

**PENYISIPAN PESAN DALAM FILE AUDIO DENGAN
METODE *SPREAD SPECTRUM* DAN N-LSB**

SKRIPSI

**OLEH
MUHAMMAD AGUS KHIFDON YUWAFI
NIM. 220601110012**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026**

**PENYISIPAN PESAN DALAM FILE AUDIO DENGAN
METODE *SPREAD SPECTRUM* DAN N-LSB**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh
Muhammad Agus Khifdon Yuwafi
NIM. 220601110012**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026**

**PENYISIPAN PESAN DALAM FILE AUDIO DENGAN
METODE *SPREAD SPECTRUM* DAN N-LSB**


SKRIPSI

Oleh
Muhammad Agus Khifdon Yuwafi
NIM. 220601110012

Telah Disetujui Untuk Diuji

Malang, 7 April 2026

Dosen Pembimbing I


Muhammad Khudzaifah, M.Si
NIP. 19700511 202321 1 029

Dosen Pembimbing II


Dr. Ach. Nashichuddin, M.A
NIP. 19730705 200003 1 002



**PENYISIPAN PESAN DALAM FILE AUDIO DENGAN
METODE *SPREAD SPECTRUM* DAN N-LSB**

SKRIPSI

Oleh
Muhammad Agus Khifdon Yuwafi
NIM. 220601110012

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperole Gelar Sarjana Matematika (S.Mat)

Tanggal, 7 Mei 2026

Ketua Penguji : Hisyam Fahmi, M.Kom.

Anggota Penguji 1 : Mohammad Nafie Jauhari, M.Si.

Anggota Penguji 2 : Muhammad Khudzaifah, M.Si.

Anggota Penguji 3 : Dr. Ach. Nashichuddin, M.A.



.....
.....
.....
.....



Mengetahui,
Ketua Program Studi Matematika


Ach. Nashichuddin, M.Si.
NIP. 19800527 200801 1 012

PERNYATAAN KEASLIAN TULISAN

Saya bertanda tangan di bawah ini

Nama : Muhammad Agus Khifdon Yuwafi

NIM : 220601110012

Program Studi: Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Penyisipan Pesan Dalam File Audio dengan Metode *Spread Spectrum* dan N-LSB

Menyatakan bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri, bukan mengambil alih tulisan atau pemikiran orang lain. Saya menyatakan skripsi ini sebagai pemikiran saya, kecuali dengan mencantumkan sumber kutipan pada daftar rujukan di halaman terakhir. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil plagiasi atau tiruan orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 7 Mei 2026



Muhammad Agus Khifdon Yuwafi
Nim. 220601110012

MOTO

“Jangan pernah ragu untuk melangkah”

“Setiap *tapasya* yang kamu jalani hari ini, adalah jalan menuju *ananda* yang akan kamu rasakan nanti.”

“Jika tidak ada jalan, maka buatlah jalanmu sendiri”

-Roronoa Zoro

HALAMAN PERSEMBAHAN

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim Malang dan menuntaskan penulisan skripsi ini. Skripsi ini penulis persembahkan dengan penuh rasa hormat dan terima kasih kepada:

1. Kedua orang tua tercinta, Bapak Purwanto dan Ibu Ainul Fitriyah serta Kakak saya Muhammad Kharis Abdul Aziz, yang telah memberikan kasih sayang, doa, serta perjuangan tiada lelah agar penulis mampu mengejar pendidikan setinggi-tingginya. Terima kasih atas keteladanan, kerja keras, dan pengorbanan yang tak ternilai selama ini.
2. Seluruh keluarga, sahabat, dan dosen pembimbing yang telah memberikan bimbingan, bantuan, serta semangat yang berarti dalam proses penyusunan skripsi ini.
3. Seluruh teman-teman Epsilon yang selalu memberikan dukungan, motivasi, dan kebersamaan sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
4. Serta kepada diri saya sendiri yang telah gigih dan berusaha semaksimal mungkin untuk bisa menyelesaikan skripsi ini sebaik mungkin, walaupun dihadapkan dengan banyak tantangan dan rintangan selama pengerjaan.

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillah Wa Syukurillah, Segala puji hanya bagi Allah SWT, penulis merasa bersyukur karena dengan rahmat dan pertolongan-Nya, penulis skripsi berjudul "Penyisipan Pesan dalam File Audio Dengan Metode *Spread Spectrum* dan N-LSB" dapat diselesaikan. Skripsi ini merupakan syarat untuk memperoleh gelar sarjana dalam bidang Matematika di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Shalawat dan salam senantiasa disampaikan kepada Nabi besar Muhammad SAW, yang telah memberikan petunjuk kepada umat manusia dari masa kebodohan (jahiliyah) menuju masa yang penuh cahaya, yaitu era agama Islam. Semoga kita semua nantinya mendapatkan syafaatnya di hari kiamat. *Aamiin, yaa rabbal, alamiin.*

Dalam proses penulisan skripsi ini, tidak dapat dipisahkan dari kontribusidan dukungan berbagai pihak. Oleh karena itu, penulis ingin mengungkapkan rasa terima kasih yang sangat besar kepada semua pihak terkait:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CHARM., CRMP, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Fachrur Rozi, M.Si, selaku Ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Muhammad Khudzaifah, M.Si, selaku Dosen Pembimbing I yang telah bersedia memberikan ilmu dan arahan serta masukan kepada penulis sehingga penulis bisa menyelesaikan skripsi.
5. Dr. Ach. Nashichuddin, M.A, selaku Dosen Pembimbing II yang telah memberikan arahan, masukan dalam bidang keagamaan kepada penulis dalam menentukan integrasi pada skripsi ini.
6. Hisyam Fahmi, M.Kom, selaku Ketua Penguji yang telah memberikan kritik, saran, dan masukan yang sangat berharga demi penyempurnaan skripsi ini.

7. Mohammad Nafie Jauhari, M.Si, selaku Anggota Penguji I yang telah memberikan kritik, saran, dan masukan yang konstruktif sehingga skripsi ini dapat tersusun dengan lebih baik.
8. Seluruh Dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim yang telah membantu, mendoakan dan memotivasi penulis dalam menyelesaikan skripsi.
1. Kedua orang tua tercinta, Ibu Ainul Fitriyah dan Bapak Purwanto, yang telah mendukung, mendoakan, dan memotivasi sepanjang waktu. Serta kakak tercinta, M. Kharis Abdul Aziz, yang selalu memberikan semangat, bantuan, dan motivasi dalam setiap langkah penulis hingga skripsi ini dapat terselesaikan.
2. Seluruh teman-teman dari segala angkatan yang telah meluangkan waktu membantu penulis dalam menyelesaikan skripsi,
3. Pihak-pihak di luar yang tidak dapat disebutkan satu-persatu, yang telah ikut serta berperan dalam membantu penulis menyelesaikan skripsi.

Penulis menyadari dalam menyusun skripsi ini masih terdapat kekurangan yang perlu diperbaiki. Meskipun demikian, harapannya adalah agar skripsi ini dapat memberikan manfaat terutama bagi penulis dan pembaca.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Malang, 7 Mei 2026

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGANTAR	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR SIMBOL	xiv
DAFTAR LAMPIRAN	xv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah	5
1.6 Definisi Istilah	5
BAB II KAJIAN TEORI	8
2.1 Teori Pendukung	8
2.1.1 Keamanan Data Digital	8
2.1.2 Steganografi	8
2.1.3 <i>Linear Congruential Generator (LCG)</i>	9
2.1.4 <i>Spread Spectrum</i>	10
2.1.5 <i>Least Significant Bit (LSB)</i>	15
2.1.6 Audio Digital	21
2.1.7 WAV	22
2.1.8 <i>Spectral Distortion</i>	23
2.2 Nilai Amanah dalam Al-Qur'an dan Hadis	24
2.3 Kajian Topik Dengan Teori Pendukung	26
BAB III METODE PENELITIAN	27
3.1 Jenis Penelitian	27
3.2 Pra Penelitian	27
3.3 Tahapan Penelitian	28
BAB IV HASIL DAN PEMBAHASAN	33
4.1 Proses Penyisipan Pesan dengan <i>Spread Spectrum</i>	33
4.2 Proses Penyisipan Pesan Menggunakan Metode 2-LSB	42
4.3 Proses Ekstraksi	48
4.4 Evaluasi Kualitas Audio dengan <i>Spectral Distortion</i>	54

4.5 Integrasi Nilai-Nilai Islam dalam Keamanan Informasi.....	63
BAB V PENUTUP	65
5.1 Kesimpulan.....	65
5.2 Saran	65
DAFTAR RUJUKAN	67
LAMPIRAN.....	69
RIWAYAT HIDUP	90

DAFTAR TABEL

Tabel 4.1 Representasi ASCII ke Biner 8-bit.....	33
Tabel 4.2 Representasi Nilai LCG dalam Bentuk Biner 8-bit	38
Tabel 4.3 Data 4 Sampel Pertama Audio Asli dan Audio Stego	55
Tabel 4.4 Rekapitulasi <i>Spectrum</i> Amplitudo	57
Tabel 4.5 Pengaruh Lokasi Penyisipan pada Nilai <i>Spectral Distortion</i>	61
Tabel 4.6 Pengaruh Nilai N-LSB terhadap <i>Spectral Distortion</i>	62

DAFTAR GAMBAR

Gambar 2.1 Penyisipan <i>Spread Spectrum</i>	14
Gambar 2.2 Ekstraksi <i>Spread Spectrum</i>	14
Gambar 2.3 Penyisipan LSB	20
Gambar 2.4 Ekstraksi LSB.....	20
Gambar 3.1 <i>Flowchart Proses Penyisipan</i>	32
Gambar 3.2 <i>Flowchart Proses Ekstraksi</i>	32
Gambar 4.1 Grafik Penyisipan di tengah	60
Gambar 4.2 Grafik SD pada 2-LSB	62

