada *cover* audio.

Hasil uji coba jika dilihat terhadap nilai SNR yang dihasilkan tetap stabil pada setiap *Layer*. Hal ini dikarenakan nilai SNR lebih dipengaruhi kemiripan pesan dan LSB *cover* audio. Semakin mirip pesan dan LSB *cover* audio maka nilai SNR yang dihasilkan semakin besar. Sedangkan sebaliknya jika semakin kecil kemiripan antara pesan dan LSB *cover* audio maka semakin kecil nilai SNRnya. Akan tetapi semakin banyak *layer* yang dipakai maka *location map* yang digenerate juga akan semakin besar.

4.3.2 Skema Penyisipan *Multi-Layer* Dengan Metode Partisi Usulan 2

Pada skenario pengujian ini dilakukan proses *encoding* sebanyak 5 kali dimana setiap proses *encoding* dilakukan sebanyak 5 *layer* dengan nilai parameter besar segmen yang bervariasi pada rentang 10 sampai 50. Dari setiap pengujian akan dilihat pengaruh penyisipan *multi-layer* dan segmen terhadap nilai kapasitas dan kualitasnya.

Tabel 4.7 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 10

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	57.7346	59.7443	529,196	55.1519	1,058,392
	2	57.1874	58.1247	529,196	55.1354	1,058,392
	3	56.9141	57.9593	529,196	55.1407	1,058,392
	4	57.5384	58.7761	529,196	55.1443	1,058,392
	5	56.2617	59.0152	529,196	55.1497	1,058,392
Total Kapasitas						5,291,960
SNR Akhir						55.1497
Total Waktu Encoding						285.6362
Total Waktu Decoding						293.6196
Total Location Map						5,291,960

Audio stereo yang *disampling* dan dikuantisasi berdurasi 3 detik dengan banyak data 132,299 per *channel* atau 264,598 data per audio. Data yang *disampling* berada pada range -1 sampai 1 yang kemudian diubah menjadi *integer* yang berada pada range 0 sampai 65535. Data *integer* ini akan diproses diubah menjadi biner dengan ukuran data 16 bit yang kemudian akan diproses menggunakan metode partisi usulan. Setiap *channel* dipartisi menjadi empat sehingga dihasilkan 1,058,392 data berukuran 4 bit dengan range nilai 0 sampai 15. Data ini selanjutnya akan disisipkan dengan pesan dengan menggunakan metode RRDE. Proses penyisipan menghasilkan dua file yaitu stego-data dan *location map*. *Location map* ini akan disimpan dalam file berbentuk .txt. Sedangkan stego-data

akan digabungkan kembali dengan metode balikan partisi usulan kedua hingga dapat kembali disimpan dalam bentuk audio .wav.

Tabel 4.7 menunjukkan hasil pengukuran kapasitas dan kualitas terhadap skema *Multi-layer* terhadap gabungan metode RRDE dan IP usulan 2 dengan segmen = 10. Dari hasil uji coba dapat dilihat bahwa setiap layer memiliki kapasitas yang sama yaitu sebesar 1,058,392 bit dengan total kapasitas adalah sebesar 5,291,960 dan nilai SNR akhir sebesar 55.1497 dB dengan 5 layer *encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 285.6362 detik sedangkan waktu *decoding* adalah 293.6196 detik.

Berikutnya dilakukan pengujian dengan besar segmen = 20. Dari hasil uji coba yang ditunjukkan Tabel 4.8 dapat dilihat bahwa setiap layer memiliki kapasitas yang sama yaitu sebesar 1,058,392 bit dengan total kapasitas adalah sebesar 5,291,960 dan total nilai SNR akhir sebesar 55.1497 dB dengan 5 layer *encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 146.8727 detik sedangkan waktu *decoding* adalah 155.7446 detik.

Tabel 4.8 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 20

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	30.533	31.0494	1,058,392	55.1519	1,058,392
	2	30.1393	33.3697	1,058,392	55.1354	1,058,392
	3	29.1233	29.9295	1,058,392	55.1407	1,058,392
	4	28.9777	31.2309	1,058,392	55.1443	1,058,392
	5	28.0994	30.165	1,058,392	55.1497	1,058,392
Total Kapasitas						5,291,960
SNR Akhir						55.1497
Total Waktu Encoding						146.8727
Total Waktu Decoding						155.7446
Total Location Map						5,291,960

Uji coba selanjutnya dilakukan dengan menggunakan segmen sebesar 30. Kapasitas totalnya adalah sebesar 5,291,960 dan nilai SNR akhir sebesar 55.1497 dB dengan 5 layer *encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 113.2984 detik sedangkan waktu *decoding* adalah 123.4747 detik. Hasil diatas dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 30

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	23.1256	24.1491	1,058,392	55.1519	1,058,392
	2	22.3693	24.7666	1,058,392	55.1354	1,058,392
	3	22.4709	24.5968	1,058,392	55.1407	1,058,392
	4	22.4559	24.464	1,058,392	55.1443	1,058,392
	5	22.8767	25.4981	1,058,392	55.1497	1,058,392
Total Kapasitas						5,291,960
SNR Akhir						55.1497
Total Waktu <i>Encoding</i>						113.2984
Total Waktu <i>Decoding</i>						123.4747
Total <i>Location Map</i>						5,291,960

Uji coba pada Tabel 4.10 dilakukan dengan menggunakan segmen sebesar 40. Total kapasitas adalah sebesar 5,291,960 dan nilai SNR akhir sebesar 55.1497 dB dengan 5 layer *encoding*. Total waktu *encoding* dibutuhkan adalah sebesar 101.2612 detik sedangkan waktu *decoding* adalah 109.0202 detik.

Hasil pengujian terakhir dilakukan dengan besar segmen = 50. Kapasitas dan kualitas hasil pengujian dapat dilihat pada Tabel 4.11. Hasil SNR akhir yang didapatkan adalah sebesar 55.1497 dB dan kapasitas totalnya adalah 5,291,960. Total waktu *encoding* dibutuhkan adalah sebesar 98.4117 detik sedangkan waktu *decoding* adalah 105.3009 detik.

Tabel 4.10 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 40

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	20.3948	20.7907	1,058,392	55.1519	1,058,392
	2	20.107	22.1261	1,058,392	55.1354	1,058,392
	3	20.2876	22.0524	1,058,392	55.1407	1,058,392
	4	20.2665	22.1826	1,058,392	55.1443	1,058,392
	5	20.2053	21.8683	1,058,392	55.1497	1,058,392
Total Kapasitas						5,291,960
SNR Akhir						55.1497
Total Waktu Encoding						101.2612
Total Waktu Decoding						109.0202
Total Location Map						5,291,960

Tabel 4.11 Hasil Pengujian *Multi-Layer* dengan Besar Segmen = 50

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	20.5455	21.109	1,058,392	55.1519	1,058,392
	2	18.9958	20.8876	1,058,392	55.1354	1,058,392
	3	19.0513	20.6763	1,058,392	55.1407	1,058,392
	4	19.3656	20.8818	1,058,392	55.1443	1,058,392
	5	20.4535	21.7462	1,058,392	55.1497	1,058,392
Total Kapasitas						5,291,960
SNR Akhir						55.1497
Total Waktu Encoding						98.4117
Total Waktu Decoding						105.3009
Total Location Map						5,291,960

Jika dilihat dari nilai kapasitasnya metode RRDE dengan menggunakan metode partisi 2 memiliki kapasitas yang sama dengan banyak data audio setelah dipartisi. Metode partisi yang digunakan juga menyebabkan kapasitas data yang sebelumnya $nx2$ untuk data stereo menjadi $nx8$ yaitu meningkat empat kali lipat dari data asli sebelum dipartisi.

Hasil uji coba pada metode partisi usulan 2 ini juga menunjukkan bahwa besar segmen juga tidak berpengaruh terhadap kapasitas dan nilai SNR audio hasil penyisipan. Namun lebih berpengaruh terhadap waktu *encoding* dimana dari hasil yang didapatkan, semakin besar nilai segmen maka semakin kecil waktu *encoding* yang dihasilkan.

Penggunaan skema *Multi-Layer* pada metode RRDE tidak mempengaruhi nilai kapasitas dari audio hasil penyisipan juga dikarenakan *location map* yang tidak disisipkan pada *cover* audio. Sedangkan nilai kualitas yang dihasilkan juga tetap stabil.

