



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR

# RANCANG BANGUN METODE DATA HIDING DENGAN MENGGUNAKAN *FUZZY LOGIC*

MOH. MUZAYYIN AMRULLOH  
NRP 0511164000095

Dosen Pembimbing I  
TOHARI AHMAD, S.Kom., MIT., Ph.D.

DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

*[Halaman ini sengaja dikosongkan]*



**TUGAS AKHIR**

# **Rancang Bangun Metode Data Hiding Dengan Menggunakan *Fuzzy Logic***

**MOH. MUZAYYIN AMRULLOH**  
**NRP 0511164000095**

**Dosen Pembimbing I**  
**TOHARI AHMAD, S.Kom., MIT., Ph.D.**

**DEPARTEMEN TEKNIK INFORMATIKA**  
**Fakultas Teknologi Elektro dan Informatika Cerdas**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2020**

*[Halaman ini sengaja dikosongkan]*



**UNDERGRADUATE THESES**

# **Design and Build Data Hiding Method Using Fuzzy Logic**

**MOH. MUZAYYIN AMRULLOH**  
**NRP 0511164000095**

**Supervisor I**  
**TOHARI AHMAD, S.Kom., MIT., Ph.D.**

**DEPARTMENT INFORMATICS**  
**FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS**  
**TECHNOLOGY**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA 2020**

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN

**Rancang Bangun Metode Data Hiding Dengan Menggunakan  
*Fuzzy Logic***

### TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer pada  
Bidang Studi Komputasi Berbasis Jaringan  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :

**MOH. MUZAYYIN AMEULLOH**  
**NRP : 0511164000095**

Disetujui oleh Dosen Pembimbing Tugas Akhir :

TOHARI AHMAD, S.Kom., MIT., Ph.D.  
NIP: 197505252003121002



*Tohari Ahmad*  
.....  
(pembimbing 1)

**SURABAYA**  
**JUNI 2020**

*[Halaman ini sengaja dikosongkan]*



## **Rancang Bangun Metode Data Hiding Dengan Menggunakan *Fuzzy Logic***

**Nama Mahasiswa** : Moh. Muzayyin Amrulloh  
**NRP** : 05111640000095  
**Jurusan** : Departemen Informatika FTEIC  
**Dosen Pembimbing I** : Tohari Ahmad, S.Kom., MIT.,  
Ph.D.

### **ABSTRAK**

Perkembangan teknologi informasi telah membawa konsekuensi yang dapat dihindari untuk keamanan data. Hal tersebut dikarenakan data, khususnya yang bersifat pribadi atau rahasia, rentan terhadap akses ilegal. Oleh karena itu, metode pengamanan, seperti penyembunyian data sangat penting. Hal ini dilakukan dengan menyisipkan pesan rahasia (*payload*) ke dalam suatu media (*cover*). Salah satu media yang biasa digunakan dalam metode ini adalah audio, di mana *payload* tertanam dalam sampel audio. Namun demikian, teknik umum ini memiliki beberapa kelemahan salah satunya adalah ruang penyisipan yang relatif kecil dan perbedaan yang besar antara stego-audio dan audio aslinya. Selain itu, ada risiko *payload* yang tidak dapat dikembalikan pada saat diekstraksi. Untuk mengatasi masalah ini, kami mengusulkan untuk menggunakan tahap *fuzzifikasi* dalam algoritma *Fuzzy Logic*. Hasil akhir menunjukkan peningkatan kualitas *stego-audio*, yang diukur dengan *Peak Signal-to-Noise Ratio* (PSNR).

**Kata kunci** : *Data hiding, Audio steganography, Network security, Fuzzy Logic, cybercrime.*

# Design and Build Data Hiding Method Using Fuzzy Logic

**Student Name** : Moh. Muzayyin Amrulloh  
**NRP** : 0511164000095  
**Major** : Informatics Departement FTEIS  
**Advisor I** : Tohari Ahmad, S.Kom., MIT.,  
Ph.D.

## ABSTRACT

The development of information technology allows threats to data security. Almost all data are very vulnerable to public access. Even if the data is private or confidential. So, we need a method to hide the data. Steganography is one method to secure data by inserting secret message (*payload*) into a particular medium (cover). One of the media commonly used in steganography is audio, where a payload will be inserted into an audio file. Base on several studies that have been done, the audio steganography technique has several disadvantages including insertion space that is too small, the similarity between stego-audio and audio cover is far different and there is a risk of payload that cannot return when extracted. Therefore, in this Paper, we proposed to design the fuzzification stage in the Fuzzy Logic method to improve the quality of the stego-audio. The final results show an increase in stego audio quality measured by the value of Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR).

**Keyword** : *Data hiding, Audio steganography, Network security, Fuzzy Logic, cybercrime.*

## KATA PENGANTAR

Alhamdulillah rabbil 'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Metode Data Hiding Dengan Menggunakan Fuzzy Logic”**.

Penulisan Tugas Akhir ini adalah salah satu pengalaman dan kesempatan yang sangat membanggakan bagi penulis. Dengan penulisan Tugas Akhir ini, penulis dapat belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan selama dalam proses perkuliahan.

Selesainya pengerjaan tugas akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak, sehingga pada kesempatan ini penulis mengucapkan syukur dan banyak terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Ayah, Ibu, kakak, dan Adik yang selalu mengingatkan, menyemangati, mendoakan dan memberikan yang terbaik untuk penulis dalam pengerjaan Tugas Akhir.
3. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. selaku pembimbing I dan selaku Kepala Laboratorium Komputasi Berbasis Jaringan yang tetap sabar dan selalu memberikan motivasi dan membimbing penulis selama pengerjaan tugas akhir.
4. Ibu Dr. Eng. Nanik Suciati, S.kom, M.Kom. Selaku dosen wali yang telah membantu dalam memberi motivasi, pengarahan, saran, dan masukan bagi penulis baik dalam hal perkuliahan maupun pengembangan diri.
5. Ibu Dr. Eng. Chastine Fatichah, S.Kom, M.Kom. selaku ketua jurusan yang telah membantu proses perkuliahan penulis.
6. Bapak Ary Mazharuddin Shiddiqi S.kom, M.Comp.Sc. selaku ketua prodi jurusan yang telah membantu dan mempermudah penulis dalam persiapan tugas akhir.

7. Ibu Eva Mursidah A.Md. selaku pustakawan Ruang Baca Teknik Informatika yang telah mempermudah penulis dalam melakukan peminjaman buku referensi untuk tugas akhir.
8. M. Iqbal Izzul Haq, Ulfatuz Zahroh, dan Itsna Dzakiatul Huriroh selaku teman kelompok yang selalu membantu penulis baik dalam pengerjaan tugas akhir maupun proses perkuliahan.
9. Ubut Eka Putra, Faizal Khilmi Muzakki, Azkiatunnisa Rahma Fajriyanti, dan Nuzha Mushafirah selaku teman angkatan dan teman admin Laboratorium Komputasi Berbasis Jaringan yang membantu penulis baik dalam pengerjaan tugas akhir maupun proses perkuliahan.
10. Teman teman Admin Laboratorium Komputasi Berbasis Jaringan yang membantu penulis dalam menyediakan fasilitas dan turut menyemangati dalam proses pengerjaan tugas akhir.
11. Sesama rekan satu RMK Tugas Akhir yang turut membantu penulis selama proses pengerjaan tugas akhir.
12. Teman-teman angkatan 2016 yang telah membantu, mendukung, berbagi pengalaman, berbagi ilmu, berbagi kebahagiaan, menjaga solidaritas, dan memberi motivasi kepada penulis.
13. kakak-kakak angkatan 2015 dan adik-adik angkatan 2017, 2018, dan 2019 yang membuat penulis untuk selalu semangat dalam belajar.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan, sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2020

Penulis

## DAFTAR ISI

<b>LEMBAR PENGESAHAN.....</b>	<b>vii</b>
<b>Abstrak .....</b>	<b>ix</b>
<b>Abstract .....</b>	<b>x</b>
<b>KATA PENGANTAR .....</b>	<b>xi</b>
<b>1 DAFTAR ISI.....</b>	<b>xiii</b>
<b>1 DAFTAR GAMBAR .....</b>	<b>xvii</b>
<b>1 DAFTAR TABEL.....</b>	<b>xix</b>
<b>2 DAFTAR PSEUDOCODE.....</b>	<b>xx</b>
<b>1 BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Permasalahan .....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan .....	2
1.5. Manfaat .....	2
1.6. Metodologi.....	3
1.6.1. Penyusunan Proposal Tugas Akhir.....	3
1.6.2. Studi Literatur.....	3
1.6.3. Perancangan Sistem.....	3
1.6.4. Implementasi Sistem.....	4
1.6.5. Pengujian dan evaluasi .....	4
1.6.6. Penyusunan buku Tugas Akhir.....	4
1.7. Sistematika Penulisan Laporan Tugas Akhir .....	4
<b>2 BAB II TINJAUAN PUSTAKA .....</b>	<b>6</b>
2.1. Steganografi .....	6
2.2. Fuzzy Logic .....	6
2.3. Fuzzifikasi.....	7
2.4. Means Square Error (MSE).....	7
2.5. Peak Signal-to-Noise Ratio (PSNR) .....	8
2.6. Python .....	8
2.7. Visual Studio Code .....	9
<b>3 BAB III PERANCANGAN SISTEM.....</b>	<b>10</b>
3.1. Deskripsi Umum .....	10
3.2. Perancangan Data.....	11

3.2.1.	Data Masukan .....	11
3.2.2.	Data Keluaran .....	12
3.2.3.	Data Informasi .....	12
3.3.	Perancangan Metode Fuzzifikasi .....	13
3.3.1.	Proses Penyisipan .....	13
3.3.2.	Proses Ekstraksi.....	19
<b>4</b>	<b>BAB IV IMPLEMENTASI.....</b>	<b>25</b>
4.1.	Modul Penyisipan .....	25
4.1.1.	Fungsi <i>Sampling</i> .....	25
4.1.2.	Fungsi Interpolasi Sampel .....	26
4.1.3.	Fungsi <i>Fuzzifikasi</i> .....	26
4.1.4.	Fungsi Penentuan Ruang Sampel .....	27
4.1.5.	Fungsi Segmentasi <i>Payload</i> .....	27
4.1.6.	Fungsi <i>Penyisipan</i> .....	28
4.1.7.	Fungsi Penggabungan.....	29
4.1.8.	Proses Pembuatan Stego-audio.....	29
4.2.	Modul Ekstraksi .....	30
4.2.1.	Fungsi <i>Sampling</i> .....	30
4.2.2.	Fungsi <i>Pemecahan Sampel</i> .....	31
4.2.3.	Fungsi Interpolasi Sampel .....	31
4.2.4.	Fungsi <i>Fuzzifikasi</i> .....	31
4.2.5.	Fungsi Penentuan Ruang Sampel .....	31
4.2.6.	Fungsi Pengambilan <i>Payload</i> .....	32
4.2.7.	Fungsi Rekonstruksi Audio .....	32
4.3.	Modul Analisis.....	33
4.3.1.	Fungsi MSE.....	33
4.3.2.	Fungsi PSNR .....	34
<b>5</b>	<b>BAB V UJI COBA DAN EVALUASI.....</b>	<b>35</b>
5.1.	Lingkungan Uji Coba.....	35
5.2.	Data Pengujian .....	35
5.2.1.	<i>Cover</i> Audio .....	35
5.2.2.	<i>Payload</i> .....	36
5.3.	Skenario Uji Coba dan Evaluasi Pengujian .....	37
5.3.1.	Skenario Uji Coba 1 .....	37
5.3.2.	Evaluasi Uji Coba 1 .....	38

5.3.3.	Skenario Uji Coba 2 .....	42
5.3.4.	Evaluasi Uji Coba 2.....	42
<b>6</b>	<b>BAB VI KESIMPULAN DAN SARAN .....</b>	<b>49</b>
6.1.	Kesimpulan .....	49
6.2.	Saran .....	49
<b>7</b>	<b>DAFTAR PUSTAKA .....</b>	<b>51</b>
<b>8</b>	<b>BIODATA PENULIS .....</b>	<b>52</b>