DAFTAR SIMBOL

x_n	:Bilangan acak ke-n hasil pembangkitan LCG
x_0	:Nilai awal s pada LCG
A	:Konstanta pengali pada LCG
C	:Konstanta penambah pada LCG
m	:Modulus pada LCG
PN	: <i>Pseudo Noise Sequence</i>
L	:Panjang pesan dalam bit
N	:Jumlah bit pada metode N-LSB
n	:Indeks atau urutan data
S	:Sinyal audio asli
S'	:Sinyal audio stego
α	:Faktor skala pada penyisipan <i>Spread Spectrum</i>
k	:Indeks frekuensi
A_k	:Spektrum amplitudo audio asli pada frekuensi ke-k
A'_k	:Spektrum amplitudo audio stego pada frekuensi ke-k
SD	: <i>Spectral Distortion</i>
\oplus	:Operasi XOR

DAFTAR LAMPIRAN

Lampiran 1 Kode Progam Mengubah Audio ke Integer	69
Lampiran 2 Kode Progam mengubah Integer ke Audio	71
Lampiran 3 Kode Progam Penyisipan Pesan	73
Lampiran 4 Kode Progam Ekstraksi Pesan	76
Lampiran 5 Kode Progam Evaluasi	79
Lampiran 6 Gambar <i>Spectral Distortion</i>	82
Lampiran 7 Progam Penyisipan <i>Direct Sequence Spread Spectrum</i> (DSSS) ...	85
Lampiran 8 Progam Ekstraksi <i>Direct Sequence Spread Spectrum</i> (DSSS).....	87
Lampiran 9 Data Audio.....	89

ABSTRAK

Yuwafi, Muhammad Agus Khifdon. 2026. Penyisipan Pesan Dalam File Audio dengan Metode *Spread Spectrum* dan N-LSB. Skripsi. Jurusan Matematika fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II) Ach. Nashichuddin, M.A.

Kata Kunci: Steganografi Audio, N-LSB, DSSS, LCG, Keamanan Data, Pesan teks.

Keamanan informasi menjadi aspek penting dalam perkembangan teknologi digital, khususnya dalam menjaga kerahasiaan data yang ditransmisikan melalui media terbuka. Steganografi audio merupakan teknik penyembunyian informasi ke dalam file audio digital tanpa menimbulkan perubahan yang mudah dikenali pada kualitas suara. Penelitian ini mengimplementasikan metode N-LSB (*Least Significant Bit*) yang dikombinasikan dengan *Direct Sequence Spread Spectrum* (DSSS) dan pembangkitan *pseudo sequence* menggunakan *Linear Congruential Generator* (LCG). Proses penyisipan dilakukan dengan mengubah pesan menjadi biner, kemudian dilakukan *Spreading* menggunakan DSSS berdasarkan *PN Sequence* dari LCG, lalu disisipkan ke dalam sampel audio menggunakan metode N-LSB dengan variasi nilai N dan posisi penyisipan, serta penyimpanan informasi parameter dalam header untuk mendukung proses ekstraksi. Proses ekstraksi dilakukan dengan mengambil kembali bit dari audio, melakukan demodulasi DSSS, dan mengonversi kembali ke bentuk teks. Berdasarkan hasil pengujian, pesan yang disisipkan dapat diekstraksi kembali sesuai dengan data aslinya menggunakan parameter yang sama. Hasil evaluasi kualitas audio menggunakan *Spectral Distortion* (SD) menunjukkan perbedaan spektrum frekuensi yang relatif rendah antara audio asli dan audio stego berdasarkan parameter pengujian yang digunakan dalam penelitian ini.

ABSTRACT

Yuwafi, Muhammad Agus Khifdon. 2026. Embedding Messages in Audio Files Using the Spread Spectrum and N-LSB Methods. Thesis. Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim, Malang. Advisors: (I) Muhammad Khudzaifah, M.Si. (II) Ach. Nashichuddin, M.A.

Keywords: Audio Steganography, N-LSB, DSSS, LCG, Data Security, Text Messages.

Information security is an important aspect in the development of digital technology, especially in maintaining the confidentiality of data transmitted through open media. Audio steganography is a technique of hiding information into digital audio files without causing easily recognizable changes in sound quality. This study implements the N-LSB (Least Significant Bit) method combined with Direct Sequence Spread Spectrum (DSSS) and pseudo sequence generation using a Linear Congruential Generator (LCG). The insertion process is carried out by converting the message into binary, then Spreading using DSSS sbased on the PN Sequence of LCG, then inserted into the audio sample using the N-LSB method with variations in N values and insertion positions, as well as storing parameter information in the header to support the extraction process. The extraction process is done by retrieving bits from the audio, performing DSSS demodulation, and converting them back to text form. Based on the test results, the inserted message can be extracted back according to its original data using the same parameters. The results of the audio quality evaluation using Spectral Distortion (SD) showed a relatively low frequency spectrum difference between the original audio and the stego audio based on the test parameters used in this study.

مستخلص البحث

يواني، محمد أكّوس حفظ. ٢٠٢٦. إدراج الرسائل في الملفات الصوتية باستخدام طرق انتشار الطيف و $N-LSBs$. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانغ. المشرف: (الأول) خديفة، املا جستري فيالعلوم. (الثاني) الأستاذ ناصح الدين، في التربية

الكلمات الأساسية: الطباعة الصوتية، $N-LSB$ ، $DSSS$ ، LCG ، أمن البيانات، الرسائل النصية.

يعد أمن المعلومات جانبا مهما في تطوير التكنولوجيا الرقمية، خاصة في الحفاظ على سرية البيانات المنقولة عبر وسائل الإعلام المفتوحة. التخفيد الصوتي هو تقنية لإخفاء المعلومات داخل ملفات الصوت الرقمية دون التسبب في تغييرات واضحة في جودة الصوت. طبقت هذه الدراسة طريقة $N-LSB$ (البت الأقل أهمية) مع طيف انتشار التسلسل المباشر ($DSSS$) وتوليد التسلسلات الزائفة باستخدام مولد التطابق الخطي (LCG). تم تنفيذ عملية الإدراج عن طريق تحويل الرسالة إلى ثنائية، ثم النشر باستخدام $DSSS$ بناء على تسلسل PN من LCG ، ثم إدراجها في العينة الصوتية باستخدام طريقة $N-LSB$ مع تغييرات في N قيم ومواقع الإدراج، بالإضافة إلى تخزين معلومات المعلمات في الرأس لدعم عملية الاستخراج. تم عملية الاستخراج عن طريق استرجاع البتات من الصوت، وإجراء إزالة التعديلات $DSSS$ ، وتحويلها مرة أخرى إلى شكل نصي. استنادا إلى نتائج الاختبار، يمكن استخراج الرسالة المدخلة مرة أخرى وفقا لبياناتها الأصلية باستخدام نفس المعايير. أظهرت نتائج تقييم جودة الصوت باستخدام التشويه الطيفي (SD) وجود فرق تردد منخفض نسبيا بين الصوت الأصلي وصوت الستيغو بناء على معلمات الاختبار المستخدمة في هذه الدراسة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi digital yang pesat telah mendorong peningkatan pertukaran informasi dalam berbagai bentuk, termasuk pesan teks dan dokumen digital. Namun, kemudahan tersebut juga diiringi dengan meningkatnya risiko terhadap keamanan dan kerahasiaan informasi. Informasi digital berisiko diakses oleh pihak yang tidak berwenang. Pesan digital yang dikirimkan melalui media komunikasi yang tidak aman berpotensi disadap, dimodifikasi, atau disalahgunakan oleh pihak yang tidak berwenang. Oleh karena itu, diperlukan suatu mekanisme pengamanan yang tidak hanya melindungi isi pesan, tetapi juga mampu menyembunyikan keberadaan pesan tersebut agar tidak mudah terdeteksi (Munir, 2021).

Dalam perspektif Islam, urgensi menjaga keamanan dan privasi pesan digital sejalan dengan nilai-nilai yang diajarkan dalam Al-Qur'an. Allah Swt. berfirman dalam Surah Al-Mu'minun ayat 8:

وَالَّذِينَ هُمْ لِأَمْتِنِهِمْ وَعَهْدِهِمْ رَاعُونَ

Artinya: "Dan (sungguh beruntung) orang yang memelihara amanah-amanah dan janjinya."

Menurut tafsir Al-Wajiz karya Wahbah az-Zuhaili, ayat ini menegaskan bahwa amanah mencakup seluruh bentuk tanggung jawab yang dipercayakan kepada individu, tidak terbatas pada aspek pernikahan, tetapi meliputi janji dan perjanjian dalam berbagai konteks sosial. Menjaga amanah merupakan karakter utama orang beriman dan menjadi indikator integritas moral dalam interaksi sosial (Az-Zuhaili, 2011).

Salah satu pendekatan yang digunakan dalam pengamanan pesan digital adalah steganografi, yaitu teknik penyembunyian pesan ke dalam suatu media penampung sehingga keberadaan pesan tidak terlihat secara langsung. Media penampung yang umum digunakan dalam steganografi antara lain citra, video, dan audio. Di antara media tersebut, audio digital memiliki karakteristik berupa jumlah sampel yang besar dan bersifat kontinu, sehingga memungkinkan proses penyisipan pesan dilakukan dengan perubahan yang sangat kecil dan sulit dikenali oleh indera pendengaran manusia (Hussain dkk., 2018).

Beberapa penelitian terdahulu telah mengkaji penerapan steganografi pada media audio sebagai upaya pengamanan pesan digital. Teknik *Spread Spectrum*, khususnya *Direct Sequence Spread Spectrum* (DSSS), digunakan dengan cara menyebarkan bit pesan ke dalam sinyal audio sehingga keberadaannya menyerupai derau (*noise*) dan sulit dideteksi oleh pihak yang tidak berwenang (Alarood dkk., 2022). Meskipun demikian, penerapan teknik ini pada beberapa penelitian masih memiliki keterbatasan, terutama dalam hal kapasitas penyisipan pesan dan efisiensi implementasi. Selain itu, kompleksitas prosesnya dapat memengaruhi efisiensi sistem steganografi.