4.3.3 Skema Penyisipan *Multi-Layer* Tanpa IP

Pada skenario pengujian ini dilakukan proses penyisipan *multilayer* tanpa menggunakan metode *Intelligent Partitioning*. Metode RRDE akan langsung digunakan pada audio. Pada skenario ini *encoding* dilakukan sebanyak 5 *layer*.

Audio stereo yang *disampling* dan dikuantisasi berdurasi 3 detik dengan banyak data 132,299 per *channel* atau 264,598 data per audio. Data yang *disampling* berada pada range -1 sampai 1 yang kemudian diubah menjadi *integer* yang berada pada range 0 sampai 65535. Data dengan ukuran data 16 bit ini kemudian akan diproses menggunakan metode RRDE. Proses penyisipan menghasilkan dua file yaitu stego-data dan *location map*. *Location map* akan disimpan dalam file berbentuk .txt, sedangkan stego-data disimpan dalam bentuk audio .wav kembali.

Tabel 4.8 Menunjukkan hasil pengukuran nilai kapasitas dan SNR terhadap skema *Multi-layer* terhadap metode RRDE. Dari hasil uji coba dapat dilihat bahwa setiap layer memiliki kapasitas yang sama yaitu sebesar 264,598 bit dengan total

kapasitas adalah sebesar 1,322,990 dan nilai SNR akhir sebesar 74.0799 dB dengan 5 layer *encoding*.

Tabel 4.12 Hasil Pengujian *Multi-Layer* Tanpa IP

Audio	Layer	Waktu Encode	Waktu Decode	Ukuran LM (bit)	SNR (dB)	Kapasitas (bit)
[cel] [jaz_blu]	1	1.02000	1.30885	264,598	74.0709	264,598
	2	0.93783	1.38096	264,598	74.0693	264,598
	3	1.07739	1.30037	264,598	74.0905	264,598
	4	0.92543	1.30135	264,598	74.0951	264,598
	5	0.85872	1.22358	264,598	74.0799	264,598
Total Kapasitas						1,322,990
SNR Akhir						74.0799
Total Waktu <i>Encoding</i>						4.8194
Total Waktu <i>Decoding</i>						6.5151
Total <i>Location Map</i>						1,322,990

Hasil uji coba juga menunjukkan bahwa dengan menggunakan metode RRDE semua data yang ada pada audio dapat sisipkan sehingga besar kapasitas yang dihasilkan sama dengan banyak data audio. Penggunaan skema *Multi-layer* pada metode RRDE tidak mempengaruhi nilai kapasitas dari audio hasil penyisipan, sedangkan nilai kualitas yang dihasilkan juga tetap stabil. Sehingga penyisipan *multi-layer* dapat dilakukan tanpa ada batasan *layer*, dengan tetap menghasilkan nilai kapasitas yang sama setiap *layer* dan nilai SNR yang tetap baik.

4.3.4 Berdasarkan Instrumen dan Genre Musik Dengan Metode Partisi

Usulan 1

Pada skenario pengujian ini dipilih beberapa genre musik yang berbeda untuk melihat hubungan antara instrumen dan genre musik terhadap nilai kapasitas

dan SNR dari hasil penyisipan. Instrumen yang digunakan adalah selo, gitar akustik, piano dan saksofon. Sedangkan genre musik yang akan diuji adalah klasik, *jazz blue*, dan *pop rock*.

Hasil pengujian berdasarkan genre musik dengan menggunakan metode partisi usulan 1 ditunjukkan pada Tabel 4.7. Dari tabel tersebut dapat dilihat bahwa nilai kapasitas yang dihasilkan semua genre sama yaitu sebesar 529,196 bit. Nilai terbesar SNR yang dihasilkan adalah 89.1982 dB yang dihasilkan dari audio musik *instrument* piano dengan genre musik klasik. Sedangkan nilai SNR terkecil dihasilkan pada audio musik *instrument* gitar akustik dengan genre musik *jazz blue* yaitu sebesar 46.3584 dB. Hasil pengujian berdasarkan genre musik ditunjukkan pada Tabel 4.6.

Tabel 4.13 Hasil Pengujian Dengan Metode Partisi Usulan 1

Audio	SNR(dB)	Kapasitas (bit)
[cel][cla]	78.9652	529,196
[gac][cla]	57.8342	529,196
[pia][cla]	89.1982	529,196
[sax][cla]	66.6129	529,196
[cel][pop_roc]	53.6274	529,196
[gac][pop_roc]	46.3584	529,196
[pia][pop_roc]	88.7280	529,196
[sax][pop_roc]	80.8266	529,196
[cel][jaz_blu]	67.3493	529,196
[gac][jaz_blu]	88.0401	529,196
[pia][jaz_blu]	49.9755	529,196
[sax][jaz_blu]	47.7359	529,196

Jika dilihat dari hasil uji coba kombinasi musik instrumen piano genre klasik, instrumen piano genre *jazz blue*, dan instrumen gitar akustik genre *pop rock* menghasilkan nilai SNR yang lebih baik dibandingkan dengan kombinasi

instrumen dan genre lain. Nilai SNR yang dihasilkan pada ketiga kombinasi instrumen dan genre diatas lebih dari 80 dB. Sedangkan kombinasi musik instrumen gitar akustik genre *jazz blue*, instrumen saksofon genre *pop rock*, dan instrumen piano genre *pop rock* menghasilkan nilai SNR yang rendah dibanding dengan kombinasi lain dengan nilai SNR dibawah 50 dB. Hal ini menunjukkan kombinasi musik instrumen dan genre dapat mempengaruhi kualitas stego-audio.

4.3.5 Berdasarkan Istrumen dan Genre Musik Dengan Metode Partisi

Usulan 2

Skenario pengujian ini menggunakan audio dengan instrumen dan genre musik sama dengan pengujian sebelumnya. Pengujian ini bertujuan untuk melihat pengaruh instrumen dan genre musik terhadap nilai kapasitas dan SNR dari hasil penyisipan. Instrumen yang digunakan adalah selo, gitar akustik, piano dan saksofon. Sedangkan genre musik yang akan diuji adalah klasik, *jazz blue* dan *pop rock*.

Tabel 4.7 menunjukkan hasil pengujian berdasarkan instrumen dan genre musik dengan menggunakan metode partisi usulan 2. Dari tabel tersebut dapat dilihat bahwa nilai kapasitas yang dihasilkan semua genre sama yaitu sebesar 1,058,392 bit. Nilai terbesar SNR yang dihasilkan adalah 77.0261 dB yang dihasilkan dari audio musik *instrument* piano dengan genre musik klasik. Sedangkan nilai SNR terkecil dihasilkan pada audio musik *instrument* gitar saksofon dengan genre musik *pop rock* yaitu sebesar 35.5664 dB. Hasil pengujian berdasarkan genre musik ditunjukkan pada Tabel 4.8.

Jika dilihat dari hasil uji coba kombinasi musik instrumen piano genre klasik, instrumen piano genre *jazz blue*, dan instrumengitar akustik genre *pop rock* menghasilkan nilai SNR yang lebih baik dibandingkan dengan kombinasi instrumen dan genre lain. Nilai SNR yang dihasilkan pada ketiga kombinasi instrumen dan genre diatas lebih dari 70 dB. Sedangkan kombinasi musik instrumen gitar akustik genre *jazz blue*, instrumen saksofon genre *pop rock*, dan instrumen piano genre *pop rock* menghasilkan nilai SNR yang rendah dibanding dengan

kombinasi lain dengan nilai SNR dibawah 40 dB. Hal ini menunjukkan kombinasi musik instrumen dan genre dapat mempengaruhi kualitas stego-audio.