*[Halaman ini sengaja dikosongkan]*



## DAFTAR GAMBAR

<b>Gambar 3.1</b> Diagram Alir Proses Penyisipan .....	14
<b>Gambar 3.2</b> Diagram Fuzzifikasi .....	16
<b>Gambar 3.3</b> Diagram Alir Proses Ekstraksi .....	20
<b>Gambar 5.1</b> Hasil Perbandingan PSNR .....	41
<b>Gambar 5.2.</b> Perbandingan Total Penggunaan Sampel Dengan dan Tanpa Distribusi Payload .....	45
<b>Gambar 5.3.</b> Metode Distribusi Payload .....	46
<b>Gambar 5.4.</b> Tanpa Distribusi Payload .....	47
<b>Gambar 5.5.</b> Hasil Perbandingan PSNR dengan dan tanpa Distribusi Payload .....	47

*[Halaman ini sengaja dikosongkan]*

**DAFTAR TABEL**

<b>Tabel 3.1</b> Data Informasi .....	13
<b>Tabel 5.1</b> Lingkungan Uji Coba .....	35
<b>Tabel 5.2</b> Detil Cover Audio.....	36
<b>Tabel 5.3</b> Detil Payload .....	37
<b>Tabel 5.4.</b> Hasil Perbandingan PSNR .....	39
<b>Tabel 5.5</b> Hasil Perbandingan PSNR.....	40
<b>Tabel 5.6</b> Total Penggunaan Sampel pada Distribusi Payload ...	43
<b>Tabel 5.7.</b> Total Penggunaan Sampel Tanpa Distribusi Payload	44

## DAFTAR PSEUDOCODE

<b>Pseudocode 4.1</b> Membaca Sampel Audio .....	25
<b>Pseudocode 4.2</b> Proses Interpolasi.....	26
<b>Pseudocode 4.3</b> Proses Fuzzifikasi.....	27
<b>Pseudocode 4.4</b> Proses Penentuan Ruang Sampel.....	27
<b>Pseudocode 4.5</b> Proses Segmentasi Payload .....	28
<b>Pseudocode 4.6</b> Proses Penyisipan Payload .....	29
<b>Pseudocode 4.7</b> Proses Penggabungan Sampel Audio .....	29
<b>Pseudocode 4.8</b> Prose Pembuatan Stego-audio .....	30
<b>Pseudocode 4.9</b> Proses Membaca Sampel Audio .....	31
<b>Pseudocode 4.10</b> Proses Pemecahan Sampel Audio.....	31
<b>Pseudocode 4.11</b> Proses Pengambilan Payload.....	32
<b>Pseudocode 4.12</b> Proses Rekonstruksi Audio.....	33
<b>Pseudocode 4.13</b> MSE.....	33
<b>Pseudocode 4.14</b> PSNR .....	34

*[Halaman ini sengaja dikosongkan]*



# BAB I

## PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Diharapkan dari penjelasan dalam bab ini gambaran tugas akhir secara umum dapat dipahami.

### 1.1. Latar Belakang

Semakin berkembangnya suatu teknologi dan informasi membuat segala macam bentuk data belum bisa dikatakan aman dari pihak yang tidak di inginkan. Hampir semua data sangat rentan diakses oleh publik. Sekalipun data atau informasi yang bersifat privasi atau rahasia. Sehingga dibutuhkan suatu metode untuk mengamankan data atau informasi tersebut. Beberapa metode telah diperkenalkan dalam mengamankan data rahasia, salah satunya adalah steganografi.

Steganografi adalah salah satu metode untuk mengamankan data dengan cara menyisipkan pesan rahasia (*payload*) ke dalam suatu media tertentu (*cover*). Media yang biasa digunakan dalam steganografi salah satunya adalah media audio, dimana suatu pesan rahasia (*payload*) akan disisipkan kedalam suatu berkas audio.

Pada Tugas Akhir ini diusulkan mengimplementasi tahap *fuzzifikasi* pada metode *Fuzzy Logic* untuk meningkatkan kualitas *stego audio*. Hasil akhir yang diharapkan adalah meningkatnya kualitas *stego audio* yang akan diukur berdasarkan nilai *Mean Square Error* (MSE) dan *Peak Signal-to-Noise Ratio* (PSNR).

## 1.2. Rumusan Permasalahan

Berikut beberapa hal yang menjadi rumusan masalah dalam Tugas Akhir ini:

1. Bagaimana implementasi yang akan dilakukan dengan *Fuzzy Logic* ?
2. Bagaimana dampak implementasi *Fuzzy Logic* terhadap kualitas *stego-audio* jika diukur berdasarkan hasil MSE dan PSNR ?
3. Apakah isi file *payload* dapat kembali seperti semula setelah dilakukan proses ekstraksi ?

## 1.3. Batasan Masalah

Batasan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Medium yang akan digunakan adalah file audio bertipe .wav.
2. File *payload* (data yang akan disembunyikan) bertipe .txt.

## 1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini antara lain :

1. Meningkatkan kualitas *stego-audio* setelah dilakukan implementasi metode *Fuzzy Logic*
2. Memastikan seluruh *payload* dapat berhasil disisipkan dan diekstraksi tanpa adanya kesalahan.

## 1.5. Manfaat

Manfaat dari pembuatan tugas akhir ini antara lain :

1. Mengetahui dampak implementasi metode *Fuzzy Logic* terhadap kualitas *stego-audio*
2. Jaminan bahwa *payload* dapat diambil kembali seperti semula.



## **1.6. Metodologi**

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

### **1.6.1. Penyusunan Proposal Tugas Akhir**

Proposal tugas akhir ini terdiri dari 8 bab. Bab 1 berisi tentang saya sebagai penulis proposal. Bab 2 berisi judul tugas akhir yang sedang diusulkan. Bab 3 yaitu berisi pendahuluan yang terdiri dari beberapa subbab antara lain latar belakang tugas akhir, rumusan masalah, batasan masalah, tujuan tugas akhir, dan manfaat dari hasil pengerjaan tugas akhir. Bab 4 berisi tinjauan pustaka yaitu data yang dipakai sebagai referensi pengerjaan tugas akhir. Bab 5 berisi ringkasan tugas akhir yang berisi aturan pengerjaannya. Bab 6 berisi metodologi yang digunakan mulai dari tahap penyusunan proposal hingga penyusunan buku tugas akhir. Kemudian, Bab 7 berisi timeline pengerjaan yang menunjukkan jadwal pengerjaan tugas akhir. Sedangkan Bab 8 berisi daftar pustaka yang dipakai dalam penyusunan proposal.

### **1.6.2. Studi Literatur**

Pada studi literatur ini akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan sistem yaitu terkait bahasa pemrograman Python, steganografi dan metode *Fuzzy Logic*

### **1.6.3. Perancangan Sistem**

Pada tahap ini berisi perancangan sistem berdasarkan studi literature dan pembelajaran konsep teknologi dari sistem yang ada. Tahap ini mendefinisikan alur dari implementasi. Serta berisi langkah-langkah yang dikerjakan. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem

yang akan dibuat. Kemudian dilakukan desain suatu sistem dan desain proses-proses yang ada.

#### **1.6.4. Implementasi Sistem**

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

#### **1.6.5. Pengujian dan evaluasi**

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat sesuai dengan desain sistem untuk mengetahui kemampuan aplikasi, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin timbul pada aplikasi yang dibuat.

#### **1.6.6. Penyusunan buku Tugas Akhir**

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

### **1.7. Sistematika Penulisan Laporan Tugas Akhir**

Pada tahap ini dilakukan penyusunan paoran tugas akhir yang berisi penjelasan dasar teori dan metode yang digunakan serta berisi hasil implementasi dan pengujian aplikasi perangkat lunak. Berikut sistematika penulisan buku tugas akhir ini secara garis besar :

#### **Bab I Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan

masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

## **Bab II Tinjauan Pustaka**

Bab ini berisi penjelasan secara detail mengenai dasar teori serta peunjang yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

## **Bab III Perancangan Sistem**

Bab ini membahas tentang desain sistem yang disajikan dalam bentuk deskripsi rinci dari tahapan-tahapan proses yang ada dalam sistem.

## **Bab IV Implementasi**

Bab ini berisi implementasi dari desain yang telah dibuat pada bab sebelumnya serta penjelasan berupa pseudocode yang digunakan untuk proses implementasi.

## **Bab V Uji Coba Dan Evaluasi**

Bab ini menjelaskan kemampuan sistem dengan melakukan pengujian dari sistem yang telah dibuat.

## **Bab VI Kesimpulan Dan Saran**

Bab ini menjelaskan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan sistem ke depannya.

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini berisi teori-teori dasar atau penjelasan dari metode dan tools yang digunakan dalam pembuatan tugas akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara garis besar tentang rancangan sistem yang akan dibuat dan sebagai penunjang dalam penelitian yang berkaitan.

#### **2.1. Steganografi**

Steganografi adalah salah satu teknik menulis pesan tersembunyi atau menyembunyikan pesan dengan suatu cara sehingga dapat membatasi seseorang untuk mengakses data atau informasi yang bersifat privasi. Jadi selain si pengirim dan si penerima, tidak ada satu orang pun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia yang terkandung didalamnya

Steganografi terdiri atas pesan rahasia (*payload*) dan cover media, dimana pesan rahasia akan disisipkan dalam cover media dengan menyamarkan perbedaan sesedikit mungkin antara cover media sebelum dan setelah disisipkan pesan rahasia. Jenis cover media yang umum dipakai dalam steganografi adalah: gambar, audio dan video

#### **2.2. Fuzzy Logic**

*Fuzzy* secara bahasa diartikan dengan samar samara atau kabur. Hal yang dimaksud samar samar adalah suatu nilai dapat bernilai benar atau salah dalam satu waktu. Namun seberapa besar nilai kebenaran dan kesalahannya tergantung pada derajat keanggotaan yang dimilikinya [1]. Logika fuzzy digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (linguistik), misalkan besaran kecepatan laju kendaraan

yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Dan logika fuzzy menunjukkan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah. Tidak seperti logika tegas, suatu nilai hanya mempunyai 2 kemungkinan yaitu merupakan suatu anggota himpunan atau tidak. Derajat keanggotaan 0 (nol) artinya nilai bukan merupakan anggota himpunan dan 1 (satu) berarti nilai tersebut adalah anggota himpunan [2].

### 2.3. Fuzzifikasi

*Fuzzifikasi* adalah salah satu tahap pada metode *Fuzzy Logic* yang berfungsi untuk mengubah suatu masukan atau input dari bentuk tegas (crisp) menjadi bentuk Fuzzy (variabel linguistik) yang biasanya disajikan dalam bentuk himpunan – himpunan fuzzy dengan fungsi keanggotaan masing masing. Persamaan tersebut dapat dituliskan sebagai berikut.

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases}$$

Keterangan

a = Nilai domain terkecil saat derajat keanggotaan terkecil.

b = Derajat keanggotaan terbesar dalam domain.

c = Nilai domain terbesar saat derajat keanggotaan terkecil [3].

### 2.4. Means Square Error (MSE)

*Means Square Error* (MSE) adalah metode untuk menghitung nilai error antara sample audio asli (*cover*) dengan

sample hasil penyisipan data rahasia (*stego audio*). Persamaan tersebut dapat dituliskan sebagai berikut :

$$MSE = \frac{1}{N} \sum_{i=1}^N (S_i - S'_i)^2 \quad (1)$$

Notasi  $N$  merupakan banyaknya sampel pada audio. Notasi  $S_i$  merupakan sample dari audio asli. Notasi  $S'_i$  merupakan sample dari *stego audio* [4].

## 2.5. Peak Signal-to-Noise Ratio (PSNR)

*Peak Signal-to-Noise Ratio* (PSNR) adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR biasanya diukur dalam satuan decibel (db). PSNR digunakan untuk mengetahui perbandingan kualitas citra cover sebelum dan sesudah proses penyisipan *payload*.

$$PSNR = 10 * \log_{10} \frac{(2^b - 1)^2}{MSE} \quad (2)$$

Notasi  $2b$  merupakan nilai maksimum dari *bit-depth* dimana  $b$  bernilai 16 bit [4].