Metode lain yang banyak digunakan dalam steganografi audio adalah *Least Significant Bit* (LSB), yang bekerja dengan memodifikasi bit dengan bobot paling rendah pada setiap sampel audio. Metode ini memiliki keunggulan dalam kemudahan implementasi dan kemampuannya mempertahankan kualitas audio. Akan tetapi, penggunaan LSB secara tunggal memiliki kelemahan terhadap deteksi dan manipulasi sinyal apabila tidak dikombinasikan dengan teknik lain yang mampu menyamarkan pola penyisipan pesan (Alqahtani dkk., 2020).

Berdasarkan keterbatasan tersebut, diperlukan pendekatan yang mengombinasikan keunggulan dari kedua metode. *Spread Spectrum* digunakan untuk menyebarkan pesan ke seluruh sinyal audio agar sulit dideteksi, sedangkan *N-Least Significant Bit* (N-LSB) digunakan untuk menyisipkan pesan ke dalam beberapa bit dengan bobot terendah pada setiap sampel audio tanpa menurunkan kualitas suara secara signifikan (Assyahid dkk., 2018). Dengan demikian, kombinasi kedua metode ini diharapkan mampu meningkatkan keamanan dan kerahasiaan pesan yang disisipkan ke dalam file audio.

Oleh karena itu, penelitian ini difokuskan pada penyisipan pesan dalam file audio WAV menggunakan metode *Spread Spectrum* dan *N-Least Significant Bit* (N-LSB) serta proses ekstraksi pesan untuk memperoleh kembali pesan asli. Pendekatan ini diharapkan dapat menghasilkan sistem steganografi audio yang efektif dalam menyembunyikan pesan digital dengan tetap menjaga kualitas audio dan keamanan informasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah penelitian ini adalah:

1. Bagaimana hasil simulasi penyisipan pesan ke dalam media audio WAV menggunakan metode *Spread Spectrum* dan *N-Least Significant Bit* (N-LSB)?
2. Bagaimana hasil simulasi ekstraksi untuk memperoleh kembali pesan asli?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penulisan dari penelitian ini adalah sebagai berikut:

1. Menganalisis dan mengimplementasikan proses penyisipan pesan ke dalam media audio WAV menggunakan metode *Spread Spectrum* dan *N-Least Significant Bit* (N-LSB).
2. Menganalisis dan menguji proses ekstraksi pesan dari audio stego untuk memperoleh kembali pesan asli secara utuh.

1.4 Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut

1. Manfaat Teoritis

Menambah literatur ilmiah di bidang keamanan informasi, khususnya pada kajian steganografi audio dengan menggunakan metode *Spread Spectrum* dan *N-Least Significant Bit* (N-LSB). Selain itu, penelitian ini juga memperkuat pemahaman konseptual mengenai teknik penyisipan dan ekstraksi pesan digital pada media audio WAV.

2. Manfaat Praktis

- a. Memberikan pemahaman dan pengalaman langsung bagi peneliti dalam mengimplementasikan metode steganografi audio menggunakan teknik *Spread Spectrum* dan *N-LSB*.
- b. Menjadi referensi tambahan bagi program studi atau instansi akademik dalam pengembangan pembelajaran dan penelitian di bidang steganografi, keamanan data, dan pengolahan sinyal digital.
- c. Memberikan wawasan kepada pembaca mengenai penerapan teknik penyisipan pesan digital pada media audio, sehingga dapat dijadikan dasar dalam pengembangan sistem pengamanan data digital yang lebih aman dan efisien.

1.5 Batasan Masalah

Berdasarkan perumusan masalah dan tujuan penelitian, penelitian ini dibatasi pada hal-hal berikut:

1. Pesan yang disisipkan berupa teks digital yang direpresentasikan menggunakan kode ASCII.
2. Media penampung yang digunakan adalah file audio WAV dengan kanal mono, sehingga setiap sampel audio hanya terdiri dari satu kanal sinyal.
3. Metode steganografi yang digunakan dalam penelitian ini adalah *Spread Spectrum* dan *N-Least Significant Bit* (N-LSB) untuk proses penyisipan dan ekstraksi pesan.
4. Proses *Spreading* pada metode *Spread Spectrum* menggunakan *Spreading factor* sebesar 4.

1.6 Definisi Istilah

Untuk menghindari kesalahpahaman dalam penafsiran istilah yang digunakan, berikut penjelasan beberapa istilah penting yang terkait dengan penelitian ini:

1. Keamanan Data Digital

Upaya melindungi informasi dari akses, perubahan, dan penyalahgunaan oleh pihak tidak berwenang agar kerahasiaan, keaslian, dan integritas data tetap terjaga.

2. Steganografi

Teknik menyembunyikan pesan rahasia ke dalam media digital seperti gambar, audio, atau video sehingga keberadaan pesan tersebut tidak diketahui oleh pihak lain.

3. *Least Significant Bit (LSB)*

Metode steganografi yang menyisipkan data pada bit paling rendah dari media digital tanpa menyebabkan perubahan yang terlihat atau terdengar.

4. *N-Least Significant Bit (N-LSB)*

Pengembangan dari LSB yang menyisipkan data pada beberapa bit terendah tiap sampel audio, meningkatkan kapasitas penyisipan tanpa menurunkan kualitas suara.

5. *Spread Spectrum*

Metode penyebaran sinyal yang menyebarkan data rahasia ke seluruh spektrum frekuensi agar pesan sulit dideteksi.

6. *Direct Sequence Spread Spectrum (DSSS)*

Teknik *Spread Spectrum* yang mengalikan bit pesan dengan urutan acak semu (*PN Sequence*), menyamarkan pesan agar menyerupai *noise* dan memungkinkan proses *DeSpreading* untuk pemulihan pesan.

7. *Despreanding*

proses untuk mengembalikan sinyal atau data yang telah disebarkan ke bentuk aslinya dengan menggunakan kode penyebar yang sama.

8. *Noise*

sinyal atau gangguan acak yang tidak diinginkan yang dapat memengaruhi kualitas sinyal utama.

9. *Pulse Code Modulation (PCM)*

Metode penyimpanan audio digital yang merekam nilai amplitudo sampel suara dalam bentuk bilangan bulat, memungkinkan modifikasi pada tingkat bit untuk penyisipan pesan.

10. Audio Cover

File audio asli yang menjadi media penyisipan pesan rahasia.

11. Audio Stego

File audio hasil penyisipan pesan rahasia yang kualitas suaranya tetap menyerupai file asli.

12. Ekstraksi

Proses mengambil kembali pesan tersembunyi dari media steganografi.

13. Python

Bahasa pemrograman yang digunakan dalam implementasi penyisipan dan ekstraksi pesan pada audio.

BAB II

KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Keamanan Data Digital

Keamanan data digital merupakan upaya untuk melindungi informasi dari akses, perubahan, dan penyalahgunaan oleh pihak yang tidak berwenang. Menurut Munir (2004), keamanan informasi meliputi perlindungan terhadap kerahasiaan, keaslian, dan integritas data, baik pada saat penyimpanan maupun proses pengiriman. Data digital memiliki tingkat kerentanan yang tinggi terhadap ancaman seperti penyadapan dan modifikasi, sehingga diperlukan mekanisme pengamanan yang mampu mencegah akses ilegal. Perkembangan teknologi komunikasi yang semakin pesat turut meningkatkan kebutuhan akan sistem pengamanan data yang lebih efektif. Oleh karena itu, penerapan metode pengamanan menjadi penting agar informasi tetap terjaga keamanannya dan hanya dapat diakses oleh pihak yang berhak.

2.1.2 Steganografi

Steganografi adalah bidang ilmu dan seni yang mempelajari cara menyembunyikan pesan rahasia di dalam media digital seperti gambar, audio, video, atau teks, agar keberadaannya tidak mudah terdeteksi oleh pihak yang tidak berwenang. Fokus utamanya bukan pada isi pesan, tetapi pada penyamaran keberadaan pesan itu sendiri di dalam media. Secara etimologis, istilah steganografi berasal dari bahasa Yunani, yaitu *steganos* yang berarti “tersembunyi” dan *graphein* yang berarti “menulis”, sehingga secara harfiah diartikan sebagai “menulis secara tersembunyi” (Hussain dkk., 2018).

Karakteristik utama steganografi:

1. Penyembunyian Data: Informasi rahasia disisipkan dalam media digital (misalnya, menyembunyikan pesan teks di dalam gambar atau video).
2. Media Penyamaran: Media yang digunakan sebagai penyamaran dapat berupa gambar, audio, video, atau dokumen teks biasa.
3. Ketidakterlihatan Pesan: Karakteristik utama steganografi adalah kemampuan menyembunyikan pesan tanpa menimbulkan perubahan yang dapat dideteksi secara kasat indera. Dengan menggunakan metode *Spread Spectrum* dan *N-Least Significant Bit (N-LSB)*, perubahan pada sinyal audio diharapkan tetap berada pada tingkat yang tidak mengganggu kualitas audio asli.

2.1.3 Linear Congruential Generator (LCG)

Linear Congruential Generator (LCG) merupakan salah satu metode pembangkit bilangan acak semu (*pseudo-random number generator*) yang banyak digunakan karena kesederhanaan dan efisiensinya. Metode ini menghasilkan deret bilangan yang tampak acak berdasarkan rumus sebagai berikut:

$$X_{n+1} = (aX_n + c) \bmod m \quad (2.1)$$

dengan:

X_n : Bilangan acak ke- n

a : Konstanta pengali

c : Konstanta penambah

m : Modulus

Metode LCG menghasilkan bilangan acak dalam rentang 0 hingga $m - 1$. Kualitas bilangan yang dihasilkan dipengaruhi oleh pemilihan parameter a , c , dan m . Agar

deret bilangan yang dihasilkan memiliki periode yang lebih panjang dan pola yang tidak cepat berulang, parameter tersebut perlu memenuhi beberapa syarat, di antaranya nilai c relatif prima terhadap modulus m , nilai $a - 1$ habis dibagi oleh seluruh faktor prima dari m , serta apabila m merupakan kelipatan 4 maka $a - 1$ juga habis dibagi 4. Pemilihan parameter yang sesuai dapat membantu menghasilkan deret bilangan acak semu yang lebih baik untuk digunakan dalam proses pembangkitan *PN Sequence*.