Tabel 4.14 Hasil Pengujian Dengan Metode Partisi Usulan 2

Audio	SNR(dB)	Kapasitas
[cel][cla]	66.7558	1,058,392
[gac][cla]	45.6538	1,058,392
[pia][cla]	77.0261	1,058,392
[sax][cla]	54.4333	1,058,392
[cel][pop_roc]	44.452	1,058,392
[gac][pop_roc]	36.7517	1,058,392
[pia][pop_roc]	76.5232	1,058,392
[sax][pop_roc]	68.6258	1,058,392
[cel][jaz_blu]	55.1519	1,058,392
[gac][jaz_blu]	75.8467	1,058,392
[pia][jaz_blu]	37.7948	1,058,392
[sax][jaz_blu]	35.5664	1,058,392

4.3.6 Berdasarkan Instrumen dan Genre Musik Tanpa IP

Hasil uji coba pada tahap ini sama seperti uji genre musik sebelumnya yaitu membandingkan kapasitas penyisipan dan nilai SNR, namun pada tahap ini akan dilakukan tanpa menggunakan metode Intelligent Partitioning. Pada skenario ini digunakan beberapa genre musik yang berbeda untuk melihat hubungan antara genre musik terhadap hasil penyisipan. Genre musik yang akan diuji adalah klasik, *jazz blue* dan *pop rock*.

Perbandingan kapasitas penyisipan dan nilai SNR pada audio berdasarkan genre musik. Pada skenario pengujian ini dipilih beberapa genre musik yang berbeda untuk melihat hubungan antara genre musik terhadap hasil penyisipan. Genre musik yang akan diuji adalah klasik, *jazz blue*, dan *pop rock*.

Hasil pengujian menunjukkan bahwa nilai kapasitas yang dihasilkan semua genre sama yaitu sebesar 264598 bit. Nilai terbesar SNR yang dihasilkan adalah 95.9539 dB yang dihasilkan dari audio musik *instrument* piano dengan genre musik klasik. Sedangkan nilai SNR terkecil dihasilkan pada audio musik *instrument* gitar akustik dengan genre musik *jazz blue* yaitu sebesar 53.1178 dB.

Tabel 4.15 Hasil Pengujian Berdasarkan Genre Musik Tanpa IP

Audio	SNR(dB)	Kapasitas
[cel][cla]	85.6985	264,598
[gac][cla]	64.5843	264,598
[pia][cla]	95.9539	264,598
[sax][cla]	73.3562	264,598
[cel][jaz_blu]	60.3697	264,598
[gac][jaz_blu]	53.1178	264,598
[pia][jaz_blu]	95.4927	264,598
[sax][jaz_blu]	87.5642	264,598
[cel][pop_roc]	74.0595	264,598
[gac][pop_roc]	94.7986	264,598
[pia][pop_roc]	56.7411	264,598
[sax][pop_roc]	54.4926	264,598

Jika dilihat dari hasil uji coba diatas sama seperti pengujian dengan menggunakan metode partisi, kombinasi musik instrumen piano genre klasik, instrumen piano genre *jazz blue*, dan instrumen gitar akustik genre *pop rock* menghasilkan nilai SNR yang lebih baik dibandingkan dengan kombinasi instrumen dan genre lain. Nilai SNR yang dihasilkan pada ketiga kombinasi instrumen dan genre diatas lebih dari 90 dB. Sedangkan kombinasi musik instrumen gitar akustik genre *jazz blue*, instrumen saksofon genre *pop rock*, dan instrumen piano genre *pop rock* menghasilkan nilai SNR yang rendah dibanding dengan

kombinasi lain dengan nilai SNR dibawah 60 dB. Hal ini menunjukkan kombinasi musik instrumen dan genre dapat mempengaruhi kualitas stego-audio.

Tabel 4.16 Perbedaan Hasil Pengujian Dengan dan Tanpa Metode Partisi 1

Audio	SNR(dB) S3-S1	Kapasitas (bit) K3-K1
[cel][cla]	6.7333	264,598
[gac][cla]	6.7501	264,598
[pia][cla]	6.7557	264,598
[sax][cla]	6.7433	264,598
[cel][jaz_blu]	6.7423	264,598
[gac][jaz_blu]	6.7594	264,598
[pia][jaz_blu]	6.7647	264,598
[sax][jaz_blu]	6.7376	264,598
[cel][pop_roc]	6.7102	264,598
[gac][pop_roc]	6.7585	264,598
[pia][pop_roc]	6.7656	264,598
[sax][pop_roc]	6.7567	264,598

Tabel 4.16 menunjukkan perbedaan hasil pengujian berdasarkan genre musik dengan dan tanpa metode partisi. S1, S2 dan S3 menunjukkan nilai SNR skenario uji coba dengan IP usulan 1, IP usulan 2 dan tanpa IP. Sedangkan K1, K2, dan K3 adalah untuk kapasitas dari skenario uji coba dengan IP usulan 1, IP usulan 2 dan tanpa IP. S3-S1 rumus perhitungan untuk mencari perbedaan nilai SNR dan PSNR dari skenario yang menggunakan IP usulan 1 dan tanpa IP. K3-K1 adalah perhitungan untuk mencari perbedaan nilai kapasitas dari skenario yang menggunakan IP usulan 1 dan tanpa IP. Tabel menunjukkan bahwa dengan menggunakan partisi nilai SNR yang dihasilkan turun dibandingkan dengan pengujian tanpa metode partisi. Dimana rata-rata penurunan nilai kualitasnya adalah sebesar 6.74811 dB. Sedangkan nilai kapasitas yang dihasilkan jika menggunakan metode partisi meningkat dua kali lipat dibanding dengan pengujian tanpa metode partisi.

S3-S2 rumus perhitungan untuk mencari perbedaan nilai SNR dan PSNR dari skenario yang menggunakan IP usulan 2 dan tanpa IP. K3-K2 adalah perhitungan untuk mencari perbedaan nilai kapasitas dari skenario yang menggunakan IP usulan 2 dan tanpa IP. Tabel 4.17 menunjukkan perbedaan hasil pengujian berdasarkan genre musik dengan dan tanpa metode partisi. Tabel menunjukkan bahwa dengan menggunakan partisi nilai SNR yang dihasilkan turun dibandingkan dengan pengujian tanpa metode partisi. Rata-rata penurunan nilai SNR adalah sebesar 18.4706 dB. Sedangkan nilai kapasitas yang dihasilkan jika menggunakan metode partisi meningkat tiga kali lipat dibanding dengan pengujian tanpa metode partisi.

Tabel 4.17 Perbedaan Hasil Pengujian Dengan dan Tanpa Metode Partisi 2

Audio	SNR(dB) S3-S2	Kapasitas (bit) K3-K2
[cel][cla]	18.9427	793,794
[gac][cla]	18.9305	793,794
[pia][cla]	18.9278	793,794
[sax][cla]	18.9229	793,794
[cel][jaz_blu]	15.9177	793,794
[gac][jaz_blu]	16.3661	793,794
[pia][jaz_blu]	18.9695	793,794
[sax][jaz_blu]	18.9384	793,794
[cel][pop_roc]	18.9076	793,794
[gac][pop_roc]	18.9519	793,794
[pia][pop_roc]	18.9463	793,794
[sax][pop_roc]	18.9262	793,794

S1-S2 adalah perhitungan untuk mencari perbedaan nilai SNR dan PSNR dari skenario yang menggunakan IP usulan 2 dan tanpa IP. K1-K2 adalah perhitungan untuk mencari perbedaan nilai kapasitas dari skenario yang menggunakan IP usulan 1 dan 2. Tabel 4.18 menunjukkan perbedaan hasil pengujian berdasarkan genre musik dengan dan tanpa metode partisi. Tabel

menunjukkan bahwa dengan menggunakan partisi nilai SNR yang dihasilkan turun dibandingkan dengan pengujian tanpa metode partisi. Rata-rata penurunan nilai SNR adalah sebesar 11.7225 dB. Sedangkan nilai kapasitas yang dihasilkan jika menggunakan metode partisi 2 meningkat dua kali lipat dibanding dengan pengujian dengan metode partisi 1.

Tabel 4.18 Perbedaan Hasil Pengujian dengan Metode Partisi 1 dan 2

Audio	SNR(dB) S1-S2	Kapasitas (bit) K1-K2
[cel][cla]	18.9427	529,196
[gac][cla]	18.9305	529,196
[pia][cla]	18.9278	529,196
[sax][cla]	18.9229	529,196
[cel][jaz_blu]	15.9177	529,196
[gac][jaz_blu]	16.3661	529,196
[pia][jaz_blu]	18.9695	529,196
[sax][jaz_blu]	18.9384	529,196
[cel][pop_roc]	18.9076	529,196
[gac][pop_roc]	18.9519	529,196
[pia][pop_roc]	18.9463	529,196
[sax][pop_roc]	18.9262	529,196

4.3.7 Uji Coba Berdasarkan Ukuran Pesan

Pada skenario pengujian ini dilakukan proses *encoding* sebanyak 4 kali dimana pada setiap proses *encoding* ukuran pesan yang disisipkan bervariasi pada rentang 150k sampai 300k. Proses pengujian ini dilakukan untuk melihat pengaruh ukuran pesan yang disisipkan terhadap nilai kapasitas dan kualitas stego-audio. Uji coba dilakukan pada metode partisi usulan 1 dan 2 dimana ukuran segmen yang dipakai adalah sebesar 50 dengan banyak *layer* penyisipan satu.