## 2.6. Python

Python adalah salah satu bahasa pemrograman tinggi yang dapat melakukan eksekusi sejumlah intruksi multiguna secara langsung (interpretatif) dengan metode orientasi objek (Object Oriented Programming) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan syntax [5]. Dalam implementasi Tugas Akhir ini, bahasa pemrograman Python yang akan digunakan adalah versi 3.8.0

## 2.7. Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Visual Studio Code mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code salah satunya adalah Python [6]. Dalam implementasi Tugas Akhir ini akan menggunakan aplikasi Visual Studio Code dengan Python sebagai *Project Interpreter*

## **BAB III**

### **PERANCANGAN SISTEM**

Bab ini menjelaskan tentang rancangan sistem yang akan dibuat. Sistem yang akan dibuat meliputi data dan proses. Data yang dimaksud adalah suatu input atau output yang akan diolah dalam untuk dijadikan sebagai pembelajaran maupun pengujian. Sedangkan yang dimaksud proses adalah tahap-tahap yang ada dalam sistem yang akan dibuat.

#### **3.1. Deskripsi Umum**

Dalam tugas Akhir ini akan dibuat sebuah sistem aplikasi perangkat lunak yang dapat melakukan steganografi audio dengan menggunakan bahasa pemrograman Python. Secara umum aplikasi ini dibagi menjadi 2 proses yaitu proses penyisipan dan proses ekstraksi. Proses penyisipan sendiri dibagi menjadi beberapa tahapan yaitu Normalisasi, interpolasi sampel, *fuzzifikasi*, penentuan ruang sampel, segmentasi *payload*, penyisipan *payload*, penggabungan, denormalisasi sampel, dan pembuatan *stego-audio*. Sedangkan pada proses ekstraksi sendiri dibagi menjadi beberapa proses yaitu normalisasi, pemecahan sampel, interpolasi sampel, *fuzzifikasi*, penentuan ruang sampel, pengambilan *payload*, denormalisasi sampel, dan pembentukan file *audio* asli.

Proses penyisipan adalah proses menyisipkan *payload* ke dalam sampel audio. Tahap pertama adalah normalisasi sampel. Tahap ini dilakukan penambahan sampel dengan nilai maksimum *bit-depth* audio yakni 16 bit. Tahap kedua yaitu interpolasi sampel audio. Tahap ini dilakukan dengan menentukan titik interpolasi antar sampel audio yang telah dinormalisasi. Tahap ketiga yaitu tahap *Fuzzifikasi*. Tahap ini dilakukan untuk menentukan derajat keanggotaan suatu sampel audio dengan grafik *fuzzy*. Tahap keempat yaitu penentuan ruang sampel. Tahap ini dilakukan dengan menentukan berapa banyak bit yang dapat



ditampung pada tiap ruang sampel. Tahap kelima adalah segmentasi *payload*. Tahap ini dilakukan dengan membagi data *payload* yang berbentuk biner menjadi bagian-bagian bit sesuai dengan distribusi *payload*. Tahap keenam adalah proses penyisipan *payload*. Tahap ini dilakukan dengan menyisipkan *payload* yang telah dibagi ke dalam setiap titik interpolasi pada sampel audio secara berurutan. Tahap ketujuh adalah proses penggabungan. Tahap ini dilakukan dengan menggabungkan sampel hasil penyisipan *payload* dengan sampel audio asli. Tahap kedelapan adalah proses denormalisasi. Tahap ini dilakukan dengan mengurangi sampel dengan nilai maksimum *bit-depth* audio yakni 16 bit. Kemudian, tahap terakhir adalah pembuatan *stego-audio*. Tahap ini sampel audio hasil denormalisasi akan dikonversi menjadi file *stego-audio* dengan format \*.wav.

Proses ekstraksi adalah proses pengambilan data *payload* dari *stego-audio*. Tahap pertama dari proses ekstraksi adalah tahap normalisasi sampel audio, dimana proses nya sama dengan proses normalisasi pada proses penyisipan. Tahap kedua adalah tahap pemecahan sampel. Tahap ini dilakukan dengan memecah sampel *stego-audio* menjadi 2 bagian yaitu sampel audio asli dan sampel penyisipan data *payload*. Tahap ketiga adalah interpolasi sampel. Tahap keempat adalah *fuzzifikasi*. Tahap kelima yaitu penentuan ruang sampel. Tahap keenam yaitu proses pengambilan *payload*. Tahap ini dilakukan pengambilan data *payload* yang disisipkan dalam sampel. Tahap ketujuh yaitu proses denormalisasi sampel . dan tahap terakhir adalah pembentukan file audio asli.

## **3.2. Perancangan Data**

Subbab ini berisi data masukan yang akan diolah dalam sistem dan data keluaran yang dihasilkan dari proses pengolahan sistem pada Tugas Akhir ini.

### **3.2.1. Data Masukan**

Data masukan adalah data awal yang dibutuhkan oleh sistem untuk diproses. Data masukan sendiri dibagi menjadi 2 jenis, yaitu data masukan pada proses penyisipan dan data masukan pada proses ekstraksi. Data masukan pada proses penyisipan terdiri dari 2 berkas, yaitu berkas audio dan berkas *payload*. Berkas audio yang digunakan memiliki format \*.wav *channel mono* dengan *bit-depth* 16 bit. Serta *frame rate* sebesar 44100 Hz. Sementara untuk berkas *payload* yang digunakan memiliki format \*.txt. Sedangkan untuk data masukan pada proses ekstraksi juga terdiri dari 2 berkas, yaitu berkas *stego-audio* dan berkas informasi tambahan. Berkas data informasi tambahan dapat dilihat pada **Tabel 3.1**. Berkas *stego-audio* yang digunakan memiliki format \*.wav, *channel-mono*, *bit-depth* 16 bit, dan *frame rates* sebesar 88200 Hz. Sementara berkas data informasi tambahan yang dibutuhkan dalam format \*.txt.

### 3.2.2. Data Keluaran

Data keluaran adalah data yang dihasilkan oleh sistem setelah melakukan proses. Data keluaran yang dihasilkan pada proses penyisipan terdiri dari 2 berkas, yaitu berkas *stego-audio* dan berkas data informasi tambahan. Berkas *stego-audio* yang dihasilkan memiliki format \*.wav, *channel-mono*, *bit-depth* 16 bit, dan *frame rates* sebesar 88200 Hz. Sementara berkas data informasi tambahan yang dihasilkan memiliki format \*.txt. Sedangkan data keluaran yang dihasilkan dalam ekstraksi terdiri dari 2 berkas, yaitu berkas *payload* dan audio. Berkas *payload* yang dihasilkan memiliki format \*.txt. Sementara berkas audio yang dihasilkan memiliki format \*.wav, *channel-mono*, *bit-depth* 16 bit, serta *frame rates* sebesar 44100 Hz.

### 3.2.3. Data Informasi

Data informasi adalah data yang dihasilkan ketika sistem melakukan proses penyisipan. Data informasi digunakan sebagai informasi tambahan untuk merekonstruksi *payload* dari *stego-audio*

dalam proses ekstraksi. Isi dari data informasi tersebut ditunjukkan pada **Tabel 3.1** Data Informasi **Tabel 3.1**.

**Tabel 3.1** Data Informasi

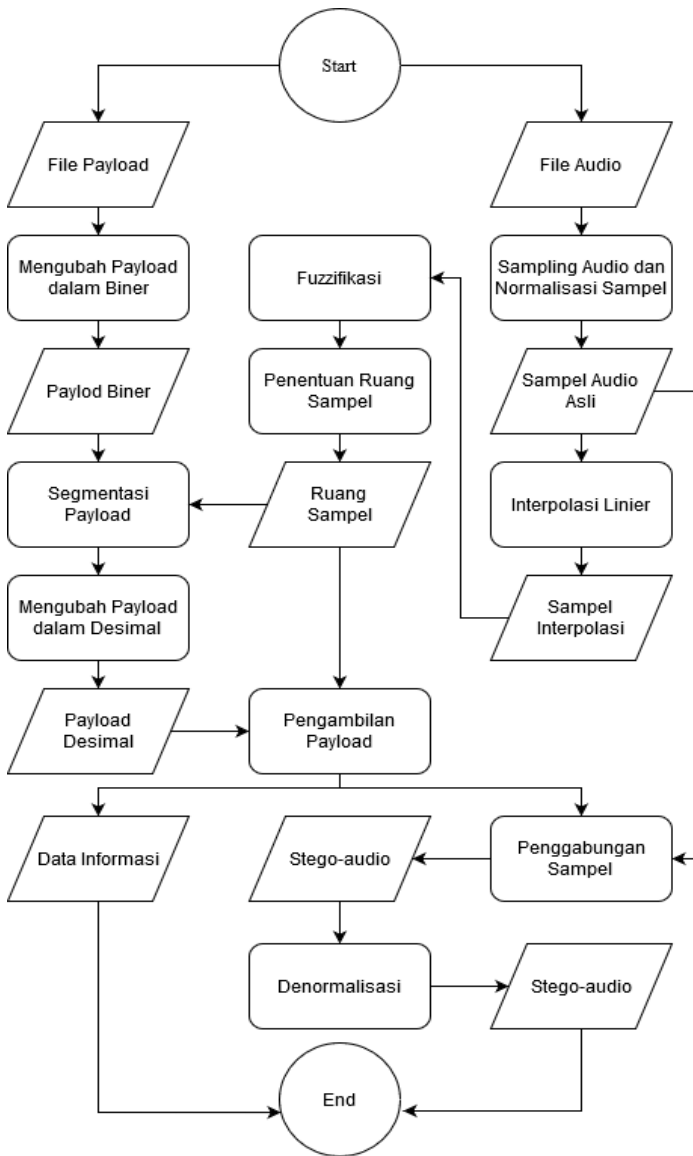
No	Nama Informasi	Keterangan
1	Distribusi payload	Jumlah bit payload yang didistribusikan pada setiap sampel.
2	Total sisipan	Total sampel yang digunakan untuk proses penyisipan payload.

### 3.3. Perancangan Metode *Fuzzifikasi*

Pada Tugas Akhir ini, metode yang digunakan dalam proses steganografi adalah metode *fuzzifikasi*. Metode ini digunakan untuk menentukan derajat keanggotaan suatu sampel audio dengan grafik Fuzzy. Dalam implementasinya, metode ini dilakukan pada proses penyisipan dan proses ekstraksi. Dasar penelitian yang digunakan pada penelitian ini dilakukan oleh Alghamdi [7] dan Ahmad dan Fiqar [8].

#### 3.3.1. Proses Penyisipan

Proses penyisipan adalah proses menyisipkan payload ke dalam sampel audio. Dalam proses ini dijelaskan langkah langkah dalam menyisipkan payload menggunakan metode *Fuzzifikasi*. Skema dari proses penyisipan dapat dilihat pada Gambar 3.1 Diagram Alir Proses Penyisipan.



**Gambar 3.1** Diagram Alir Proses Penyisipan

### 3.3.1.1. Normalisasi

Dalam tahap ini, dilakukan pengambilan nilai sampel dari data masukan yang berupa berkas audio dalam format \*.wav. Kemudian, dilakukan penambahan semua nilai sampel dengan 32768. Jadi, nilai sampel yang mulanya dengan range -32768 hingga 32767 diubah menjadi dalam *range* 0 hingga 65535. Hasil normalisasi sampel ini yang akan dilanjutkan ke tahap interpolasi sampel.

### 3.3.1.2. Interpolasi sampel

Interpolasi sampel adalah tahap menentukan titik interpolasi antar sampel audio yang telah dinormalisasi untuk penyisipan *payload*. Metode yang digunakan dalam menentukan titik-titik ini yaitu interpolasi linear. Rumus dari metode ini dapat dilihat pada persamaan (3).

$$S'_n = \left\lfloor \frac{S_n + S_{(n+1)}}{2} \right\rfloor \quad (3)$$

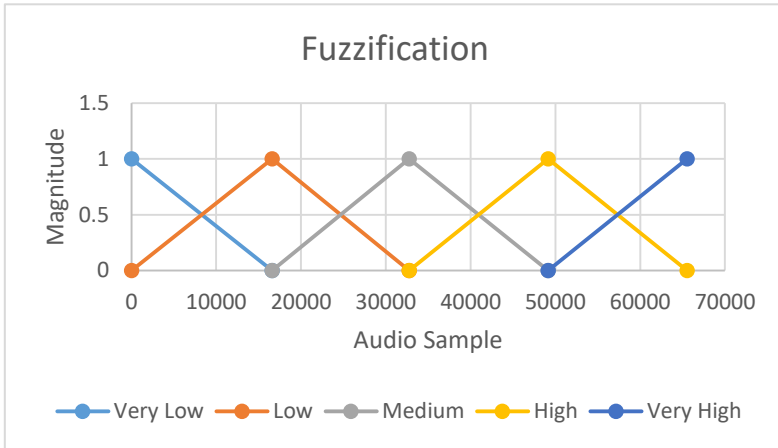
Notasi  $S'_n$  adalah hasil interpolasi pada index ke- $n$ . Lalu notasi  $S_n$  adalah nilai asli sampel audio pada index ke- $n$ .