2.1.4 Spread Spectrum

Spread Spectrum merupakan metode penyembunyian data dengan cara menyebarkan sinyal pesan ke dalam spektrum yang lebih luas sehingga menyerupai derau (*noise*). Dalam bidang steganografi, teknik ini digunakan untuk menyamarkan keberadaan pesan rahasia dengan memodulasi pesan menggunakan urutan bilangan acak semu (*pseudo noise sequence* atau *PN Sequence*). Dengan penyebaran tersebut, pola pesan menjadi sulit dikenali sehingga meningkatkan tingkat keamanan data (Assyahid dkk., 2018).

Salah satu bentuk penerapan metode *Spread Spectrum* adalah *Direct Sequence Spread Spectrum* (DSSS). DSSS merupakan teknik penyebaran sinyal yang bekerja dengan cara mengoperasikan XOR setiap bit pesan dengan deret bilangan acak semu (*pseudo noise sequence*). Proses ini menyebabkan sinyal pesan tersebar ke dalam spektrum yang lebih luas sehingga menyerupai derau dan sulit dibedakan dari sinyal asli (Assyahid dkk., 2018).

Hasil dari proses DSSS adalah sinyal pesan yang telah tersebar (*spread signal*). Sinyal ini memiliki karakteristik menyerupai derau dan memiliki spektrum frekuensi yang lebih lebar dibandingkan sinyal pesan asli. Sinyal hasil

penyebaran tersebut selanjutnya digunakan sebagai sinyal pesan yang akan disisipkan ke dalam media penampung pada tahap steganografi. Untuk memperoleh kembali pesan tersebut, dilakukan proses *DeSpreading* menggunakan *PN Sequence* yang sama dengan yang digunakan pada proses penyebaran.

Keberhasilan metode DSSS sangat dipengaruhi oleh kesesuaian *PN Sequence* yang digunakan pada tahap penyisipan dan ekstraksi. Jika *PN Sequence* yang digunakan berbeda, maka proses demodulasi tidak dapat dilakukan dengan tepat sehingga pesan asli berpotensi gagal dipulihkan. Oleh karena itu, digunakan pembangkit bilangan acak semu yang bersifat deterministik seperti LCG agar *PN Sequence* dapat dibangkitkan kembali secara identik pada proses ekstraksi. Dengan demikian, metode DSSS tidak hanya meningkatkan keamanan melalui penyamaran sinyal, tetapi juga memastikan proses ekstraksi pesan dapat dilakukan dengan baik selama parameter yang digunakan tetap sama.

Perlu diperhatikan bahwa proses penyebaran menggunakan metode DSSS hanya diterapkan pada data pesan, sedangkan *Header* tidak melalui proses penyebaran. Hal ini bertujuan agar informasi pada *Header* dapat diakses secara langsung tanpa memerlukan proses *DeSpreading*.

Langkah-langkah kerja *Direct Sequence Spread Spectrum* (DSSS) sebagai berikut:

1. Ambil data yang akan disembunyikan.

Pesan rahasia yang akan disisipkan dikonversi terlebih dahulu ke dalam bentuk biner agar dapat diproses secara digital.

2. Melakukan proses *Spreading* bit.

Setiap bit pesan kemudian diperluas menjadi beberapa bit yang sama sesuai dengan panjang *Spreading* yang digunakan. Misalnya jika panjang

Spreading adalah 4, maka bit 0 diubah menjadi 0000 dan bit 1 diubah menjadi 1111.

3. Membangkitkan urutan acak semu (*PN Sequence*)

Sistem kemudian membangkitkan *pseudo noise sequence (PN Sequence)* menggunakan metode pembangkit bilangan acak semu, misalnya LCG.

4. Melakukan proses penyebaran sinyal (*Spreading*)

Bit hasil *Spreading* kemudian di XOR-kan dengan *PN Sequence* sehingga menghasilkan *spread signal*. Proses ini membuat sinyal pesan memiliki spektrum yang lebih lebar dan menyerupai derau (*noise*).

5. Menghasilkan sinyal pesan tersebar (*spread signal*)

Hasil dari proses DSSS adalah sinyal pesan yang telah tersebar, yang kemudian dapat digunakan pada tahap berikutnya dalam sistem steganografi untuk disisipkan ke dalam media penampung.

Contoh perhitungan sederhana dari *Spread Spectrum*

Misalkan pesan yang akan disisipkan adalah huruf “A”, yang dalam kode ASCII bernilai 65 atau 01000001 dalam biner.

Langkah 1 : Konversi Pesan ke Biner

Pesan : A

Biner : 01000001

Langkah 2 : Penyebaran (*Spreading*)

Pesan biner ini akan disebar menggunakan DSSS dengan faktor pengali 4.

Setiap bit pesan direplikasi sebanyak 4 kali, menghasilkan:

0 → 0000

1 → 1111

Hasil: 00001111 00000000 00000000 00001111

Langkah 3 : Pembangkitan *Pseudonoise (PN Sequence)*

Urutan acak semu (*pseudonoise*) dihasilkan dari *seed* (misalnya dari kata kunci “key”) menggunakan LCG.

Contoh *PN Sequence* (acak semu):

10101100 11001010 01110110 00100101

Langkah 4 : Modulasi (XOR)

Pesan tersebar kemudian dimodulasikan dengan *PN Sequence* menggunakan operasi XOR untuk menghasilkan sinyal campuran (*spread signal*):

Pesan tersebar : 00001111 00000000 00000000 00001111

PN Sequence : 10101100 11001010 01110110 00100101

Hasil XOR : 10100011 11001010 01110110 00101010

Hasil inilah yang kemudian akan disisipkan ke dalam bit-bit audio

Langkah 5 : Ekstraksi

Saat proses ekstraksi, sinyal hasil *embedding* akan dilakukan operasi XOR ulang dengan *PN sequence* untuk memperoleh kembali bit pesan asli.

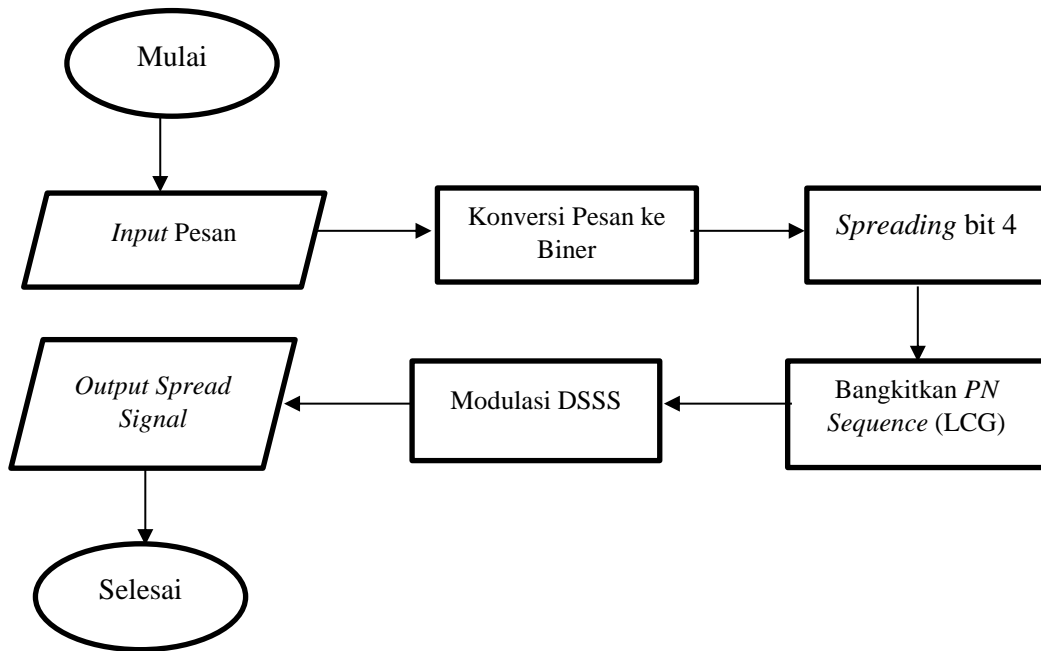
PN Spread signal : 10100011 11001010 01110110 00101010

PN Sequence : 10101100 11001010 01110110 00100101

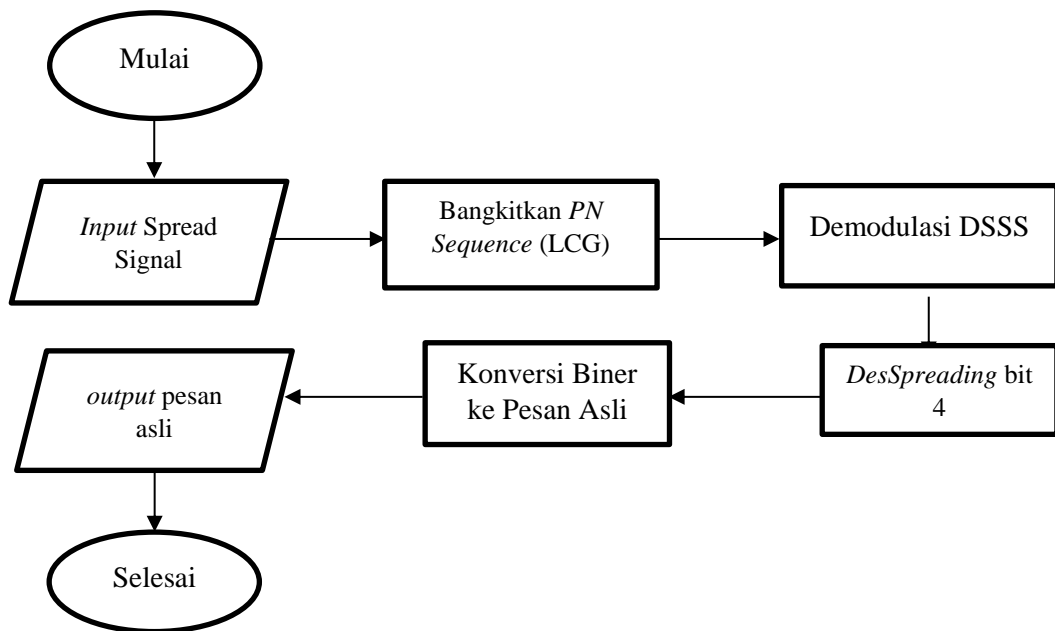
Hasil XOR : 00001111 00000000 00000000 00001111

Hasil operasi XOR berupa deretan bit 00001111 00000000 00000000 00001111 kemudian diproses melalui tahap *deSpreading* dengan menggunakan kode penyebar yang sama seperti pada proses *Spreading*. Dengan *Spreading factor* sebesar 4, deretan bit dibagi menjadi beberapa kelompok 4-bit yaitu 0000,1111 | 0000,0000,0000,0000,1111. Selanjutnya setiap kelompok dikonversi

kembali berdasarkan aturan $0000 \rightarrow 0$ dan $1111 \rightarrow 1$, sehingga diperoleh deretan bit 01000001 yang kemudian dikonversi ke kode ASCII dan menghasilkan karakter "A". Langkah-langkah proses Spread Spectrum pada tahap penyisipan dan ekstraksi pesan dapat dilihat pada Gambar 2.1 dan Gambar 2.2..