Tabel 4.19 Hasil SNR Berdasarkan Ukuran Pesan Metode Partisi 1

Data/ Ukuran Pesan	SNR (dB)			
	150k	200k	250k	300k
[cel][cla]	78.2314	79.8972	79.7803	79.5009
[gac][cla]	59.6425	59.4686	58.7812	58.4663
[pia][cla]	90.5269	90.6837	90.1211	89.8077
[sax][cla]	69.1337	68.5019	67.6044	67.2757
[cel][jaz_blu]	69.7189	69.1712	68.3162	67.9879
[gac][jaz_blu]	89.9585	89.7281	89.019	88.6956
[pia][jaz_blu]	52.0689	51.7037	50.9418	50.6186
[sax][jaz_blu]	48.0261	48.9066	48.5956	48.3035
[cel][pop_roc]	54.6992	55.0222	54.5209	54.2179
[gac][pop_roc]	44.7744	47.1022	47.1338	46.861
[pia][pop_roc]	91.7478	90.8126	89.7487	89.4072
[sax][pop_roc]	84.4282	83.1213	81.8693	81.5162

Tabel 4.20 Hasil SNR Berdasarkan Ukuran Pesan Metode Partisi 2

Data/ Ukuran Pesan	SNR (dB)			
	150k	200k	250k	300k
[cel][cla]	72.9617	71.3538	70.5933	70.2875
[gac][cla]	52.0237	50.4247	49.6672	49.3602
[pia][cla]	83.4367	81.8379	81.0806	80.7689
[sax][cla]	60.5946	58.9889	58.2332	57.9231
[cel][jaz_blu]	61.3522	59.752	58.996	58.6858
[gac][jaz_blu]	82.2269	80.6194	79.8636	79.5512
[pia][jaz_blu]	44.1083	42.4998	41.744	41.4348
[sax][jaz_blu]	41.9364	40.3337	39.5805	39.2723
[cel][pop_roc]	82.3239	80.7251	79.9762	79.6672
[gac][pop_roc]	40.1506	38.5502	37.7956	37.4847
[pia][pop_roc]	82.3239	80.7251	79.9762	79.6672
[sax][pop_roc]	72.925	71.3264	70.5703	70.2668

Tabel 4.19 menunjukkan hasil pengujian terhadap kualitas stego audio pada metode usulan 1. Hasil uji coba menunjukkan pada sebagian besar audio semakin besar pesan yang disisipkan maka semakin kecil nilai SNR yang dihasilkan. Sedangkan kapasitas yang dihasilkan tetap untuk metode partisi 1 yaitu 529,196 bit.

Hasil pengujian berdasarkan ukuran pesan terhadap kualitas stego audio pada metode usulan 2 dapat dilihat pada Tabel 4.20. Dimana dari tabel dapat ditunjukkan bahwa semakin besar pesan yang disisipkan maka semakin kecil juga nilai SNR yang dihasilkan. Sedangkan kapasitas yang dihasilkan juga tetap untuk metode partisi 2 adalah sebesar 1,058,392. Hasil uji kapasitas dari metode partisi 1 dan 2 dapat dilihat pada Tabel 4.21.

Tabel 4.21 Hasil Kapasitas Berdasarkan Ukuran Pesan Metode Partisi 1 dan 2

Data/ Metode	Kapasitas (bit)	
	IP 1	IP 2
[cel][cla]	529,196	1,058,392
[gac][cla]	529,196	1,058,392
[pia][cla]	529,196	1,058,392
[sax][cla]	529,196	1,058,392
[cel][jaz_blu]	529,196	1,058,392
[gac][jaz_blu]	529,196	1,058,392
[pia][jaz_blu]	529,196	1,058,392
[sax][jaz_blu]	529,196	1,058,392
[cel][pop_roc]	529,196	1,058,392
[gac][pop_roc]	529,196	1,058,392
[pia][pop_roc]	529,196	1,058,392
[sax][pop_roc]	529,196	1,058,392

4.3.8 Uji *Reversibility* Dari Hasil Proses *Decoding*.

Pada pengujian ini akan dilakukan uji coba terhadap karakteristik *reversible* dari metode yang digunakan di dalam sistem *data hiding*. Pada uji coba ini akan

dilakukan proses *encoding* pada setiap genre musik yang ada. Kemudian hasil dari proses akan di *decode* kembali kemudian hasil *decode* akan dibandingkan kemiripannya dengan berkas asli. Perbandingan dilakukan dengan menghitung nilai SNRnya. Jika Nilai SNR yang dihasilkan adalah infinitif maka hasil *encoding* dan *decoding* sama.

Uji coba pertama dilakukan pada berkas audio dengan ukuran 517 Kilobyte berdurasi 3 detik. Parameter yang digunakan adalah besar segmen 50. Berkas audio dan berkas *payload* hasil *Decoding* masing-masing dibandingkan nilai SNRnya untuk mengetahui perbedaan pada berkas. Tabel 5.10 menunjukkan kemiripan antara berkas hasil *decode* dan berkas asli.

Tabel 4.22 Hasil Pengujian Reversibility

Audio	Reversibility Audio	Reversibility Pesan
[cel][cla]	100%	100%
[gac][cla]	100%	100%
[pia][cla]	100%	100%
[sax][cla]	100%	100%
[cel][jaz_blu]	100%	100%
[gac][jaz_blu]	100%	100%
[pia][jaz_blu]	100%	100%
[sax][jaz_blu]	100%	100%
[cel][pop_roc]	100%	100%
[gac][pop_roc]	100%	100%
[pia][pop_roc]	100%	100%
[sax][pop_roc]	100%	100%

Berdasarkan Tabel 4.22 diatas dapat dilihat bahwa semua hasil uji coba yang dihasilkan bahwa data *encoding* dan *decoding* pada berkas dan pesan memiliki kemiripan 100%. Hal ini karena nilai SNR yang dihasilkan infinitif sehingga data audio asli dan data *recover* sama atau identik.

4.3.9 Perbandingan Dengan Metode Penyisipan Lain

Sub bab ini membahas mengenai perbandingan hasil *encoding* dari metode yang diusulkan dengan metode yang telah ada. Hasil yang dibandingkan adalah hasil dari metode data hiding pada audio yaitu metode penyisipan *Difference Expansion* dan *Reduce Difference Expansion*. Selain itu membandingkan dengan metode partisi yang sudah ada yaitu *Intelligent Partitioning*.

Tabel 4.23 Hasil Perbandingan nilai SNR dengan Metode *Existing*

Data/ SNR(dB)	DE		RDE		RRDE	
	IP	IP Usulan 1	IP	IP Usulan 1	IP	IP Usulan 1
[cel][cla]	10.8427	11.4088	15.0739	15.6013	78.9652	78.9652
[pia][cla]	24.5139	10.2158	29.1663	14.7329	89.1982	89.1982
[pia][pop_roc]	13.5957	15.8059	18.4019	20.1511	88.728	88.728
[sax][pop_roc]	6.0653	11.0802	11.0792	14.7454	69.4979	80.8266
[gac][jaz_blu]	14.2805	11.9677	19.2775	16.0304	76.7488	88.0401

Hasil pengujian akan dibandingkan berdasarkan kapasitas dalam satuan bit, nilai SNR dari metode. Parameter yang digunakan adalah panjang segmen sebesar 50 dengan nilai treshold maksimal untuk DE dan RDE. Hasil uji coba hanya menampilkan lima macam audio. Pemilihan audio ini didasarkan pada nilai SNR yang dihasilkan, dimana audio yang dipakai hanya yang memiliki nilai SNR diatas 5 dB. Hasil dari perbandingan kualitas setiap metode dapat dilihat pada Tabel 4.23. Hasil eksperimen yang didapatkan menunjukkan jika dilihat terhadap metode penyisipannya nilai kualitas metode usulan lebih baik dari metode *existing*. Dimana untuk kedua metode IP rata-rata nilai SNR yang dihasilkan lebih dari 70 dB.