### 3.3.1.3. Fuzzifikasi

*Fuzzifikasi* adalah tahap menentukan kategori sampel dari berkas masukan berupa berkas audio yang telah di interpolasi menjadi bentuk *Fuzzy*. Pada tahap ini, kategori sampel dibagi menjadi 5 kategori, yaitu Very Low, Low, Medium, High, dan Very High [9] seperti pada **Gambar 3.2**.

Kemudian sampel akan dilakukan perhitungan menggunakan persamaan (4). Keluaran dari tahap fuzzifikasi terdiri dari 2 nilai, yaitu batas atas dan batas bawah yang berkisar

antara 0 hingga 1 sesuai pada grafik *fuzzy* pada **Gambar 3.2**. Hasil tersebut yang akan dilanjutkan ke tahap penentuan ruang sample.



**Gambar 3.2** Diagram Fuzzifikasi

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \end{cases} \quad (4)$$

Keterangan

a = Nilai domain terkecil saat derajat keanggotaan terkecil.

b = Derajat keanggotaan terbesar dalam domain.

c = Nilai domain terbesar saat derajat keanggotaan terkecil [3].

### 3.3.1.4. Penentuan Ruang Sampel

Tahap penentuan ruang sampel digunakan untuk menentukan jumlah bit yang dapat disisipkan dari setiap nilai sampel hasil dari *fuzzifikasi*. Data masukan dari tahap ini terdiri dari 2 jenis, yaitu batas atas dan batas bawah dari keluaran tahap

*fuzzifikasi*. Kemudian, 2 nilai tersebut dilakukan perhitungan menggunakan persamaan (5), (6) dan dilanjutkan pada persamaan (7).

$$Xa_n = ba_n | S_n - S_{n+1} | \quad (5)$$

$$Xb_n = bb_n | S_n - S_{n-1} | \quad (6)$$

$$Total_n = \lfloor Xa_n + Xb_n \rfloor \quad (7)$$

Notasi  $Total_n$  menunjukkan ruang sampel dalam desimal pada index ke- $n$ .  $ba_n$  dan  $bb_n$  menunjukkan keluaran dari tahap fuzzifikasi, yakni batas atas dan batas bawah pada index ke- $n$ . Kemudian  $S_n$  menunjukkan sampel audio pada index ke- $n$ .

Setelah didapatkan nilai  $Total_n$ , selanjutnya mencari jumlah bit yang dapat disisipkan dalam tiap ruang sampel yang ada menggunakan rumus (8).

$$N_n = \lfloor \log_2 Total_n \rfloor \quad (8)$$

Notasi  $N_n$  menunjukkan jumlah bit yang dapat disisipkan pada sampel interpolasi index ke- $n$ . Nilai tersebut yang akan menjadi masukan pada tahap segmentasi *payload*.

### 3.3.1.5. Segmentasi *Payload*

Pada tahap ini, mula-mula *payload* yang telah diubah ke dalam format biner akan dibagi sesuai dengan hasil perhitungan bit pada distribusi *payload* (9). Hal tersebut bertujuan untuk menentukan berapa banyak bit yang akan dibagikan ke tiap sampel interpolasi. Jumlah bit ini menyesuaikan kapasitas bit yang telah disediakan pada tahap penentuan ruang sampel (10).

Notasi  $Sp_n$  menunjukkan segmentasi *payload* pada index ke- $n$ . Sedangkan Notasi  $Dp$  menunjukkan jumlah bit yang dibagikan ke setiap sampel interpolasi.

$$Dp = \left\lceil \frac{\text{Total payload bit}}{\text{Total interpolation sample}} \right\rceil \quad (9)$$

$$Sp_n = \begin{cases} N_n, & \text{if } N_n < Dp \\ Dp, & \text{if } N_n \geq Dp \end{cases} \quad (10)$$

Jika pada tahap segmentasi *payload* terdapat sisa *payload* yang belum disisipkan dikarenakan kekurangan sampel untuk disisipkan atau banyaknya sampel yang tidak dapat disisipkan *payload*, maka nilai dari hasil perhitungan pada rumus (9) akan diperbarui dengan cara dilakukan penambahan dengan nilai 1 (satu) seperti pada persamaan (11). Kemudian, dilakukan perhitungan rumus (10) kembali hingga dapat dipastikan semua *payload* dapat disisipkan ke dalam sampel interpolasi.

$$Dp' = Dp + 1 \quad (11)$$

Setelah dibagi, hasilnya dikonversi kembali dalam bentuk decimal. Hasil akhir segmentasi akan dinotasikan dalam  $P$ . Berikut Contoh dari tahap ini.

$$\begin{aligned} N &= \{3,2,4,1,2,4,3\} \\ \text{Payload} &= 1011001011001 \\ Dp &= 2 \\ \text{Segmentasi}_{\text{payload}} &= |10|11|00|1|01|10|01| \\ P &= \{2,3,0,1,1,2,1\} \end{aligned}$$

### 3.3.1.6. Penyisipan *Payload*

Tahap penyisipan *payload* adalah proses menyisipkan data *payload* dari hasil proses segmentasi pada sampel interpolasi yang sudah ditentukan sesuai dengan index yang ada. Metode penyisipan yang dilakukan dapat dilihat pada persamaan (12).

$$S'_n = S_n - P_n \quad (12)$$



Notasi  $S'_n$  menunjukkan nilai sampel baru setelah disisipkan *payload* pada index ke- $n$ .

### 3.3.1.7. Penggabungan

Pada tahap ini dilakukan penggabungan antara sampel dari hasil penyisipan dengan sampel audio asli. Hal ini akan mengakibatkan penambahan jumlah sampel yang mulanya dari  $n$  menjadi  $2n - 1$ . Metode penggabungan yang dilakukan dapat dilihat pada (13).

$$S_i = \begin{cases} S_{i/2}, & \text{if } i \text{ is even number} \\ S'_{t,(i-1)/2}, & \text{if } i \text{ is odd number} \end{cases} \quad (13)$$

### 3.3.1.8. Denormalisasi Sampel

Pada tahap ini dilakukan proses mengembalikan nilai audio pada *range* seperti sebelum dilakukan normalisasi, yaitu dengan cara semua sampel hasil penggabungan dilakukan pengurangan dengan nilai 32768 untuk mengembalikan nilai sampel dalam *range* -32768 hingga 32767.

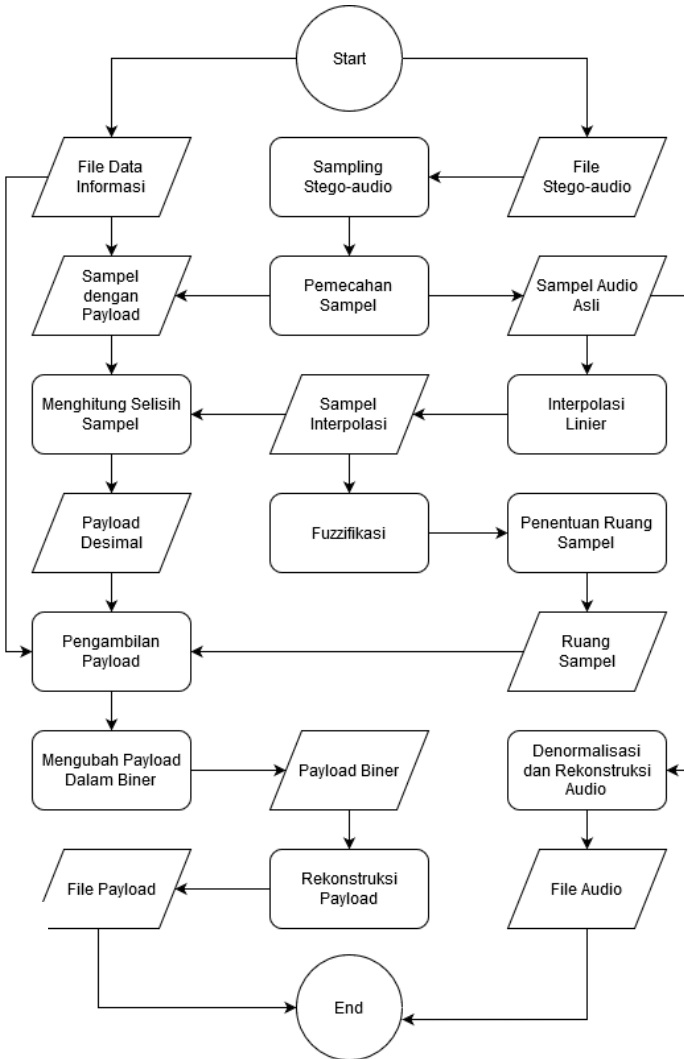
### 3.3.1.9. Pembuatan *Stego-Audio*

Tahap ini merupakan tahap terakhir dalam proses penyisipan. Pada tahap ini dilakukan rekonstruksi ulang menjadi berkas audio dalam format WAV dengan spesifikasi *bit-depth* 16 bit dan *frame rate* 88200 Hz.

### 3.3.2. Proses Ekstraksi

Proses ekstraksi adalah proses untuk mengambil data *payload* yang telah disisipkan dalam berkas *stego-audio*. Dalam proses ini dijelaskan langkah langkah dalam mengekstrak *stego-*

*audio* menggunakan metode yang sama, yakni *fuzzifikasi*. Skema dari proses ekstraksi dapat dilihat pada **Gambar 3.3**.



**Gambar 3.3** Diagram Alir Proses Ekstraksi

### 3.3.2.1. Normalisasi

Dalam tahap ini, dilakukan pengambilan nilai sampel dari data masukan dari proses ekstraksi yang berupa berkas *stego-audio* dalam format \*.wav. Kemudian, dilakukan penambahan semua nilai sampel dengan 32768. Jadi, nilai sampel yang mulanya dengan range -32768 hingga 32767 diubah menjadi dalam *range* 0 hingga 65535. Hasil normalisasi sampel ini yang akan dilanjutkan ke tahap pemecahan sampel.

### 3.3.2.2. Pemecahan Sampel

Setelah tahap normalisasi dilakukan tahap pemecahan sampel. Dalam tahap ini nilai sampel yang telah dinormalisasi dibagi menjadi 2 bagian, yaitu sampel audio asli dan sampel yang telah disisipi *payload* (sampel penyisipan). Sampel audio asli berada pada index genap sedangkan sampel penyisipan berada pada index ganjil. Metode yang digunakan untuk membagi sampel *stego-audio* menjadi 2 bagian tersebut dapat dilihat pada (14) dan (15).

$$S_{n/2} = SSA_n \text{ for } n \text{ is even number} \quad (14)$$

$$S''_{(n-1)/2} = SSA_n \text{ for } n \text{ is odd number} \quad (15)$$

Notasi  $S_n$  menunjukkan sampel audio asli pada index ke- $n$ . Notasi  $S''_n$  menunjukkan sampel ke- $n$  yang disisipkan *payload*. Lalu notasi  $SSA_n$  menunjukkan sampel *stego-audio* pada index ke- $n$ .

### 3.3.2.3. Interpolasi Sample

Dalam tahap ini dilakukan interpolasi sampel untuk menentukan titik baru diantara tiap sampel audio asli hasil keluaran dalam tahap pemecahan sampel secara berurutan. Rumus yang

digunakan sama persis dengan rumus interpolasi (3) pada proses penyisipan.

#### **3.3.2.4. Fuzzifikasi**

Dalam tahap ini dilakukan tahap *fuzzifikasi* nilai sampel keluaran dari tahap interpolasi. Langkah-langkahnya pun sama persis dengan tahap *fuzzifikasi* pada proses penyisipan, yaitu untuk menentukan kategori sampel dari berkas masukan berupa berkas audio yang telah di interpolasi menjadi bentuk *Fuzzy*. Keluaran dari tahap *fuzzifikasi* ini juga terdiri dari 2 nilai, yaitu batas atas dan batas bawah dengan kisaran antara 0 hingga 1. Rumus yang digunakan dapat dilihat pada persamaan (4).