Gambar 2.1 Penyisipan *Spread Spectrum*



Gambar 2.2 Ekstraksi *Spread Spectrum*

2.1.5 *Least Significant Bit (LSB)*

Metode *Least Significant Bit (LSB)* merupakan teknik steganografi yang sederhana dan mudah diterapkan. Prinsip kerjanya adalah mengganti bit dengan bobot paling rendah LSB pada data penampung, seperti piksel gambar atau sampel audio, dengan bit-bit pesan rahasia (Alarood dkk., 2022). Karena perubahan dilakukan pada bit terkecil, perbedaan yang dihasilkan tidak terlihat secara visual maupun terdengar secara auditori, sehingga kualitas media tetap terjaga meskipun telah dimodifikasi.

Meskipun sederhana dan efektif, metode LSB memiliki kelemahan yaitu kapasitas penyisipan yang terbatas serta rentan terhadap manipulasi atau kompresi media. Oleh karena itu, dikembangkan metode N-LSB yang memungkinkan penyisipan data pada lebih dari satu bit terendah untuk meningkatkan kapasitas tanpa menurunkan kualitas media secara signifikan (Alqahtani dkk., 2020).

Pada metode LSB dalam audio digital, sinyal suara direpresentasikan sebagai sampel dengan nilai amplitudo dalam bentuk biner, umumnya 8-bit atau 16-bit pada format WAV mono. Teknik ini menyisipkan pesan dengan mengganti bit paling rendah (*least significant bit*) dari setiap sampel. Karena perubahan terjadi pada bit dengan bobot terkecil, modifikasi tersebut tidak memengaruhi kualitas audio secara signifikan dan sulit dideteksi oleh pendengaran manusia.

Dalam penelitian ini, digunakan *Header* sepanjang 72-bit yang disisipkan pada awal audio untuk menyimpan informasi penting yang dibutuhkan dalam proses ekstraksi. *Header* ini berfungsi sebagai penanda parameter penyisipan agar dapat digunakan kembali secara tepat pada tahap ekstraksi. Keberadaan header membantu memastikan proses ekstraksi dapat dilakukan secara akurat.

Header 72-bit tersebut terdiri dari beberapa bagian dengan fungsi masing-masing sebagai berikut:

1. 16 bit pertama digunakan untuk menyimpan informasi panjang pesan dalam satuan bit. Nilai tersebut merepresentasikan jumlah keseluruhan data pesan yang akan disisipkan ke dalam audio, sehingga proses ekstraksi dapat mengetahui ukuran pesan yang harus diambil dari media stego secara tepat.
2. 8 bit kedua digunakan untuk menyimpan informasi posisi penyisipan pesan dalam audio, dengan representasi sebagai berikut:
 - a. 00 : penyisipan pada bagian awal audio
 - b. 01 : penyisipan pada bagian tengah audio
 - c. 10 : penyisipan pada bagian akhir audio
3. 8 bit berikutnya digunakan untuk menyimpan nilai N pada metode N-LSB, yang menunjukkan jumlah bit paling rendah yang digunakan dalam proses penyisipan.
4. 40 bit berikutnya (Parameter LCG): Bagian ini menyimpan variabel kunci untuk algoritma LCG. Parameter ini dibagi menjadi empat bagian masing-masing 8 bit:
 - a. 8 bit untuk nilai *Seed*
 - b. 8 bit untuk nilai (a)
 - c. 8 bit untuk nilai (c)
 - d. 16 bit untuk nilai Modulus (m)

Header disisipkan pada bagian awal audio sebelum data pesan utama. Dengan demikian, pada proses ekstraksi, sistem terlebih dahulu membaca 72-bit awal dari audio stego untuk memperoleh informasi yang tersimpan di dalam

Header. Informasi tersebut meliputi panjang pesan, lokasi penyisipan, nilai parameter N-LSB, serta nilai parameter LCG yang digunakan dalam proses penyisipan.

Berdasarkan informasi pada *Header*, sistem menentukan jumlah bit yang diekstraksi dan posisi penyisipan. Parameter LCG digunakan kembali untuk membangkitkan *PN Sequence* yang sesuai, sehingga proses ekstraksi diawali dengan pembacaan *Header* agar pengambilan data pesan berlangsung tepat, terstruktur, dan konsisten dengan proses penyisipan. Tahap ini berperan penting dalam menjaga keakuratan hasil ekstraksi pesan yang disembunyikan.

Langkah-langkah kerja metode LSB dalam penyisipan pesan pada audio adalah sebagai berikut:

1. Ambil data yang akan disembunyikan. Pesan rahasia dikonversi terlebih dahulu ke dalam bentuk biner.
2. Mengambil file audio penampung, yaitu audio berformat WAV yang digunakan sebagai media penyisipan karena bersifat tidak terkompresi dan mudah dimodifikasi pada tingkat bit.
3. Bentuk *Header* 72-bit. *Header* dibentuk untuk menyimpan informasi panjang pesan, lokasi penyisipan, serta nilai parameter N-LSB.
4. Sisipkan data rahasia. Bit-bit pesan disisipkan ke dalam bit paling rendah dari setiap sampel audio secara berurutan hingga seluruh data tersimpan.
5. Menghasilkan file stego-audio, yaitu audio yang telah disisipi pesan rahasia dan disimpan dengan kualitas suara yang tetap mendekati audio asli.
6. Ambil file stego-audio. File audio yang telah mengandung pesan rahasia digunakan untuk proses ekstraksi.

7. Baca *Header* 72-bit. Sistem mengambil 72-bit awal dari audio untuk memperoleh informasi panjang bit pesan, lokasi penyisipan, serta nilai parameter N-LSB yang digunakan.
8. Ekstrak bit LSB. Bit paling rendah dari setiap sampel diambil secara berurutan untuk mendapatkan kembali bit-bit pesan yang telah disisipkan.
9. Gabungkan bit menjadi data biner. Bit-bit hasil ekstraksi disusun kembali menjadi deretan biner sesuai urutan penyisipannya.
10. Konversi biner ke pesan asli. Data biner diubah kembali ke bentuk teks menggunakan kode ASCII sehingga diperoleh pesan rahasia semula.

Metode ini kemudian dikembangkan menjadi N-LSB, yaitu teknik penyisipan data yang dilakukan pada lebih dari satu bit terendah di setiap sampel audio. Penggunaan beberapa bit sekaligus memungkinkan kapasitas penyimpanan pesan menjadi lebih besar dibandingkan metode LSB biasa. Meskipun demikian, perubahan pada sinyal audio tetap relatif kecil sehingga kualitas audio masih dapat dipertahankan dengan baik.

Sebagai contoh sederhana, sebuah pesan berukuran 8-bit disisipkan ke dalam data audio menggunakan metode *2-Least Significant Bit (2-LSB)*, yaitu dengan memodifikasi dua bit paling tidak signifikan pada setiap sampel audio. Karena setiap sampel audio mampu menampung 2-bit pesan pada dua bit terakhirnya, maka diperlukan sebanyak 4 sampel audio untuk menyimpan keseluruhan pesan tersebut.

Misalkan kita ingin menyembunyikan data rahasia "A" (dalam ASCII, huruf "A" adalah 65 atau 01000001 dalam biner) ke dalam audio digital.