Hasil uji coba jika dilihat terhadap metode partisi dengan metode penyisipan lain yaitu DE dan RDE, maka metode IP memiliki nilai SNR lebih baik dibandingkan dengan metode IP usulan 1. Dimana nilai SNR terbaik dihasilkan pada audio dengan instrumen piano dan genre klasik. Pada audio tertentu metode IP usulan 1 memiliki nilai SNR lebih baik dari IP yaitu pada audio dengan

instrumen selo genre klasik, instrumen saksofon genre *pop rock*, dan instrumen piano genre *pop rock* walaupun nilai SNR yang dihasilkan dibawah 20 dB.

Tabel 4.24 menunjukkan hasil perbandingan kapasitas dengan metode lain, dari hasil yang ditunjukkan pada tabel menunjukkan metode RRDE dengan kedua metode partisi dapat menghasilkan nilai kapasitas yang lebih baik dibanding metode DE dan RDE. Sedangkan jika dilihat pengaruh metode IP usulan 1 terhadap metode DE dan RDE, metode IP usulan 1 pada penelitian ini dapat meningkatkan nilai kapasitas dari berkas audio dimana pada audio dengan instrumen piano genre *pop rock*, instrumen saksofon genre *pop rock* dan instrumen gitar jenre *jazz blue* memiliki nilai kapasitas yang lebih baik dibanding metode IP.

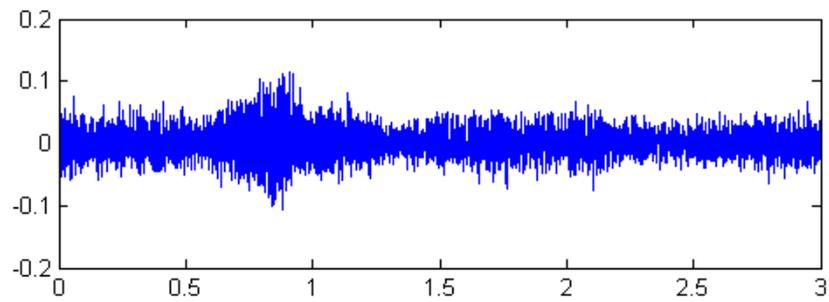
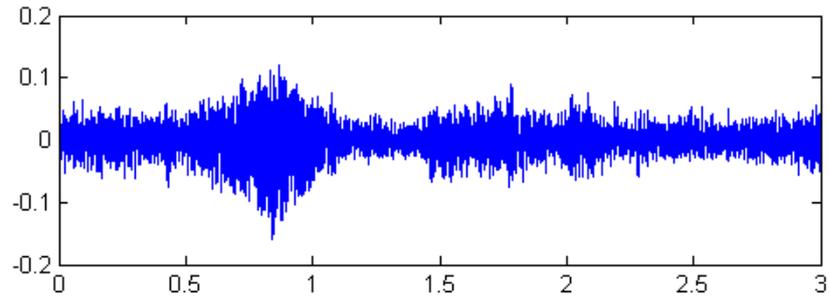
Tabel 4.24 Hasil Perbandingan Kapasitas dengan Metode *Existing*

Data/ Kapasitas (bit)	DE		RDE		RRDE	
	IP	IP Usulan 1	IP	IP Usulan 1	IP	IP Usulan 1
[cel][cla]	263,314	262,491	263,314	262,491	529,196	529,196
[pia][cla]	263,801	263,200	263,801	263,200	529,196	529,196
[pia][pop_roc]	263,050	264,025	263,050	264,025	529,196	529,196
[sax][pop_roc]	262,717	263,640	262,717	263,640	529,196	529,196
[gac][jaz_blu]	262,609	263,755	262,609	263,755	529,196	529,196

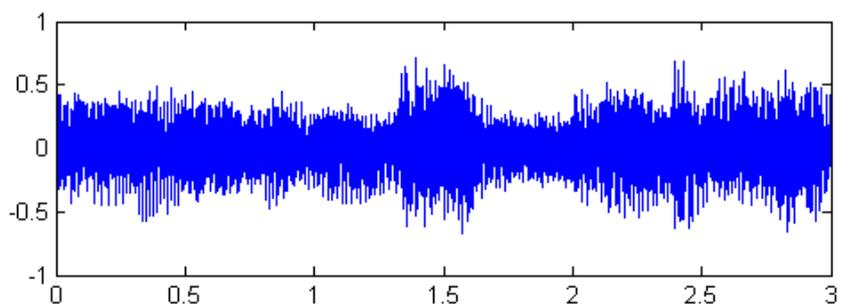
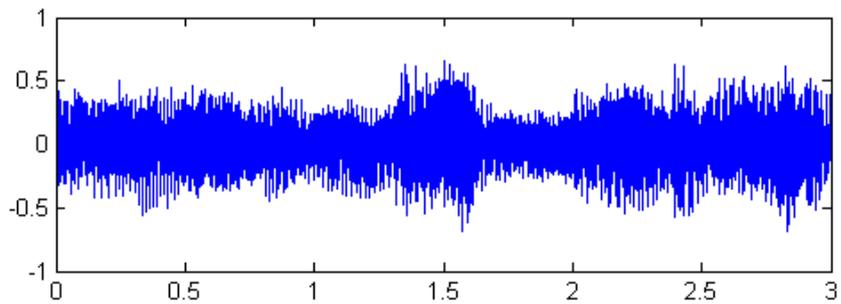
Jika dilihat hasil uji coba secara menyeluruh metode penyisipan yang disusulkan pada penelitian ini yaitu RRDE memiliki nilai SNR dan kapasitas dibanding metode penyisipan lain. Akan tetapi memiliki kelemahan pada *location map*nya jika dibandingkan dengan metode penyisipan lain. Metode RRDE memiliki ukuran *location map* yang besar sama dengan ukuran asli data akan tetapi tidak dapat disisipkan kecover audio karena redundansinya yang rendah. Jika dilihat pada metode DE, *location map* yang digenerate setengah dari ukuran asli data dengan redundansi yang tinggi sehingga seluruh *location map* dapat di *compress* dan disisipkan pada cover audio. Sedangkan metode RDE memiliki *location map* sama dengan ukuran asli data dimana setengah dari *location map* dapat di *compress* dan disisipkan pada cover audio. Untuk metode partisi, secara keseluruhan metode IP

yang diusulkan memiliki performa yang lebih baik dibanding metode IP asli, akan tetapi pada kasus-kasus tertentu metode IP asli dapat mengungguli metode IP usulan. Metode usulan ini juga tidak perlu menyimpan indeks partisi sebagai informasi tambahan audio saat akan *recover* pesan dan *cover* audio kembali seperti hal yang harus dilakukan pada metode IP.

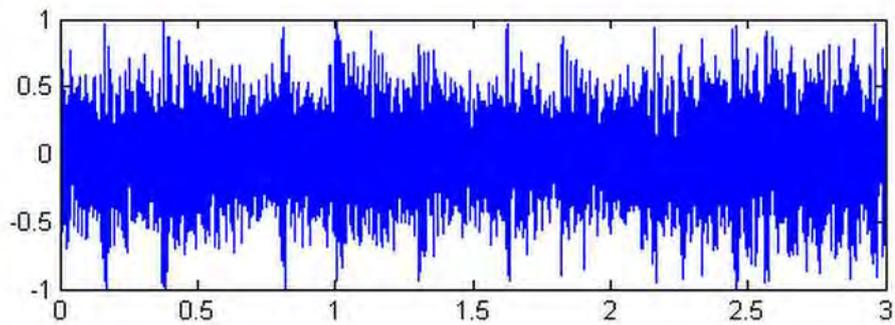
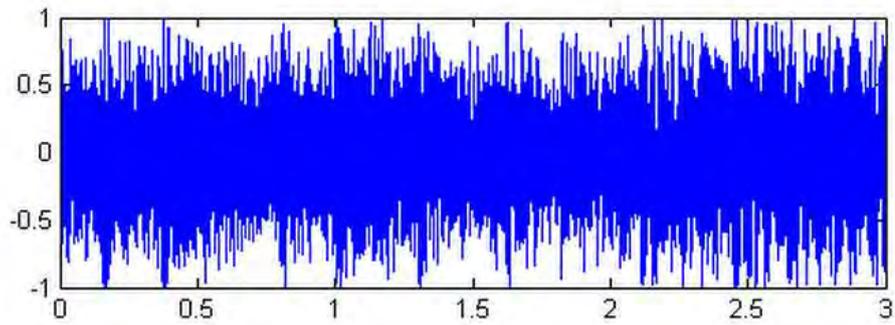
LAMPIRAN