#### **3.3.2.5. Penentuan Ruang Sampel**

Dalam tahap ini juga sama persis dengan yang dilakukan dalam proses penyisipan. sampel yang digunakan adalah keluaran dari tahap *fuzzifikasi* yang berupa 2 nilai yaitu, batas atas dan batas bawah keluaran dari tahap *fuzzifikasi* yang berkisar antara 0 hingga 1. Langkah-langkah dari tahap ini dapat dilihat pada (5), (6), (7), dan (8).

#### **3.3.2.6. Pengambilan *payload***

Dalam tahap ini dilakukan pengambilan data *payload* dari berkas keluaran dari tahap pemecahan sampel, yaitu sampel penyisipan. Dalam tahap pengambilan sampel ini dibagi lagi menjadi 2 tahap, yaitu tahap pengambilan *payload* decimal dan konversi *payload* kedalam bentuk biner.

Tahap pengambilan *payload* decimal adalah proses dimana *payload* diambil dari sampel penyisipan hasil dari tahap pemecahan sampel. Namun *payload* masih dalam bentuk desimal. Masukan dari tahap ini terdiri dari 3 jenis, yaitu sampel penyisipan, sampel interpolasi, dan “total masukan” pada **Tabel 3.1**. Tahap ini

dilakukan pengurangan nilai sampel interpolasi terhadap sampel penyisipan sebanyak “total masukan” sampel. Rumus yang digunakan dapat dilihat pada (16).

$$Pd_n = Si_n - Sp_n \quad (16)$$

Notasi  $Pd_n$  menunjukkan *payload* decimal pada index ke- $n$ . Notasi  $Si_n$  menunjukkan sampel interpolasi pada index ke- $n$ . Sedangkan notasi  $Sp_n$  menunjukkan sampel penyisipan pada index ke- $n$ .

Tahap konversi *payload* dilakukan untuk merubah format *payload* dari bentuk decimal ke bentuk biner. Masukan dari tahap ini terdiri dari 3 jenis, yaitu *payload* decimal, “distribusi *payload*” pada **Tabel 3.1**, dan ruang sampel yang tersedia. Berikut contoh dari tahap ini.

$$\begin{aligned} P &= \{3,2,1,1,0,1\} \\ N &= \{3,2,4,1,0,4\} \\ \text{distribusi} &= 2 \\ \text{Payload}_{\text{segmented}} &= |11|10|01|1||01| \\ \text{Payload} &= 111001101 \end{aligned}$$

Dari contoh diatas, *payload* menunjukkan nilai data *payload* yang telah dikonversi dalam bentuk biner. Notasi  $P$  menunjukkan data *payload* dalam bentuk decimal. Sedangkan notasi  $N$  menunjukkan ruang sampel yang tersedia. Selanjutnya berkas *payload* direkonstruksi menjadi berkas *payload* dalam format \*.txt.

### 3.3.2.7. Denormalisasi Sampel

Pada tahap ini dilakukan proses mengembalikan nilai audio pada *range* seperti sebelum dilakukan normalisasi, yaitu dengan cara pengurangan nilai sampel audio asli dari hasil tahap pemecahan sampel dengan nilai 32768 untuk mengembalikan nilai sampel dalam *range* yang tepat, yaitu -32768 hingga 32767.

### **3.3.2.8. Pembentukan File Audio Asli**

Tahap ini merupakan tahap terakhir dalam proses ekstraksi. Pada tahap ini sampel audio asli hasil keluaran dari tahap denormalisasi akan dilakukan rekonstruksi menjadi berkas audio dalam format \*.wav dengan spesifikasi *channel mono* dan *frame rate* 44100 Hz.

## BAB IV IMPLEMENTASI

Pada bab ini akan membahas tentang bagaimana implementasi yang akan dilakukan dari perancangan yang telah dijelaskan pada bab sebelumnya. Implementasi yang dijelaskan berupa *pseudocode* yang dirancang untuk membangun program dengan menggunakan bahasa pemrograman Python.

### 4.1. Modul Penyisipan

Pada modul ini berisi rancangan dari tiap tahap-tahap yang ada pada proses penyisipan. Rancangan yang dijelaskan berupa *pseudocode* yang dibuat untuk membangun program.

#### 4.1.1. Fungsi *Sampling*

Proses yang dilakukan dalam fungsi ini yaitu membaca berkas audio, kemudian mengambil nilai sampel dari audio. *Pseudocode* dari fungsi *Sampling* bisa dilihat pada **Pseudocode 4.1**.

```
Sampling (nama_file, sampel)
1. set nama_file = "path/to/audio.wav"
2. open nama_file
3. read nama_file
4. get sampel from nama_file
5. add sampel with 32768
6. return sampel
```

#### **Pseudocode 4.1** Membaca Sampel Audio

### 4.1.2. Fungsi Interpolasi Sampel

Pada fungsi ini dilakukan proses interpolasi pada sampel audio. *Pseudocode* dapat dilihat pada **Pseudocode 4.2**.

```

Interpolation (sampel)
1. ganjil = [], genap = []
2. index = sampel.length
3. for x=0 to index-1
4.   set ganjil[x] = 2*x + 1
5. endfor
6. for x=0 to index
7.   genap[x] = 2*x
8. endfor
9. interpolasi = numpy.interp (ganjil, genap, sampel)
10. interpolasi = FLOOR (interpolasi)
11. Return interpolation_sampel

```

#### Pseudocode 4.2 Proses Interpolasi

### 4.1.3. Fungsi Fuzzifikasi

Pada fungsi ini dilakukan proses *fuzzifikasi* pada sampel audio hasil interpolasi. *Pseudocode* dapat dilihat pada **Pseudocode 4.3**.

```

Fuzzifikasi(sampel_interpolasi, sampel_asli)
1. data = [0, 16384, 32768, 49152, 65536]
2. upper = [], lower = []
3. for x=0 to sampel_interpolasi.length
4.   for y=1 to data.length
5.     If data[y-1] <= sampel_interpolasi[x] <
       data[y]
6.       Up = (data[y] - sampel_interpolasi[x])
           /(data[y] - data[y-1])
7.       Low = 1 - up
8.     Endif
9.   Endfor
10. Endfor
11. Return up, low

```



### Pseudocode 4.3 Proses Fuzzifikasi

#### 4.1.4. Fungsi Penentuan Ruang Sampel

Pada fungsi ini dilakukan proses menentukan ruang sampel pada setiap sampel hasil *fuzzifikasi*. *Pseudocode* dapat dilihat pada **Pseudocode 4.4** Proses Penentuan Ruang Sampel **Pseudocode 4.4**.

```

SpaceSample(up, low, sampel_interpolasi, sampel_asli)
1. data : [0, 16384, 32768, 49152, 65536]
2. bit = [], upper = [], lower = []
3. for x=0 to sampel_interpolasi.length
4.   for y=1 to data.length
5.     selisih_up = ABSOLUTE (sampel_interpolasi[x] -
sampel_asli[x+1])
6.     selisih_down = ABSOLUTE (sampel_interpolasi
[x] - sampel_asli[x])
7.     Add (up * selisih_up) to Upper
8.     Add (down * selisih_down) to lower
9.     total = FLOOR (upper[x] + lower[x])
10.    If total = 0
11.      Add 0 to bit
12.    Else
13.      Add FLOOR (LOGBASE2(Total[x])) to bit
14.    Endfor
15. Endfor
16. Return bit

```

#### Pseudocode 4.4 Proses Penentuan Ruang Sampel

#### 4.1.5. Fungsi Segmentasi *Payload*

Proses yang dilakukan dalam fungsi ini adalah untuk melakukan pemecahan data *payload* berdasarkan ruang sampel yang sudah dicari dari tahap sebelumnya. *Pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.5**.

```

Segmentation(payload_biner, sampel_interpolasi, bit,
file_info
1. embedded = []
2. mean_payload = length of payload_biner / length of
  sampel_interpolasi
3. index, proses_payload = append_pl(bit,
  mean_payload, payload_biner
4. if index < payload_biner.length
5.   mean_payload = cek_sisa(payload_biner.length, bit,
  mean_payload)
6.   Index, proses_pl = append_pl(bit, mean_payload,
  payload_biner
7. endif
8. Set file_info = "path/to/file_info.txt"
9. Open file_info
10. Set mean_sample = distribution of payload
11. Set total_append_pl = used_sample.length
12. Close file_info
13. Change payload from binary to decimal
14. Return process_pl

```

#### **Pseudocode 4.5** Proses Segmentasi Payload

##### **4.1.6. Fungsi *Penyisipan***

Proses yang dilakukan dalam fungsi ini adalah menyisipkan data *payload* decimal kedalam ruang sampel audio yang telah ditentukan. *Pseudocode* dapat dilihat pada ***Pseudocode 4.6***.

```

Embedding(sampel_interpolasi, payload_desimal)
1. embedded_sample = []
2. for x=0 to payload_desimal.length
3.   embedded_Sample[x] = sampel_interpolasi[x] -
  payload_desimal[x]
4. Endfor
5. Return embedded_sample

```

## Pseudocode 4.6 Proses Penyisipan Payload

### 4.1.7. Fungsi Penggabungan

Proses yang dilakukan pada fungsi ini adalah untuk menggabungkan sampel audio asli dengan sampel hasil penyisipan *payload* menjadi satu sampel audio. *pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.7**.

```

Penggabungan(original_sample, embed_data,
sampel_interpolasi )
1. new_embed_data = []
2. for x=0 to sampel_interpolasi.length
3.   if x < length of embed_data
4.     new_embed_data [x] = embed_data[x]
5.   else
6.     new_embed_data [x] = sampel_interpolasi[x]
7.   endif
8. Endfor
9. stego_data = []
10. index_stego = 0
11. index_sample = 0
12. index_embed = 0
13. for x=0 to length of (original_sampel*2)-1
14.   if x = even number
15.     stego_data[x] = original_sampel
        [index_sample1]
16.     index_sample = index_sample + 1
17.   else
18.     stego_data[x] = new_embed_data [index_embed]
19.     index_embed = index_embed + 1
20.   Endif
21.   Index_stego = index_stego + 1
22. Endfor
23. Return stego_data

```

## Pseudocode 4.7 Proses Penggabungan Sampel Audio

### 4.1.8. Proses Pembuatan Stego-audio

Proses yang dilakukan dalam fungsi ini adalah merekonstruksi sampel hasil penggabungan menjadi berkas audio. Spesifikasi audio yang dihasilkan memiliki jumlah sampel 2 kali lebih banyak karena menggunakan metode interpolasi linier. Namun *frame rate* yang dihasilkan sebesar 88200 Hz hal tersebut dapat membuat durasi audio nya tidak memiliki perbedaan dengan audio asli. *pseudocode* dari fungsi ini dapat dilihat pada ***Pseudocode 4.8***.

1. variable : stego\_sample, stego\_audio
2. subtract stego\_sample with 32768
3. change datatype of stego\_sample to integer 16 bit
4. create stego\_audio from stego\_sample with frame rate 88200 Hz.

#### **Pseudocode 4.8** Prose Pembuatan Stego-audio

## 4.2. Modul Ekstraksi

Pada modul ini berisi rancangan dari tiap tahap-tahap yang ada pada proses ekstraksi. Rancangan yang dijelaskan berupa *pseudocode* yang dibuat untuk membangun program.

### 4.2.1. Fungsi *Sampling*

Proses yang dilakukan dalam fungsi ini yaitu membaca berkas *stego-audio*, kemudian mengambil nilai sampel dari *stego-audio* tersebut. *Pseudocode* dari fungsi *Sampling* bisa dilihat pada ***Pseudocode 4.9***.

- ```
Sampling (nama_file, stego_sample)
1. set nama_file = "path/to/stego_audio.wav"
2. open nama_file
3. read nama_file
4. get stego_sample from nama_file
5. add stego_sample with 32768
6. return stego_sample
```

### **Pseudocode 4.9** Proses Membaca Sampel Audio

#### **4.2.2. Fungsi Pemecahan Sampel**

Proses yang dilakukan dalam fungsi ini adalah membagi sampel *stego-audio* menjadi 2 sampel, yaitu sampel audio asli dan sampel audio dengan *payload*. *Pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.10**.

```

Pemecahan (stego_sample, cover_sampel)
1. cover_sampel = [], stego_sample = []
2. for x=0 to n
3.   if x is odd number
4.     Add (stego_sample[x]) to cover_sample
5.   else
6.     Add (stego_sample[x]) to embed_sample
7.   Endif
8. Endfor
9. Return cover_sample, embed_sample

```

### **Pseudocode 4.10** Proses Pemecahan Sampel Audio

#### **4.2.3. Fungsi Interpolasi Sampel**

Proses yang dilakukan pada fungsi ini sama persis dengan fungsi interpolasi sampel yang dilakukan pada modul penyisipan, yaitu dengan membuat titik baru dengan metode interpolasi. *Pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.2**.