Langkah 1 : Konversi data rahasia ke biner

Data rahasia : “A”

Biner dari “A” : 01000001

Dan data audio tempat penyisipan adalah

00110011 01011001 10100010 01101110

Maka hasil penyisipan pada bit N-LSB adalah:

001100**01** 010110**00** 101000**00** 011011**01**

Langkah 2 : Proses ekstraksi pesan

Pada proses ekstraksi, diambil bit paling rendah dari setiap sampel audio stego sebagai berikut:

00110001 → 01

01011000 → 00

10100000 → 00

01101101 → 01

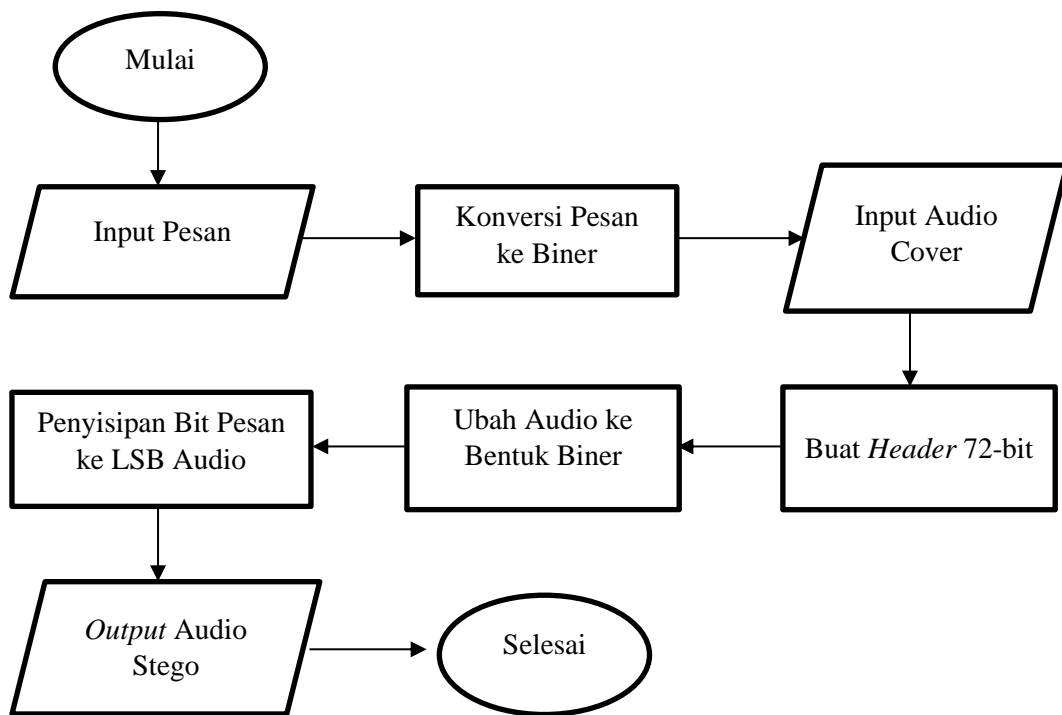
Sehingga diperoleh kembali deretan bit:

01000001

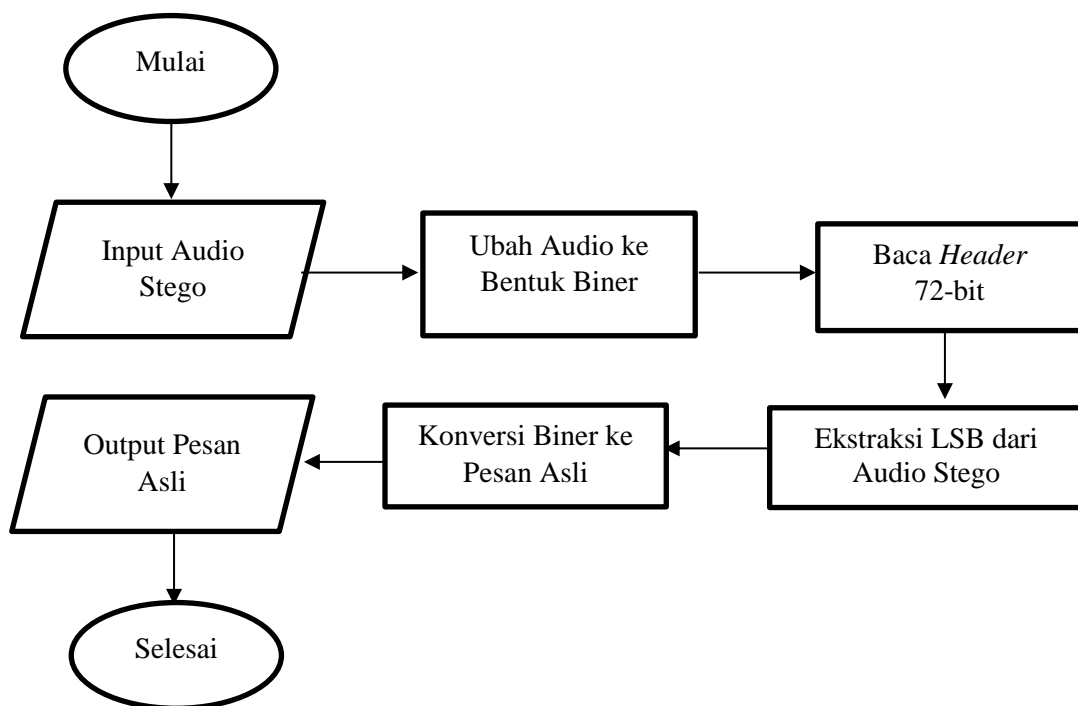
Deretan bit tersebut kemudian dikonversi kembali ke dalam bentuk kode ASCII sehingga menghasilkan karakter “A” sebagai hasil akhir proses ekstraksi data.

Hasil ini menunjukkan keberhasilan proses ekstraksi pesan.

Secara teoritis, penggunaan metode LSB tidak menyebabkan perubahan ukuran media audio secara signifikan karena proses penyisipan hanya memodifikasi bit-bit bernilai paling rendah. Akibatnya, perubahan pada kualitas audio juga relatif kecil sehingga sulit dikenali oleh indra pendengaran manusia. Adapun langkah-langkah proses Least Significant Bit (LSB) pada tahap penyisipan dan ekstraksi pesan dapat dilihat pada Gambar 2.3 dan Gambar 2.4.



Gambar 2.3 Penyisipan LSB



Gambar 2.4 Ekstraksi LSB

2.1.6 Audio Digital

Audio digital merupakan hasil konversi sinyal suara analog ke bentuk digital melalui proses *sampling* dan *kuantisasi*. Proses *sampling* menghasilkan nilai amplitudo suara pada setiap interval waktu tertentu, sedangkan *kuantisasi* mengubahnya menjadi nilai numerik sesuai kedalaman bit (*bit depth*). Format audio seperti WAV umumnya menggunakan metode *Pulse Code Modulation* (PCM) yang bersifat tidak terkompresi, sehingga setiap sampel direkam sebagai bilangan bulat yang merepresentasikan amplitudo gelombang suara. Nilai-nilai tersebut dapat diubah menjadi bentuk biner, misalnya 8-bit atau 16-bit, yang memudahkan manipulasi data pada tingkat bit tanpa menurunkan kualitas suara (Munir, 2004).

Representasi biner inilah yang dimanfaatkan dalam teknik steganografi, terutama metode *N-Least Significant Bit* (*N-LSB*). Pada metode ini, beberapa bit terendah dari setiap sampel audio diganti dengan bit-bit pesan rahasia sehingga perubahan amplitudo yang terjadi sangat kecil dan tidak terdeteksi oleh sistem pendengaran manusia (*Human Auditory System*). Misalnya, perubahan dua bit terakhir dari suatu sampel hanya mengubah nilai amplitudo dalam jumlah yang sangat kecil, sehingga tidak memengaruhi persepsi suara. Dengan demikian, kemampuan audio digital untuk direpresentasikan dalam bentuk bit menjadi dasar penting dalam proses penyisipan dan ekstraksi pesan terenkripsi menggunakan metode N-LSB (Alsolami, 2022).

Dalam sistem audio digital, sinyal suara direpresentasikan dalam bentuk bilangan biner menggunakan metode PCM. Nilai amplitudo sinyal audio dapat

berupa bilangan positif maupun negatif, sehingga diperlukan metode khusus untuk merepresentasikan bilangan negatif dalam bentuk biner.

Sebagai solusi atas kebutuhan tersebut, metode yang umum digunakan adalah komplemen dua (*two's complement*). Dalam penerapannya, bilangan negatif direpresentasikan dengan cara membalik seluruh bit dari bilangan positifnya atau disebut sebagai komplemen satu, yang kemudian diakhiri dengan menambahkan nilai satu pada hasil pembalikan tersebut untuk memperoleh nilai akhir dalam bentuk biner.

Sebagai contoh, untuk merepresentasikan bilangan -5 dalam format 8-bit:

1. Representasi biner dari 5:

0000101

2. Komplemen satu (dibalik):

11111010

3. Tambahkan 1:

11111011

Sehingga, representasi biner dari -5 adalah 11111011.

2.1.7 WAV

Format WAV (*Waveform Audio File Format*) merupakan salah satu format penyimpanan audio digital yang bersifat tidak terkompresi (*uncompressed*) dan menggunakan metode PCM dalam representasi datanya. Setiap sampel suara dalam format ini direkam sebagai nilai amplitudo yang merepresentasikan bentuk gelombang asli, sehingga kualitas suara yang dihasilkan hampir identik dengan sumber aslinya (Alqahtani dkk., 2020).

Struktur file WAV terdiri dari tiga bagian utama, yaitu *Header*, data, dan chunk informasi tambahan. Bagian *Header* berisi metadata seperti ukuran file, jumlah kanal, *sample rate*, dan *bit depth* yang menunjukkan resolusi sinyal. Sementara bagian data menyimpan nilai-nilai amplitudo hasil sampling. Karena bersifat tidak terkompresi, format WAV sering digunakan dalam penelitian steganografi audio karena memungkinkan modifikasi bit secara langsung tanpa menurunkan kualitas suara secara signifikan.

2.1.8 Spectral Distortion

Spectral Distortion (SD) merupakan salah satu metrik evaluasi yang digunakan untuk mengukur tingkat perbedaan spektrum frekuensi antara dua sinyal, yaitu sinyal asli dan sinyal stego. Dalam pengolahan sinyal audio, SD digunakan untuk menilai sejauh mana proses penyisipan pesan memengaruhi karakteristik spektrum frekuensi audio (Alqahtani dkk., 2020). Nilai SD diperoleh dari perbedaan spektrum logaritmik antara sinyal asli dan sinyal hasil penyisipan. Semakin kecil nilai SD, maka semakin kecil perubahan spektrum frekuensi yang terjadi, sehingga kualitas audio tetap terjaga.

Secara matematis, *Spectral Distortion* dirumuskan sebagai berikut:

$$\sqrt{\frac{1}{K} \sum_{k=1}^K \left[10 \log_{10} \left(\frac{S_o(k)}{S_s(k)} \right) \right]^2} \quad (2.2)$$

Dimana:

K : Jumlah komponen frekuensi.

$S_o(k)$: Nilai spektrum amplitudo sinyal audio asli pada frekuensi ke k .

$S_s(k)$: Nilai spektrum amplitudo sinyal audio stego pada frekuensi ke k .