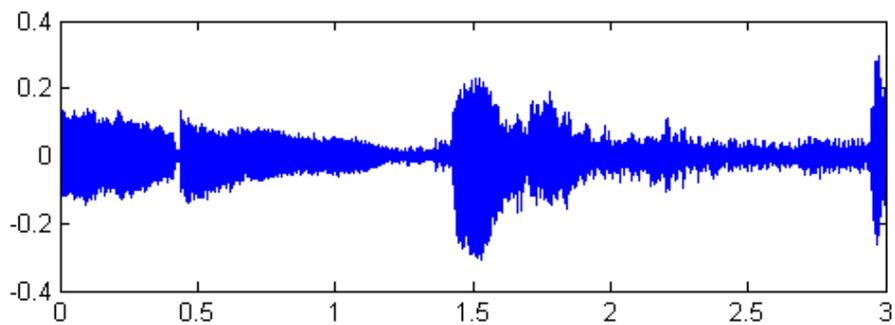
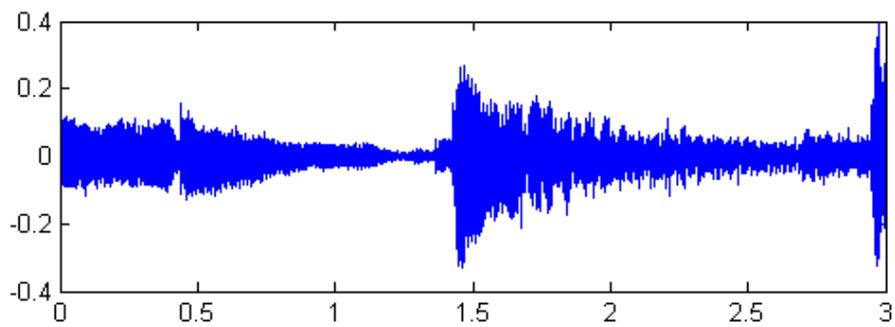
Lampiran 1.1 Audio Instrumen Selo Genre Klasik [cel] [cla]



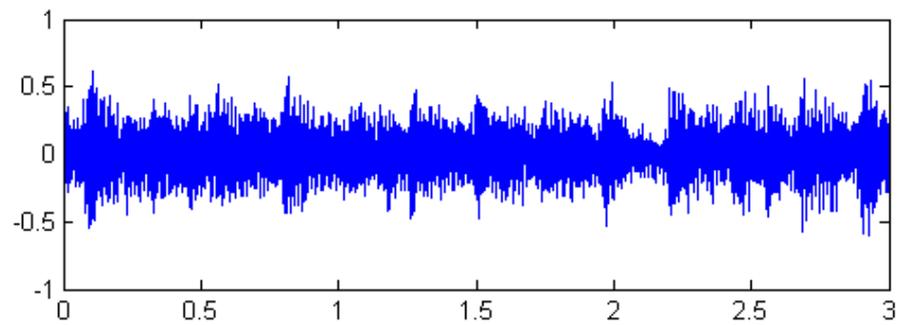
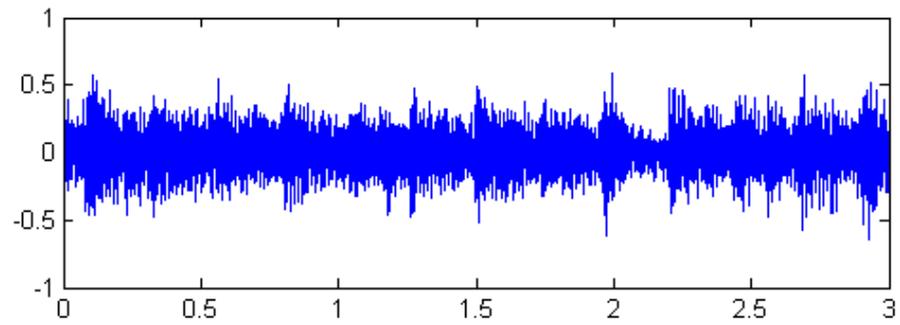
Lampiran 1.2 Audio Instrumen Selo Genre *Jazz Blue* [cel] [jaz_blu]



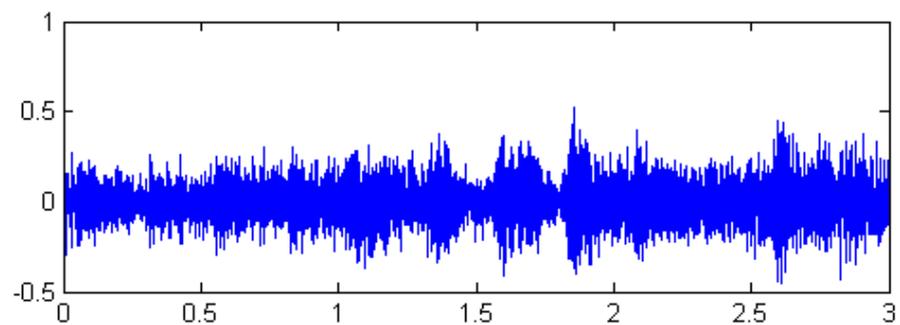
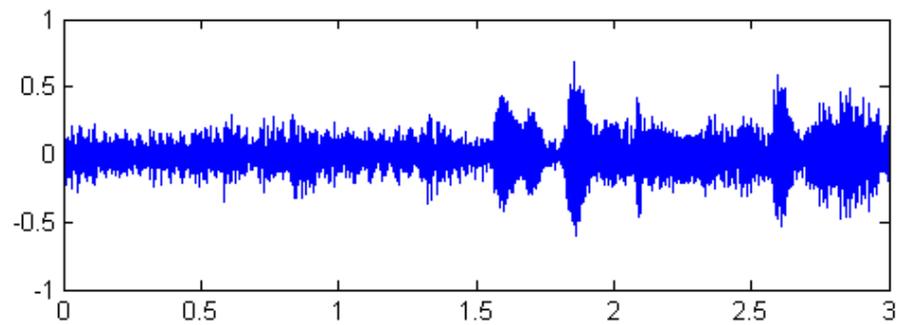
Lampiran 1.3 Audio Instrumen Selo Genre *Pop Rock* [cel] [pop_roc]



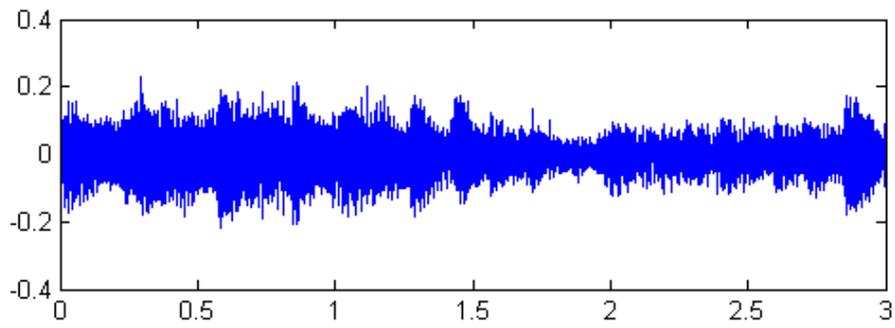
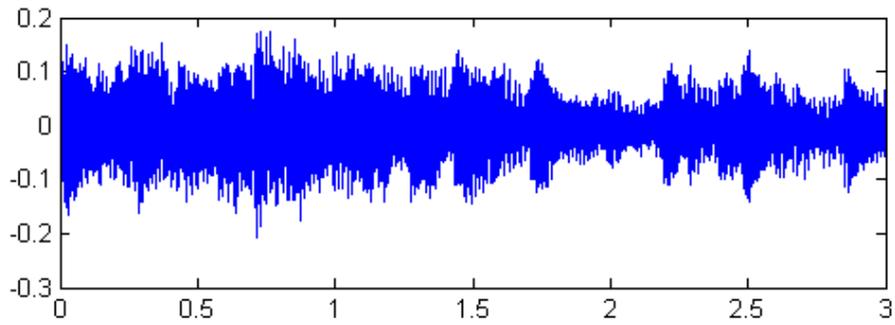
Lampiran 1.4 Audio Instrumen Gitar Genre Klasik [gac] [cla]