#### **4.2.4. Fungsi Fuzzifikasi**

Proses yang dilakukan pada fungsi ini sama persis dengan fungsi *fuzzifikasi* yang dilakukan pada modul penyisipan. *Pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.3**.

#### **4.2.5. Fungsi Penentuan Ruang Sampel**

Proses yang dilakukan pada fungsi ini sama persis dengan fungsi penentuan ruang sampel yang dilakukan pada modul penyisipan. *Pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.4**.

#### 4.2.6. Fungsi Pengambilan *Payload*

Proses yang dilakukan pada fungsi ini adalah untuk mengambil *payload* dari “sampel dengan *payload*” hasil dari proses pemecahan sampel. *Pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.11**.

```
Pengambilan (payload_desimal, ruang_sample, dist_pl,
payload)
1. for x = 0 to payload_desimal.length
2.   if ruang_sample[x] = 0
3.     Continue
4.   If ruang_Sample[x] < dist_pl
5.     payload[x] = change decimal_payload[x] to binary
with total bit of ruang_sample[x]
6.   if ruang_Sample[x] >= dist_pl
7.     Payload[x] = change decimal_payload[x] to binary
with total bit of dist_pl
8. Endfor
9. Return payload
```

#### **Pseudocode 4.11** Proses Pengambilan Payload

#### 4.2.7. Fungsi Rekonstruksi Audio

Proses yang dilakukan pada fungsi ini adalah untuk merekonstruksi sampel audio asli. Serta, mengembalikan *frame rates* pada 44100 Hz. *Pseudocode* dari fungsi ini dapat dilihat pada **Pseudocode 4.12**.

1. variable : sample, original\_audio
2. subtract sample with 32768
3. change datatype of sample to integer 16 bit
4. create original\_audio from sample with frame rate 44100 Hz

***Pseudocode 4.12 Proses Rekonstruksi Audio***

### 4.3. Modul Analisis

Modul ini berisi fungsi yang digunakan untuk menganalisa hasil uji coba. *Library* yang digunakan pada modul ini adalah *sklearn.metrics* dan Numpy. Fungsi yang digunakan adalah MSE dan PSNR.

#### 4.3.1. Fungsi MSE

Fungsi MSE ini menggunakan *library* dari *sklearn.metrics*. *Pseudocode* dari fungsi MSE dapat dilihat pada **Pseudocode 4.13**.

```
MSE (original_sampel, stego_sampel, mse, original_audio,
stego_audio)
1. open original_audio
2. read original_audio
3. get original_sampel from original_audio
4. open stego_audio
5. read stego_audio
6. get stego_sampel from stego_audio
7. mse = mean.squared.error (original_sampel,
stego_sampel)
```

**Pseudocode 4.13 MSE**

### 4.3.2. Fungsi PSNR

*Pseudocode* fungsi PSNR dapat dilihat pada *Pseudocode*

#### 4.14.

```
PSNR (original_sampel, stego_sampel, mse, psnr)
1. mse = mean.squared.error (original_sampel,
    stego_sampel)
2. psnr = 10 * LOG (POWER ((POWER (2, 16) - 1), 2) /
    mse)
```

#### **Pseudocode 4.14 PSNR**



## BAB V UJI COBA DAN EVALUASI

Pada bab ini dilakukan uji coba dan evaluasi sistem yang sudah dibangun sesuai dengan rumusan masalah dan implementasi dari bab sebelumnya. Setelah mendapatkan hasil dari uji coba tersebut akan dilakukan evaluasi sehingga dapat ditarik kesimpulan yang akan dijelaskan pada bab berikutnya.

### 5.1. Lingkungan Uji Coba

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk pengujian ditunjukkan pada **Tabel 5.1**.

**Tabel 5.1** Lingkungan Uji Coba

| Perangkat       | Jenis Perangkat      | Spesifikasi                         |
|-----------------|----------------------|-------------------------------------|
| Perangkat Keras | Prosesor             | Intel® Core™ i5-5200U CPU @ 2.20GHz |
|                 | Memori               | 6 Gb                                |
| Perangkat Lunak | Sistem Operasi       | Windows 10 Pro                      |
|                 | Perangkat Pengembang | Visual Studio Code Version: 1.44.2  |
|                 | Bahasa Pemrograman   | Python 3.8.0 64-bit                 |

### 5.2. Data Pengujian

Pada subbab ini akan dijelaskan semua data yang akan digunakan untuk pengujian pada Tugas Akhir ini.

#### 5.2.1. Cover Audio

*Cover Audio* adalah berkas audio yang akan digunakan sebagai media penyisipan *payload*. Spesifikasi *cover audio* yang digunakan dalam uji coba ini memiliki format *\*.wav, channel*

*mono*, dan *bit-depth* yaitu 16 bit. Pada uji coba ini akan menggunakan 15 *cover* audio dengan jenis suara yang bervariasi. Detil audio dapat dilihat pada **Tabel 5.2**.

**Tabel 5.2** Detil *Cover* Audio

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

### 5.2.2. *Payload*

*Payload* merupakan data bersifat rahasia yang akan disisipkan kedalam file *cover* audio. Spesifikasi *payload* yang akan digunakan dalam uji coba ini memiliki format \*.txt. Pada uji coba

ini akan menggunakan 11 *payload* dengan jenis ukuran yang bervariasi. Detil *payload* dapat dilihat pada **Tabel 5.3**.

**Tabel 5.3** Detil *Payload*

| No. | Nama Payload  | Ukuran File (kb) |
|-----|---------------|------------------|
| 1   | Payload1.txt  | 1                |
| 2   | Payload2.txt  | 10               |
| 3   | Payload3.txt  | 20               |
| 4   | Payload4.txt  | 30               |
| 5   | Payload5.txt  | 40               |
| 6   | Payload6.txt  | 50               |
| 7   | Payload7.txt  | 60               |
| 8   | Payload8.txt  | 70               |
| 9   | Payload9.txt  | 80               |
| 10  | Payload10.txt | 90               |
| 11  | Payload11.txt | 100              |

### 5.3. Skenario Uji Coba dan Evaluasi Pengujian

Skenario uji coba dilakukan untuk menguji kebenaran dari hasil rancangan dan implementasi. Uji coba akan dilakukan berdasarkan skenario yang dibuat. Penentuan skenario untuk uji coba adalah :

1. Perbedaan kualitas *stego-audio* berdasarkan nilai *Means Square Error* (MSE) dan *Peak Signal-to-Noise Ratio* (PSNR).
2. Perbedaan jumlah penggunaan sampel untuk melakukan penyisipan

#### 5.3.1. Skenario Uji Coba 1

Skenario yang akan dilakukan pada uji coba 1 adalah mengukur kualitas *stego-audio* menggunakan nilai *Mean Squared Error* (MSE) dan *Peak Signal-to-Noise Ratio* (PSNR). Skenario uji coba 1 dilakukan perbandingan *cover* audio asli yang sudah diinterpolasi dengan *stego-audio* untuk mengetahui kualitas dari *stego-audio* berdasarkan nilai PSNR yang dihasilkan. kemudian, dilakukan perbandingan dengan metode usulan dari Ahmad dan

Fiqar [8] yang telah di smoothing sebanyak 5 kali pada penelitian sebelumnya. Hasil perhitungan PSNR dapat dilihat pada **Tabel 5.4** dan **Tabel 5.5**.

### 5.3.2. Evaluasi Uji Coba 1

Berdasarkan hasil pengujian pada skenario didapatkan hasil perbandingan kualitas *stego-audio* yang diukur dengan PSNR dari 2 metode, yaitu metode yang diusulkan oleh Ahmad dan Fiqar [8] pada penelitian sebelumnya dan metode usulan dari makalah ini.

Informasi yang didapat dari hasil hasil tersebut yaitu semakin besar ukuran file *payload* maka semakin rendah kualitas *stego-audio*. Hal tersebut mengakibatkan kesamaan antara *stego-audio* dan audio asli berbeda jauh. Sedangkan kriteria yang diharapkan adalah semakin tinggi nilai PSNR maka semakin bagus kualitas *stego-audio*.

Berdasarkan hasil pengujian metode usulan Ahmad dan Fiqar [8] menunjukkan bahwa hasil PSNR terbaik diperoleh pada audio 13 dengan ukuran *payload* 1 kb dengan hasil 119 db dan hasil PSNR terburuk diperoleh pada audio 3 dengan ukuran *payload* 100 kb dengan hasil 51,1373 db. Sedangkan pada metode usulan, kualitas PSNR terbaik diperoleh pada semua audio dengan ukuran *payload* 1 kb dengan hasil 123.67 db dan PSNR terburuk diperoleh pada semua audio dengan ukuran *payload* 100 kb sebesar 103.56 db. Hal tersebut dikarenakan nilai PSNR dipengaruhi oleh perhitungan nilai MSE yang menunjukkan hasil yang sama pada semua audio untuk setiap *payload*.

Dari hasil perhitungan data MSE menunjukkan setiap audio memiliki nilai MSE yang sama untuk setiap *payload*. Hal tersebut dikarenakan metode usulan ini didukung dengan distribusi *payload*. Sehingga memungkinkan seluruh sampel digunakan dalam proses penyisipan sesuai banyaknya bit yang didistribusikan dan ruang sampel yang tersedia serta memungkinkan setiap sampel memiliki porsi bit yang sama untuk penyisipan *payload*.

**Tabel 5.4.** Hasil Perbandingan PSNR

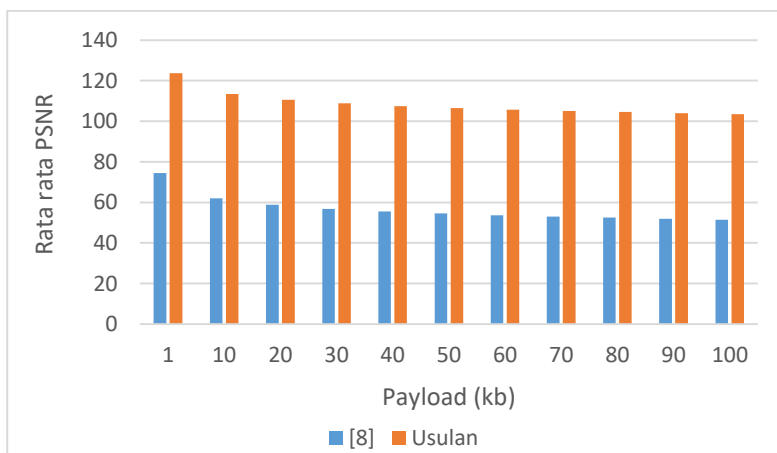
| Audio | Payload 1 |        | Payload 2 |        | Payload 3 |        | Payload 4 |        | Payload 5 |        | Payload 6 |        |
|-------|-----------|--------|-----------|--------|-----------|--------|-----------|--------|-----------|--------|-----------|--------|
|       | [8]       | Usulan | [8]       | Usulan | [8]       | Usulan | [8]       | Usulan | [8]       | Usulan | [8]       | Usulan |
| 1     | 71.3      | 123.67 | 61.19     | 113.5  | 58.11     | 110.55 | 56.42     | 108.8  | 55.21     | 107.5  | 54.22     | 106.6  |
| 2     | 72.38     | 123.67 | 62.55     | 113.5  | 60.3      | 110.55 | 58        | 108.8  | 56.77     | 107.5  | 55.83     | 106.6  |
| 3     | 70.92     | 123.67 | 61.05     | 113.5  | 58.26     | 110.55 | 56.45     | 108.8  | 55.12     | 107.5  | 54.13     | 106.6  |
| 4     | 71.64     | 123.67 | 61.11     | 113.5  | 58.21     | 110.55 | 56.38     | 108.8  | 55.24     | 107.5  | 54.33     | 106.6  |
| 5     | 71.13     | 123.67 | 61.19     | 113.5  | 58.44     | 110.55 | 56.6      | 108.8  | 55.34     | 107.5  | 54.33     | 106.6  |
| 6     | 71.29     | 123.67 | 61.39     | 113.5  | 58.49     | 110.55 | 56.63     | 108.8  | 55.45     | 107.5  | 54.37     | 106.6  |
| 7     | 71.24     | 123.67 | 61.29     | 113.5  | 58.43     | 110.55 | 56.77     | 108.8  | 55.28     | 107.5  | 54.42     | 106.6  |
| 8     | 71.15     | 123.67 | 61.28     | 113.5  | 58.11     | 110.55 | 56.48     | 108.8  | 55.25     | 107.5  | 54.26     | 106.6  |
| 9     | 71.32     | 123.67 | 61        | 113.5  | 58.33     | 110.55 | 56.3      | 108.8  | 55.22     | 107.5  | 54.21     | 106.6  |
| 10    | 71.25     | 123.67 | 61.02     | 113.5  | 58.29     | 110.55 | 56.54     | 108.8  | 55.23     | 107.5  | 54.23     | 106.6  |
| 11    | 71.27     | 123.67 | 62        | 113.5  | 59.43     | 110.55 | 57.63     | 108.8  | 56.27     | 107.5  | 55.29     | 106.6  |
| 12    | 70.77     | 123.67 | 61.2      | 113.5  | 58.14     | 110.55 | 56.53     | 108.8  | 55.18     | 107.5  | 54.17     | 106.6  |
| 13    | 119       | 123.67 | 71.38     | 113.5  | 63.22     | 110.55 | 59.02     | 108.8  | 57.24     | 107.5  | 55.8      | 106.6  |
| 14    | 70.99     | 123.67 | 61.21     | 113.5  | 58.07     | 110.55 | 56.38     | 108.8  | 55.14     | 107.5  | 54.18     | 106.6  |
| 15    | 72.07     | 123.67 | 61.13     | 113.5  | 58.22     | 110.55 | 56.32     | 108.8  | 55.3      | 107.5  | 54.14     | 106.6  |