2.2 Nilai Amanah dalam Al-Qur'an dan Hadis

Penelitian ini mengajarkan tentang sebuah tanggungjawab seseorang agar selalu menjalankan kewajiban sesuai dengan apa yang dikerjakannya atau dapat diartikan sebagai seseorang harus menjaga amanah dalam menjalani kehidupan. Berikut adalah firman Allah SWT yang membahas tentang amanah pada Surat An-Nisa ayat 58 sebagai berikut:

﴿ إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمَانَاتِ إِلَىٰ أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ إِنَّ اللَّهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا ﴾

Artinya: "Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan apabila kamu menetapkan hukum di antara manusia, hendaklah kamu menetapkannya dengan adil. Sesungguhnya Allah sebaik-baik yang memberi pengajaran kepadamu. Sesungguhnya Allah Maha Mendengar lagi Maha Melihat."(QS. An-Nisa: 58)

Menurut Tafsir Al-Wajiz karya Wahbah Az-Zuhaili (2013), dua ayat terakhir menjelaskan perbedaan balasan bagi kaum mukmin dan kafir berupa kenikmatan dan siksaan. Selanjutnya, Al-Qur'an memberikan tuntunan hidup terkait amanah, yaitu perintah untuk menyampaikan amanah secara tepat kepada yang berhak serta menetapkan hukum dengan adil ketika terjadi perselisihan. Perintah ini menunjukkan bahwa Allah menekankan pentingnya menjaga amanah dan berlaku adil sebagai bentuk pengajaran yang utama.

Dalam Tafsir Ibnu Katsir (2000) dijelaskan bahwa amanah yang dimaksud dalam ayat ini mencakup segala urusan agama dan dunia, mulai dari pelaksanaan hukum, titipan harta, hingga tugas-tugas sosial dan kepemimpinan. Ibnu Katsir menegaskan bahwa seseorang yang diberi tanggung jawab harus mengembalikannya kepada pemilik atau pihak yang berhak, dan tidak boleh menyelewengkan atau menunda penunaian amanah tersebut. Dengan demikian,

ayat ini menjadi dasar penting dalam menegakkan kejujuran, keadilan, dan tanggung jawab dalam setiap aspek kehidupan.

Selain itu, Rasulullah SAW juga menegaskan pentingnya menjaga rahasia dalam sabdanya:

“Apabila seseorang berkata kepada kamu tentang suatu hal, kemudian dia menoleh ke kanan dan ke kiri, maka itu adalah amanah.” (HR. Abu Dawud, dari Jabir bin ‘Abdillah r.a. no. 4868)

Dalam *‘Awn al-Ma‘būd* karya Al-‘Azīm Abādī (2009) dijelaskan bahwa makna hadis ini menunjukkan apabila seseorang menoleh ke kanan dan ke kiri ketika berbicara, hal itu menandakan bahwa ia tidak menginginkan perkataannya diketahui oleh orang lain. Dengan demikian, pembicaraan tersebut menjadi amanah yang wajib dijaga.

Dengan demikian, amanah dalam Al-Qur’an mencakup tanggung jawab moral dan sosial yang menjadi dasar dalam menjaga kejujuran, keadilan, serta kepercayaan. Nilai tersebut juga mendorong pemanfaatan ilmu dan teknologi secara bertanggung jawab bagi kemaslahatan masyarakat.

Selain itu, dalam mengemban amanah terdapat beberapa prinsip dasar yang harus diperhatikan, yaitu kejujuran sebagai landasan utama dalam menjaga kepercayaan, tanggung jawab karena setiap amanah akan dipertanggungjawabkan, keadilan dalam menyampaikan amanah kepada pihak yang berhak, profesionalitas dalam melaksanakan tugas secara optimal sesuai kemampuan, serta menjaga kerahasiaan informasi dari pihak yang tidak berwenang. Penerapan prinsip-prinsip ini menjadikan amanah tidak hanya sebagai kewajiban moral, tetapi juga sebagai bentuk integritas dalam kehidupan, termasuk dalam konteks penelitian ini yang menekankan pentingnya menjaga keamanan dan kerahasiaan pesan.

2.3 Kajian Topik Dengan Teori Pendukung

Perkembangan teknologi digital mempermudah komunikasi dan pertukaran data, namun juga meningkatkan risiko kebocoran informasi dan serangan siber. Oleh karena itu, diperlukan sistem keamanan yang tidak hanya melindungi isi pesan, tetapi juga menyamarkan keberadaannya agar tidak mudah terdeteksi oleh pihak yang tidak berwenang. Salah satu pendekatan yang dapat digunakan adalah steganografi, yaitu teknik penyembunyian pesan pada media digital.

Penelitian ini menerapkan teknik steganografi audio sebagai upaya pengamanan data dengan cara menyembunyikan pesan ke dalam media audio digital. Metode *Spread Spectrum* digunakan untuk menyebarkan bit-bit pesan ke seluruh spektrum frekuensi audio dengan pola *pseudo noise* sehingga menyerupai derau alami dan sulit dideteksi. Selanjutnya, metode N-LSB digunakan untuk menyisipkan pesan pada beberapa bit dengan bobot terendah dari setiap sampel audio, sehingga proses penyisipan dapat dilakukan tanpa menurunkan kualitas suara secara signifikan. Kombinasi kedua metode tersebut diharapkan dapat meningkatkan keamanan pesan tanpa mengurangi kualitas audio secara signifikan.

Proses penyisipan pesan dilakukan melalui pengolahan sinyal digital dan manipulasi bit biner pada sampel audio. Kualitas audio dievaluasi menggunakan SD, di mana nilai yang lebih kecil menunjukkan perubahan spektrum yang minimal sehingga kualitas tetap terjaga. Selain mempertahankan kualitas audio, sistem yang dikembangkan juga harus mampu mengekstraksi kembali pesan yang disisipkan secara akurat tanpa mengalami perubahan isi informasi. Dengan demikian, kombinasi metode *Spread Spectrum* dan N-LSB diharapkan menghasilkan sistem steganografi audio yang efektif, efisien, dan sulit dideteksi.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini termasuk dalam kategori penelitian eksperimental deskriptif dengan pendekatan komputasional yang bertujuan untuk mengkaji dan menerapkan teknik steganografi audio sebagai upaya pengamanan data digital. Penelitian difokuskan pada implementasi dan analisis proses penyisipan pesan ke dalam media audio digital menggunakan teknik *Spread Spectrum* dan metode N-LSB, yang diawali dengan pengkajian literatur terkait steganografi audio dan pengolahan sinyal digital, kemudian dilanjutkan dengan perancangan serta simulasi proses penyisipan dan ekstraksi pesan pada file audio berformat WAV. Analisis dilakukan dengan membandingkan spektrum frekuensi audio sebelum dan sesudah penyisipan pesan guna menilai pengaruh metode yang digunakan terhadap kualitas audio, sehingga hasil penelitian ini diharapkan dapat memberikan gambaran mengenai efektivitas kombinasi *Spread Spectrum* dan N-LSB dalam menyembunyikan pesan pada media audio digital serta menjadi referensi bagi penelitian selanjutnya dalam bidang steganografi audio dan keamanan data digital.

3.2 Pra Penelitian

Untuk memperoleh hasil penyisipan pesan yang optimal pada steganografi audio menggunakan metode *Spread Spectrum* dan N-LSB, dilakukan serangkaian langkah pra penelitian guna memperkuat dasar teori dan menentukan arah pengembangan sistem. Tahapan ini juga bertujuan untuk mengidentifikasi kebutuhan serta parameter yang digunakan dalam penelitian. Tahapan pra penelitian yang dilakukan meliputi:

1. Mencari literatur utama yang menjadi dasar penelitian mengenai konsep dasar steganografi, khususnya steganografi audio dan teknik penyisipan pesan pada media audio digital.
2. Mengumpulkan literatur pendukung dari jurnal ilmiah, buku, dan penelitian terdahulu yang membahas penerapan steganografi pada media audio, terutama metode *Spread Spectrum* dan N-LSB.
3. Mempelajari secara mendalam prinsip kerja steganografi audio, termasuk mekanisme penyebaran bit pesan menggunakan *Spread Spectrum* dan teknik penyisipan bit pada N-LSB.
4. Memahami konsep pengolahan sinyal audio digital, representasi sampel audio, serta perubahan spektrum frekuensi yang terjadi akibat proses penyisipan pesan sebagai dasar analisis kualitas audio.

3.3 Tahapan Penelitian

Tahapan yang perlu dilakukan untuk melakukan penelitian ini yaitu sebagai berikut:

1. Proses Penyebaran dan Penyisipan
 - a. Penyebaran Pesan dengan *Spread Spectrum*
 - i. Pesan teks rahasia dikonversi ke dalam bentuk deretan bit biner agar dapat diproses secara digital pada tahap penyebaran menggunakan metode *Spread Spectrum*.
 - ii. Gunaka *PN Sequence*, yaitu urutan bilangan acak semu yang berfungsi untuk menyebarkan bit-bit pesan ke seluruh spektrum frekuensi audio.

- iii. Proses penyebaran dilakukan dengan metode DSSS, yaitu melakukan operasi XOR setiap bit pesan dengan bit pada *PN Sequence*. Hasilnya berupa sinyal tersebar (*spread signal*) yang menyerupai derau (*noise*), sehingga sulit dibedakan dari sinyal audio asli.
 - iv. Membentuk *Header* sepanjang 72-bit yang berisi informasi panjang pesan, lokasi penyisipan, nilai parameter N-LSB dan parameter LCG.
 - v. Sinyal tersebar kemudian ditambahkan ke sinyal audio asli dengan faktor skala kecil (α) agar tidak menimbulkan distorsi yang terdengar oleh telinga manusia.
- b. Penyisipan dengan Metode N-LSB
- i. Gunakan file audio berformat WAV sebagai media penampung karena bersifat tidak terkompresi, sehingga modifikasi pada bit dapat dilakukan tanpa menurunkan kualitas suara secara signifikan.
 - ii. Ubah sinyal audio menjadi bentuk sampel digital biner (8-bit atau 16-bit) agar bit-bit penyusun data dapat diakses, dianalisis, dan dimodifikasi selama proses penyisipan pesan..
 - iii. Gantikan N bit terakhir (*Least Significant Bits*) dari setiap sampel audio dengan bit-bit hasil penyebaran dari tahap *Spread Spectrum*. Penggunaan beberapa bit terakhir (N-LSB) memungkinkan kapasitas penyisipan lebih besar tanpa menurunkan kualitas suara secara signifikan.