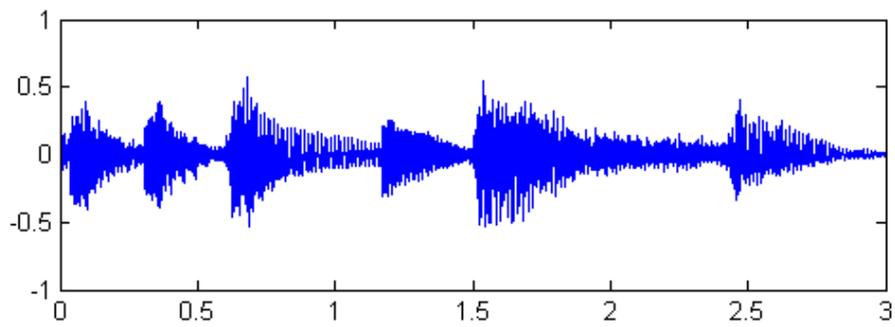
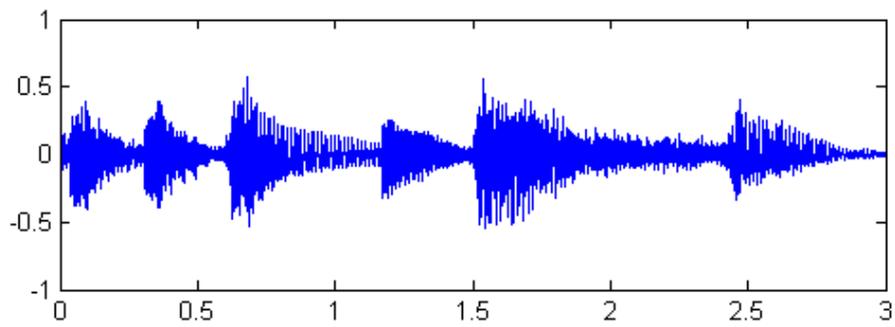
Lampiran 1.5 Audio Instrumen Gitar Genre *Jazz Blue* [gac] [jaz_blu]



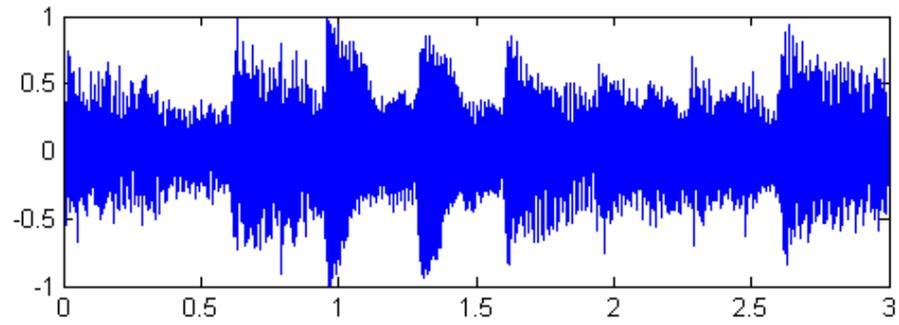
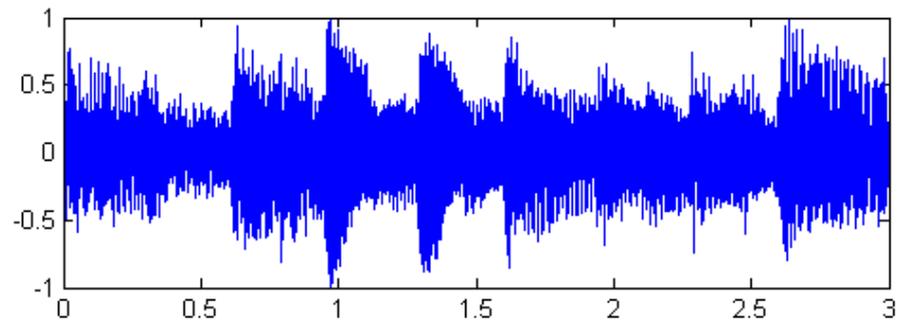
Lampiran 1.6 Audio Instrumen Gitar Genre *Pop Rock* [gac] [pop_roc]



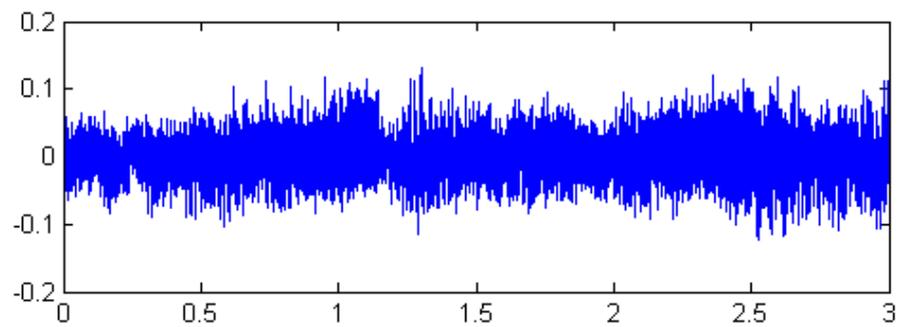
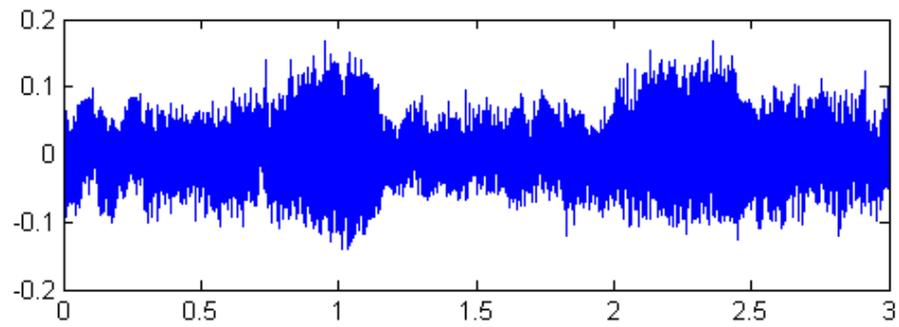
Lampiran 1.7 Audio Instrumen Piano Genre Klasik [pia] [cla]



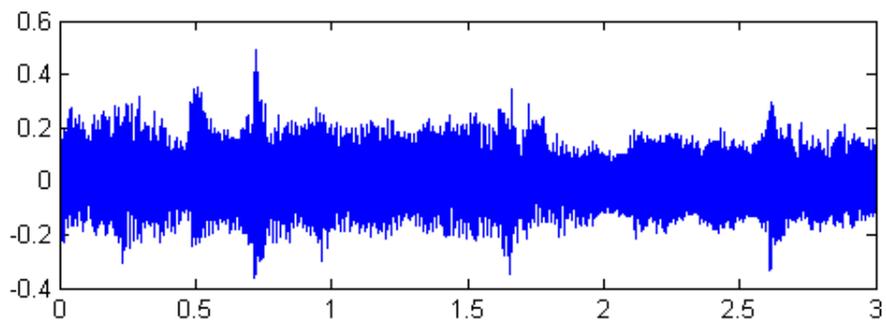
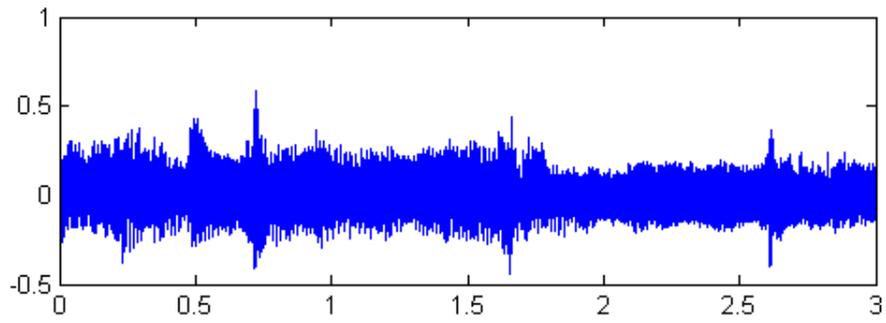
Lampiran 1.8 Audio Instrumen Piano Genre *Jazz Blue* [pia][jaz_blu]