**Tabel 5.5** Hasil Perbandingan PSNR

| Audio | Payload 7 |        | Payload 8 |        | Payload 9 |        | Payload 10 |        | Payload 11 |        |
|-------|-----------|--------|-----------|--------|-----------|--------|------------|--------|------------|--------|
|       | [8]       | Usulan | [8]       | Usulan | [8]       | Usulan | [8]        | Usulan | [8]        | Usulan |
| 1     | 53.46     | 105.76 | 52.73     | 105.1  | 52.26     | 104.57 | 51.73      | 104    | 51.24      | 103.6  |
| 2     | 55.07     | 105.76 | 54.45     | 105.1  | 53.78     | 104.57 | 53.2       | 104    | 52.79      | 103.6  |
| 3     | 53.27     | 105.76 | 52.7      | 105.1  | 52.25     | 104.57 | 51.61      | 104    | 51.14      | 103.6  |
| 4     | 53.28     | 105.76 | 52.8      | 105.1  | 52.2      | 104.57 | 51.77      | 104    | 51.28      | 103.6  |
| 5     | 53.69     | 105.76 | 52.93     | 105.1  | 52.4      | 104.57 | 51.83      | 104    | 51.47      | 103.6  |
| 6     | 53.62     | 105.76 | 52.86     | 105.1  | 52.36     | 104.57 | 51.88      | 104    | 51.44      | 103.6  |
| 7     | 53.5      | 105.76 | 52.95     | 105.1  | 52.3      | 104.57 | 51.66      | 104    | 51.22      | 103.6  |
| 8     | 53.5      | 105.76 | 52.85     | 105.1  | 52.38     | 104.57 | 51.73      | 104    | 51.29      | 103.6  |
| 9     | 53.39     | 105.76 | 52.79     | 105.1  | 52.26     | 104.57 | 51.6       | 104    | 51.18      | 103.6  |
| 10    | 53.44     | 105.76 | 52.89     | 105.1  | 52.26     | 104.57 | 51.65      | 104    | 51.27      | 103.6  |
| 11    | 54.41     | 105.76 | 53.74     | 105.1  | 53.08     | 104.57 | 52.55      | 104    | 52.06      | 103.6  |
| 12    | 53.33     | 105.76 | 52.7      | 105.1  | 52.25     | 104.57 | 51.64      | 104    | 51.14      | 103.6  |
| 13    | 54.65     | 105.76 | 53.87     | 105.1  | 53.15     | 104.57 | 52.4       | 104    | 51.91      | 103.6  |
| 14    | 53.39     | 105.76 | 52.78     | 105.1  | 52.23     | 104.57 | 51.62      | 104    | 51.17      | 103.6  |
| 15    | 53.28     | 105.76 | 52.78     | 105.1  | 52.18     | 104.57 | 51.66      | 104    | 51.2       | 103.6  |

Dari dataset yang telah diuji menunjukkan bahwa metode distribusi *payload* hanya mendistribusikan 1 bit di setiap sampel. Hal tersebut dikarenakan dari dataset audio yang digunakan menunjukkan bahwa jumlah sampel audio selalu lebih banyak daripada jumlah bit dari setiap berkas *payload*. Maka dari itu seluruh *payload* hanya mendistribusikan 1 bit di setiap nilai sampelnya. Berhubung berkas *payload* merupakan file binary, Maka kemungkinan bit yang ditampung pada setiap sampel antara 1 atau 0. Hasil tersebut mempengaruhi hasil perhitungan MSE, yaitu selisih antara file *stego-audio* dan audio asli bergantung pada banyaknya angka 1 pada file *payload*. Maka dari itu hasil MSE pada semua audio untuk setiap *payload* menunjukkan hasil yang sama. Begitu juga dengan hasil PSNR yang dihasilkan.

Namun meski nilai MSE dan PSNR pada semua audio untuk setiap *payload* menunjukkan nilai yang sama, jumlah sampel yang digunakan untuk penyisipan berbeda. Proses pendistribusian *payload* tetap mengacu pada ruang sampel yang tersedia. Total sampel yang digunakan setiap *payload* dalam proses penyisipan dapat dilihat pada hasil uji coba skenario ke 2.



**Gambar 5.1** Hasil Perbandingan PSNR

Secara keseluruhan metode usulan dapat menghasilkan kualitas stego-audio lebih baik daripada metode metode yang diusulkan oleh Ahmad dan Fiqar [8] tanpa harus dilakukan smoothing. Hal tersebut ditunjukkan dengan adanya Peningkatan hasil PSNR hingga 48 persen. Grafik hasil perbandingan PSNR dapat dilihat pada **Gambar 5.1**.

### 5.3.3. Skenario Uji Coba 2

Skenario yang akan dilakukan pada uji coba 2 adalah menghitung jumlah sampel yang digunakan dalam proses penyisipan *payload*. Hal tersebut untuk membuktikan bahwa meskipun nilai MSE dan nilai PSNR pada metode usulan memiliki nilai yang sama pada setiap audio untuk setiap *payload*, Namun jumlah sampel yang digunakan dalam penyisipan berbeda. Hal tersebut dikarenakan metode usulan didukung dengan metode distribusi *payload* sesuai ruang sampel yang tersedia dan bit *payload* yang didistribusikan. Kemudian dilakukan perbandingan dengan hasil total penggunaan sampel tanpa menggunakan distribusi *payload*. Hasil perhitungan total penggunaan sampel dengan distribusi *payload* dapat dilihat pada **Tabel 5.6**. Sedangkan hasil perhitungan total penggunaan sampel tanpa distribusi *payload* dapat dilihat pada **Tabel 5.7**.

### 5.3.4. Evaluasi Uji Coba 2

Hasil pengujian pada **Tabel 5.6** merupakan hasil total sampel yang digunakan pada metode usulan disertai dengan distribusi *payload*. Dari informasi yang diperoleh pada **Tabel 5.6** adalah terdapat perbedaan total sampel yang digunakan dalam proses penyisipan meskipun nilai MSE dan PSNR menunjukkan hasil yang sama pada semua audio untuk setiap *payload*.



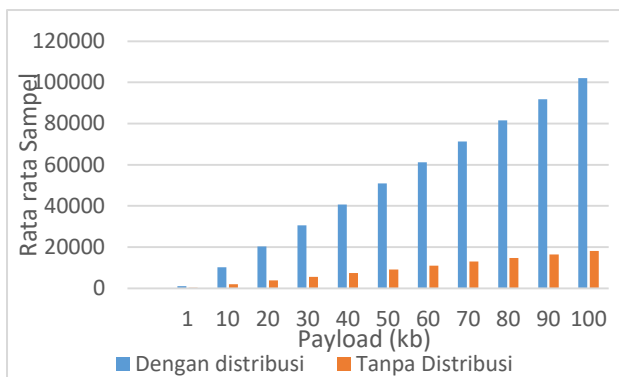
**Tabel 5.6** Total Penggunaan Sampel pada Distribusi Payload

| audio | Payload (kb) |       |       |       |       |       |       |       |       |       |        |
|-------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
|       | 1            | 10    | 20    | 30    | 40    | 50    | 60    | 70    | 80    | 90    | 100    |
| 1     | 1008         | 10068 | 20122 | 30149 | 40200 | 50309 | 60381 | 70416 | 80491 | 90549 | 100581 |
| 2     | 1064         | 10787 | 21620 | 32231 | 42721 | 53705 | 65363 | 76314 | 87520 | 98685 | 109934 |
| 3     | 1001         | 10007 | 20011 | 30020 | 40029 | 50038 | 60046 | 70057 | 80065 | 90072 | 100073 |
| 4     | 1006         | 10060 | 20135 | 30207 | 40264 | 50324 | 60404 | 70477 | 80542 | 90646 | 100758 |
| 5     | 1018         | 10165 | 20378 | 30604 | 40929 | 51346 | 63229 | 73369 | 83669 | 94220 | 104763 |
| 6     | 1023         | 10172 | 20365 | 30489 | 40589 | 50730 | 60891 | 71059 | 81219 | 91297 | 101370 |
| 7     | 1024         | 10130 | 20288 | 30496 | 40740 | 51340 | 61397 | 71558 | 81645 | 91867 | 102185 |
| 8     | 1012         | 10168 | 20370 | 30521 | 40675 | 50821 | 61027 | 71175 | 81509 | 92013 | 102352 |
| 9     | 1004         | 10029 | 20067 | 30105 | 40128 | 50157 | 60186 | 70227 | 80253 | 90289 | 100332 |
| 10    | 1010         | 10079 | 20140 | 30188 | 40254 | 50338 | 60405 | 70461 | 80528 | 90615 | 100667 |
| 11    | 1058         | 10485 | 20923 | 31306 | 41725 | 52192 | 62606 | 72944 | 83207 | 93537 | 103903 |
| 12    | 1006         | 10035 | 20069 | 30142 | 40197 | 50244 | 60282 | 70339 | 80399 | 90477 | 100560 |
| 13    | 1325         | 10809 | 20980 | 31051 | 41105 | 51205 | 61437 | 71608 | 81792 | 91976 | 102061 |
| 14    | 1005         | 10023 | 20051 | 30086 | 40123 | 50164 | 60207 | 70248 | 80280 | 90299 | 100327 |
| 15    | 1005         | 10042 | 20094 | 30123 | 40140 | 50167 | 60187 | 70212 | 80237 | 90263 | 100285 |