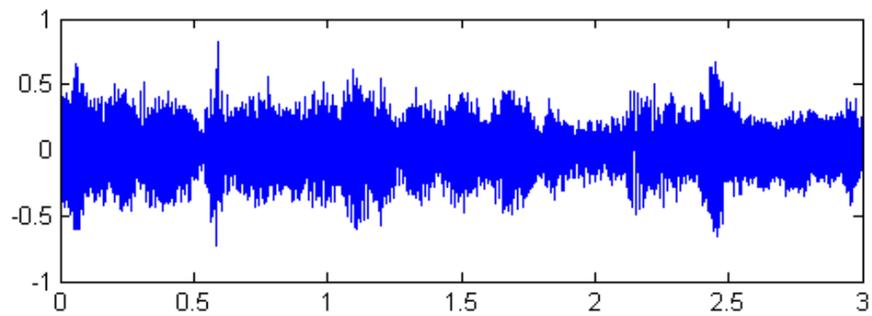
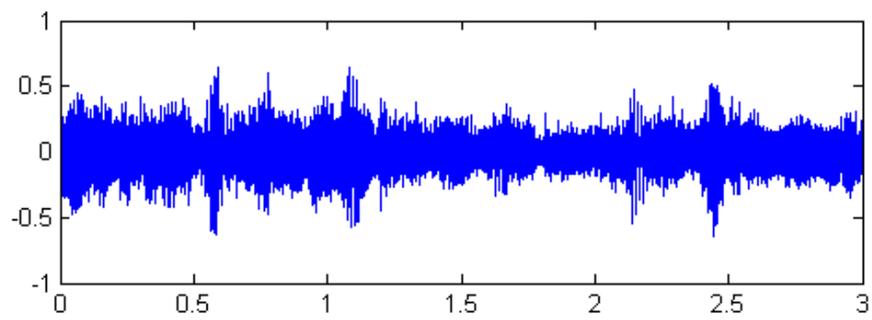
Lampiran 1.9 Audio Instrumen Piano Genre *Pop Rock* [pia][pop_roc]



Lampiran 1.10 Audio Instrumen Saksofon Genre Klasik [sax][cla]



Lampiran 1.11 Audio Instrumen Saksofon Genre *Jazz Blue* [sax] [jaz_blu]



Lampiran 1.12 Audio Instrumen Saksofon Genre *Pop Rock* [sax] [pop_roc]

BAB 5

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan sistem dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, diuraikan juga beberapa saran yang dapat diperhatikan untuk pengembangan selanjutnya.

5.1 Kesimpulan

Berdasarkan eksperimen yang telah dilakukan, terdapat beberapa kesimpulan yang dapat ditarik dari penelitian ini adalah:

1. Metode data *hiding* usulan *Random Reduce Difference Expansion* dapat mengatasi masalah *overflow* dan *underflow*. Dimana nilai kapasitas yang dihasilkan sama dengan banyak data. Akan tetapi *location map* yang dihasilkan memiliki redundansi yang rendah sehingga *location map* tidak dapat *dicompress* dan disisipkan pada *cover* audio.
2. Metode partisi usulan yang dihasilkan mampu meningkatkan kapasitas penyisipan akan tetapi nilai SNR yang dihasilkan lebih kecil jika dibandingkan dengan hasil pengujian tanpa menggunakan metode partisi. Metode partisi usulan jika digunakan dengan metode penyisipan DE dan RDE, metode ini dapat meningkatkan kualitas dan kapasitas hasil penyisipan.
3. Penggunaan skema *Multi-layer* pada metode RRDE tidak mempengaruhi nilai kapasitas dari audio hasil penyisipan, sedangkan nilai kualitas yang dihasilkan tetap stabil sehingga penyisipan dapat dilakukan dengan banyak *Layer* yang tidak terbatas. Akan tetapi dengan nilai kapasitas yang semakin meningkat ukuran *Locatio Map* juga ikut meningkat.

5.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Metode *Random Reduce Difference Expansion* masih dapat dikembangkan seperti mengganti nilai random menjadi satu nilai tetap yang nantinya akan dijadikan acuan yang dapat digunakan untuk me *recover* media, pesan dan juga dapat men *generate location map* yang memiliki redundansi tinggi sehingga dapat di *compress* dan dijadikan *payload* agar dapat disisipkan pada *cover* audio.
2. Masih terdapat banyak modifikasi dari metode *Intelligent Partitioning* yang pada kasus-kasus tertentu dapat memiliki hasil yang lebih baik. Karena metode partisi yang baik adalah metode yang dapat mendukung metode penyisipannya.

DAFTAR PUSTAKA

- Ahmad, T., Holil, M., Wibisono, W., & others. (2013). An improved Quad and RDE-based medical data hiding method. In *Computational Intelligence and Cybernetics (CYBERNETICSCOM), 2013 IEEE International Conference on* (pp. 141–145). IEEE.
- Alattar, A. M. (2004a). Reversible watermark using difference expansion of quads. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on* (Vol. 3, p. iii–377). IEEE.
- Alattar, A. M. (2004b). Reversible watermark using the difference expansion of a generalized integer transform. *Image Processing, IEEE Transactions on*, 13(8), 1147–1156.
- Choi, K.-C., Pun, C.-M., & Chen, C. P. (2015). Application of a generalized difference expansion based reversible audio data hiding algorithm. *Multimedia Tools and Applications*, 74(6), 1961–1982.
- Faruque, S. (2015). Pulse Code Modulation (PCM). In *Radio Frequency Source Coding Made Easy* (pp. 65–90). Springer.
- Irmas Dataset. (2012). Retrieved June 14, 2016, from <http://www.mtg.upf.edu/download/datasets/irmas?p=Download>
- Lin, C.-C., Yang, S.-P., & Hsueh, N.-L. (2008). Lossless data hiding based on difference expansion without a location map. In *Image and Signal Processing, 2008. CISP'08. Congress on* (Vol. 2, pp. 8–12). IEEE.
- Lin, Y., & Abdulla, W. (2011). Objective quality measures for perceptual evaluation in digital audio watermarking. *IET Signal Processing*, 5(7), 623–631.
- Liu, W. M., Jellyman, K., Mason, J. S., Evans, N. W., & others. (2006). Assessment of objective quality measures for speech intelligibility estimation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on* (Vol. 1, pp. I–I). IEEE.
- Lou, D.-C., Hu, M.-C., & Liu, J.-L. (2009). Multiple layer data hiding scheme for medical images. *Computer Standards & Interfaces*, 31(2), 329–335.

- Quackenbush, S. R., Barnwell, T. P., & Clements, M. A. (1988). *Objective measures of speech quality*. Prentice Hall.
- Tian, J. (2003). Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Techn.*, 13(8), 890–896.
- Zölzer, U. (2008). *Digital audio signal processing*. John Wiley & Sons.

BIODATA PENULIS



Dwi Shinta Angreni yang sering di panggil Shinta lahir pada tanggal 20 Juli 1991 di kota Palu provinsi Sulawesi Tengah. Penulis merupakan anak kedua dari tiga bersaudara dan putri pertama dari bapak Suyatno dan ibu Nurjannah. Memulai pendidikannya pada umur enam tahun di SDN Inpres 1 Tanamodindi pada tahun 1997. Setelah menamatkan pendidikan dasar pada tahun 2003 penulis melanjutkan pendidikannya di SMP Negeri 1 Palu pada tahun yang sama. Pada tahun 2006 penulis berhasil melanjutkan pendidikannya di SMA Negeri 1 Palu dan menyelesaikannya pada tahun 2009. Penulis melanjutkan pendidikan S1 di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam pada tahun 2009-2014. Pada tahun yang sama penulis terdaftar menjadi mahasiswa S2 Jurusan Teknik Informatika Fakultas Teknologi Informasi. Selama mengikuti program S2, penulis mengambil bidang minat Komputasi Berbasis Jaringan(KBJ) dengan topik penelitian Keamanan Jaringan. Penulis dapat dihubungi melalui email: shintadwi.suyatno@gmail.com.