**Tabel 5.7.** Total Penggunaan Sampel Tanpa Distribusi Payload

| audio | Payload (kb) |      |      |       |       |       |       |       |       |       |       |
|-------|--------------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|
|       | 1            | 10   | 20   | 30    | 40    | 50    | 60    | 70    | 80    | 90    | 100   |
| 1     | 151          | 1545 | 3132 | 4767  | 6429  | 7949  | 9433  | 10950 | 12361 | 13827 | 15342 |
| 2     | 315          | 3336 | 6790 | 10113 | 13449 | 16875 | 20255 | 23567 | 26702 | 29729 | 32520 |
| 3     | 107          | 1082 | 2165 | 3218  | 4287  | 5352  | 6422  | 7525  | 8578  | 9646  | 10697 |
| 4     | 164          | 1633 | 3340 | 4746  | 6099  | 7564  | 9191  | 10754 | 12400 | 14062 | 15568 |
| 5     | 213          | 2090 | 4146 | 6379  | 8629  | 10797 | 13122 | 15415 | 17682 | 20198 | 22199 |
| 6     | 216          | 2086 | 3939 | 5854  | 7833  | 9823  | 11984 | 13960 | 15947 | 17984 | 20091 |
| 7     | 219          | 2093 | 3576 | 5079  | 6714  | 8561  | 10700 | 13207 | 14867 | 16301 | 17835 |
| 8     | 194          | 1811 | 3693 | 5682  | 7872  | 10033 | 12067 | 13916 | 15913 | 18053 | 20073 |
| 9     | 144          | 1437 | 2925 | 4425  | 5769  | 7034  | 8366  | 9751  | 11153 | 12519 | 13743 |
| 10    | 165          | 1635 | 3207 | 4839  | 6486  | 8064  | 9619  | 11157 | 12646 | 14182 | 15714 |
| 11    | 290          | 2988 | 5827 | 8494  | 11387 | 14052 | 16766 | 19525 | 22145 | 24814 | 27388 |
| 12    | 123          | 1295 | 2600 | 3912  | 5285  | 6697  | 8086  | 9519  | 10886 | 12262 | 13797 |
| 13    | 695          | 4334 | 6232 | 8020  | 9894  | 11886 | 14418 | 16579 | 17897 | 19185 | 20492 |
| 14    | 138          | 1330 | 2689 | 3977  | 5256  | 6478  | 7673  | 9013  | 10252 | 11466 | 12672 |
| 15    | 152          | 1331 | 2595 | 3906  | 5260  | 6650  | 8067  | 9531  | 11035 | 12534 | 14051 |

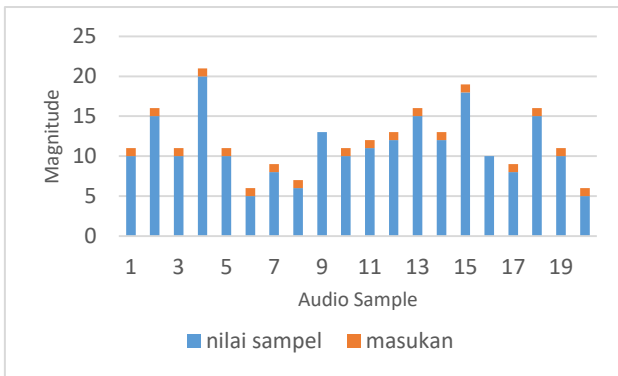
Sedangkan Tabel **Tabel 5.7** merupakan hasil total sampel yang digunakan pada metode usulan namun tidak disertai dengan distribusi payload. Informasi yang diperoleh pada tabel **Tabel 5.7** adalah terdapat perbedaan jumlah sampel yang digunakan dalam proses penyisipan. Namun, jumlah yang digunakan lebih sedikit daripada hasil yang ditunjukkan pada tabel **Tabel 5.6**.



**Gambar 5.2.** Perbandingan Total Penggunaan Sampel Dengan dan Tanpa Distribusi Payload

Informasi yang dapat diambil dari 2 hasil perbandingan sampel tersebut adalah semakin besar ukuran berkas *payload* maka penggunaan sampel untuk penyisipan juga semakin banyak (**Gambar 5.2**). Namun, metode usulan dengan disertai distribusi payload menunjukkan penggunaan sampel lebih banyak dalam proses penyisipan. Hal tersebut dapat memungkinkan seluruh sampel digunakan dalam proses penyisipan serta memiliki porsi bit *payload* yang sama untuk disisipkan. Sehingga dapat dipastikan seluruh sampel dapat disisipkan kedalam audio. Hal tersebut terbukti dapat meningkatkan kualitas dari *stego-audio*. Hasil perbandingan total sampel antara dengan dan tanpa metode distribusi payload dapat dilihat pada **Gambar 5.2**.

Namun metode distribusi *payload* yang diusulkan pada Tugas Akhir ini tidak sepenuhnya melakukan distribusi *payload* secara merata. Distribusi yang dilakukan tetap memperhatikan ruang sampel yang tersedia. Hal tersebut dikarenakan setiap sampel memiliki kapasitas yang berbeda beda untuk menampung bit *payload*. Ilustrasi metode distribusi *payload* yang diusulkan pada Tugas Akhir ini dapat dilihat pada gambar **Gambar 5.3**.

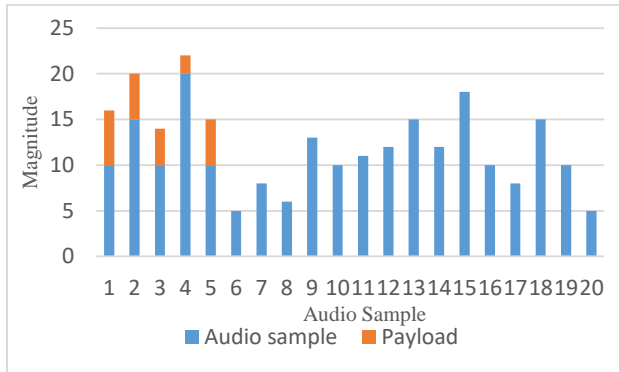


**Gambar 5.3.** Metode Distribusi Payload

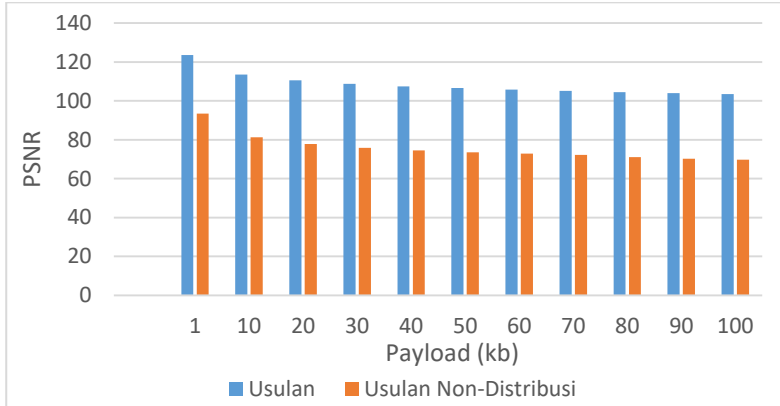
Informasi yang didapat dari **Gambar 5.3** adalah setiap sampel mempunyai porsi yang sama dalam menampung bit *payload* yang didistribusi kecuali sampel yang memiliki ruang sampel lebih kecil daripada jumlah bit yang didistribusikan seperti sampel nomor 9 dan nomor 16 pada **Gambar 5.3**. Sampel tersebut tidak dapat menampung bit yang didistribusikan karena sampel tersebut memiliki ruang sampel lebih kecil dari jumlah bit yang didistribusikan.

Sedangkan hasil pengujian metode usulan tanpa disertai metode distribusi *payload* menunjukkan bahwa penyisipan dilakukan dengan memaksimalkan ruang sampel yang tersedia. Sehingga sampel yang digunakan dalam penyisipan hanya 5 sampel pertama. Sisanya tidak digunakan dalam penyisipan karena bit *payload* sudah habis disisipkan hanya pada 5 sampel pertama.

Sehingga diperlukan tahap *smoothing* untuk memperbaiki nilai PSNR dari metode tersebut. Ilustrasi metode usulan tanpa disertai metode distribusi *payload* yang telah diuji coba dapat dilihat pada **Gambar 5.4.**



**Gambar 5.4.** Tanpa Distribusi Payload



**Gambar 5.5.** Hasil Perbandingan PSNR dengan dan tanpa Distribusi Payload

Secara keseluruhan metode usulan dengan disertai distribusi *payload* dapat menghasilkan kualitas stego-audio lebih

baik daripada metode usulan tanpa disertai distribusi *payload*. Hal tersebut ditunjukkan dengan adanya Peningkatan hasil PSNR sebesar 30 persen. Grafik hasil perbandingan PSNR dari metode dengan dan tanpa distribusi *payload* dapat dilihat pada **Gambar 5.5**.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dibahas mengenai kesimpulan yang dapat diambil dari tujuan pembangunan perangkat lunak dengan hasil uji coba yang telah dilakukan. Hasil uji coba ini digunakan untuk menjawab rumusan masalah yang telah dikemukakan. Selain itu juga ada saran yang bertujuan untuk memberikan tanggapan apabila ada pengembangan dari sistem ini lebih lanjut.

#### **6.1. Kesimpulan**

Kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan pada bab sebelumnya adalah sebagai berikut:

1. Fuzzy Logic telah diimplementasikan dengan cara memanfaatkan tahap fuzzifikasi untuk menentukan berapa banyak bit yang dapat ditampung dalam suatu sampel audio.
2. Metode usulan dapat menghasilkan kualitas *stego-audio* lebih baik daripada metode yang diusulkan oleh Ahmad dan Fiqar [8] pada penelitian sebelumnya. Hal tersebut ditunjukkan dengan adanya peningkatan nilai PSNR sebesar 48 persen tanpa harus dilakukan tahap *smoothing*.
3. Penggunaan distribusi payload pada metode usulan memungkinkan semua sampel digunakan dalam proses penyisipan serta dapat dipastikan semua payload berhasil disisipkan dan diekstraksi tanpa adanya kesalahan.

#### **6.2. Saran**

Saran yang diberikan dari hasil uji coba dan evaluasi yang telah dilakukan untuk pengembangan sistem selanjutnya adalah sebagai berikut:

1. Adanya metode penyisipan dengan cara mengembangkan baik metode distribusi *payload* atau metode *smoothing* pada penelitian sebelumnya.
2. Adanya metode yang dapat meningkatkan kualitas dan kuantitas dari *stego-audio*.



## DAFTAR PUSTAKA

- [1] H. Nasution, "Implementasi Logika Fuzzy pada Sistem Kecerdasan Buatan," *Jurnal ELKHA*, vol. 4, no. 2, 2012.
- [2] R. A. Septiawan, Implementasi Logika Fuzzy Mamdani Untuk Menentukan Harga Gabah, Semarang: Teknik Informatika Universitas Dian Nuswantoro, 2009.
- [3] L. W. Trimartanti, Penerapan Sistem Fuzzy Untuk Diagnosis Campuran Bahan Bakar dan Udara Pada Mobil F15 Gurt, Yogyakarta: Universitas Negeri Yogyakarta, 2016.
- [4] M. H. A. Al-Hooti, S. Djanali and T. Ahmad, "Audio Data Hiding Based on Sample Value Modification," *Journal of Information Processing System*, 2015.
- [5] python.org, "What is python? Executive Summary," [Online]. Available: <https://www.python.org/doc/essays/blurb/>. [Accessed 03 April 2020].
- [6] code.visualstudio.com, "Get Started," [Online]. Available: <https://code.visualstudio.com/docs>. [Accessed 03 April 2020].
- [7] A. A. Alghamdi, "Computerized Steganographic Technique using Fuzzy Logic," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 3, pp. 155-159, 2018.
- [8] T. Ahmad and T.P.Fiqar, "Enhancing the Performance of Audio Data Hiding Method by Smoothing Interpolated Samples," *International Journal of Information and Control*, vol. 7, pp. 1-10, 2011.
- [9] R. F. Cadiz, "Fuzzy logic in the arts: Applications in audiovisual composition and sound synthesis," in *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, 2005.

## **BIODATA PENULIS**



Penulis dilahirkan di Sidoarjo, 9 April 1998 dengan nama lengkap Moh. Muzayyin Amrulloh. Penulis menempuh pendidikan formal di SD Punggul 1 Sidoarjo, SMPN 3 Peterongan Jombang, SMA Darul Ulum 1 Unggulan BPP-Teknologi Jombang. Kemudian penulis diterima sebagai mahasiswa Departemen Informatika ITS pada tahun 2016. Selama masa perkuliahan, penulis aktif dalam acara Schematics 2017 dan 2018 sebagai staff Humas, Himpunan Mahasiswa Teknik Computer - Informatika sebagai staff Dagri, Keluarga Muslim Informatika sebagai staff departemen Keilmuan pada periode 2018-2019. Penulis juga aktif sebagai admin Laboratorium Komputasi Berbasis Jaringan mulai periode 2018. Selain itu penulis juga pernah melakukan magang di DPTSI (Direktorat Pengembangan Teknologi dan Sistem Informasi) ITS Surabaya. Bagi pembaca yang ingin berdiskusi, memberikan saran, dan/atau kritik mengenai Tugas Akhir ini dapat disampaikan melalui email [muzayyin937@gmail.com](mailto:muzayyin937@gmail.com) atau melalui nomor 085733643715.