- iv. Setelah seluruh bit pesan tersisip, ubah kembali data biner menjadi bentuk sinyal audio dan simpan sebagai file stego-audio, yaitu audio yang telah berisi pesan rahasia namun tetap terdengar seperti audio asli.

2. Proses Ekstraksi Sistem

a. Proses Ekstraksi

- i. Ambil file audio stego dan ubah ke bentuk sampel digital biner.
- ii. Membaca 72-bit awal sebagai *Header* untuk memperoleh informasi panjang pesan, lokasi penyisipan, nilai parameter N-LSB dan parameter LCG.
- iii. Ekstraksi bit pesan dengan mengambil N bit terakhir dari setiap sampel audio sesuai lokasi yang telah ditentukan.
- iv. *DeSpreading* pesan dilakukan menggunakan *PN Sequence* yang sama seperti pada proses penyisipan untuk mengembalikan sinyal pesan asli dari hasil penyebaran.
- v. Rekonstruksi pesan dilakukan dengan menyusun kembali hasil *DeSpreading* berupa deretan bit sehingga membentuk pesan teks yang dapat dibaca.

3. Proses Menghitung SD

a. Input Audio

Ambil dua file audio dengan format dan durasi yang sama.

Masukan:

- i. audio asli sebelum penyisipan pesan (A)
- ii. audio hasil penyisipan pesan (A') yang akan dibandingkan.

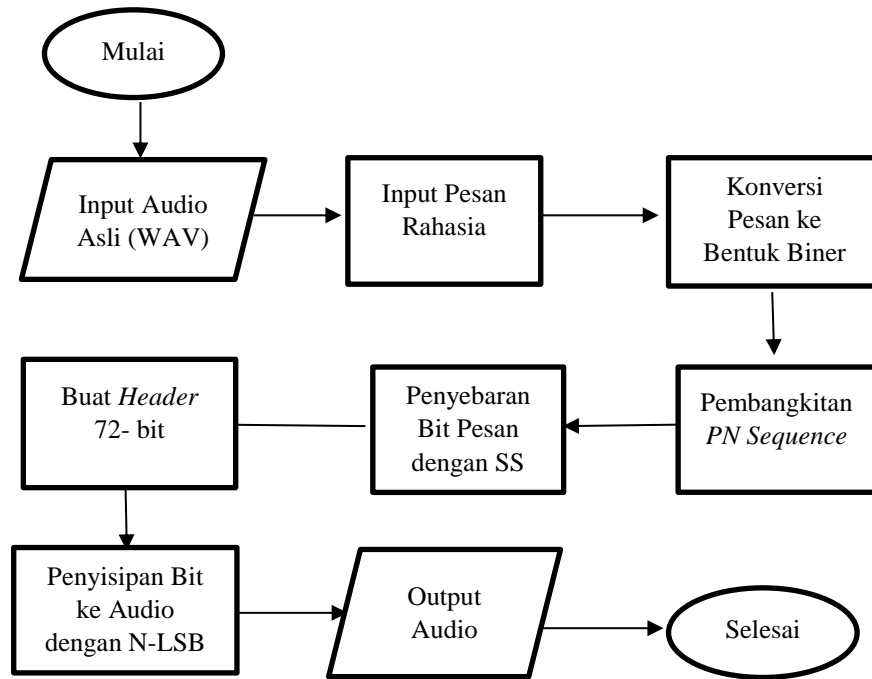
Kedua audio harus memiliki jumlah sampel yang sama agar proses perhitungan dapat dilakukan secara akurat.

b. Menghitung SD

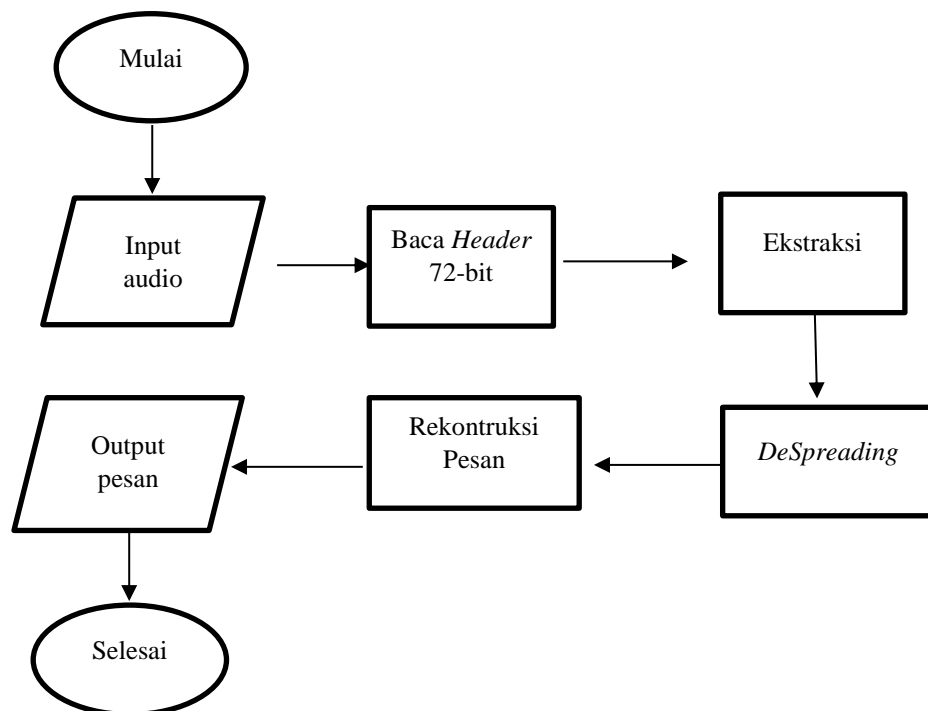
- i. Lakukan transformasi sinyal audio dari domain waktu ke domain frekuensi, misalnya menggunakan Transformasi Fourier, untuk memperoleh spektrum frekuensi dari audio asli dan audio stego.
- ii. Untuk setiap komponen frekuensi, hitung selisih spektrum antara audio asli dan audio stego dalam skala logaritmik guna mengukur perubahan spektrum yang terjadi.
- iii. Kuadratkan selisih spektrum tersebut untuk memastikan seluruh perbedaan bernilai positif.
- iv. Jumlahkan seluruh hasil kuadrat selisih spektrum frekuensi, kemudian bagi dengan jumlah total komponen frekuensi untuk memperoleh nilai rata-rata perubahan spektrum.

c. Hasil Akhir

- i. Menampilkan spektrum frekuensi audio asli dan audio stego sebagai perbandingan untuk melihat perubahan akibat penyisipan, serta memastikan kualitas audio tetap terjaga dan tidak berbeda signifikan.
- ii. Nilai SD yang diperoleh merupakan nilai yang menunjukkan tingkat perubahan spektrum frekuensi antara audio asli dan audio stego. Semakin kecil nilai SD, semakin kecil perubahan spektrum yang terjadi, sehingga kualitas audio hasil penyisipan semakin mendekati audio asli.



Gambar 3.1 Flowchart Proses Penyisipan



Gambar 3.2 Flowchart Proses Ekstraksi

BAB IV

HASIL DAN PEMBAHASAN

Setelah menyelesaikan tahapan perancangan sistem yang dijabarkan dalam Bab III, langkah selanjutnya adalah melakukan pembahasan serta analisis terhadap data dan hasil implementasi sistem steganografi audio yang telah dibangun. Pembahasan ini difokuskan pada penerapan kombinasi metode *Spread Spectrum* dan *N-Least Significant Bit* (N-LSB) dalam proses penyisipan dan ekstraksi pesan pada file audio WAV. Melalui hasil pembahasan tersebut, dapat diketahui bagaimana kinerja sistem dalam menyembunyikan dan mengekstraksi pesan, serta sejauh mana proses penyisipan memengaruhi kualitas audio. Selain itu, analisis ini juga bertujuan untuk mengevaluasi perbedaan antara audio asli dan audio stego berdasarkan parameter *Spectral Distortion*, sehingga dapat dinilai efektivitas metode yang digunakan dalam menjaga kerahasiaan pesan tanpa menurunkan kualitas audio secara signifikan.

4.1 Proses Penyisipan Pesan dengan *Spread Spectrum*

1. Konversi Pesan Rahasia ke Biner

Pesan yang akan disampaikan adalah “UIN”, dengan menggunakan kode ASCII.

Tabel 4.1 Representasi ASCII ke Biner 8-bit

Karakter	ASCII	Biner (8-bit)
U	85	01010101
I	73	01001001
N	78	01001110

Berdasarkan hasil konversi pada Tabel 4.1, diperoleh total data pesan sebanyak 24 bit. Data biner tersebut kemudian digunakan sebagai masukan pada tahap penyebaran (*spreading*) menggunakan metode DSSS.

2. Penyebaran Pesan dengan *Spread Spectrum* (DSSS)

Penyebaran pesan dengan metode DSSS bertujuan untuk menyamarkan pesan rahasia dengan cara memperluas bit pesan sehingga menyerupai sinyal derau (*noise*). Pada tahap ini, pesan biner tidak langsung dimodulasi dengan *PN Sequence*, melainkan terlebih dahulu diperluas melalui proses *Spreading* (*chip expansion*).

Pada penelitian ini digunakan *Spreading* factor sebesar 4, yang berarti setiap satu bit pesan diperluas menjadi empat bit chip. Aturan *Spreading* yang digunakan adalah:

Bit 0 direpresentasikan sebagai 0000

Bit 1 direpresentasikan sebagai 1111

Aturan tersebut diterapkan secara konsisten pada seluruh bit pesan.

Sebagai contoh, proses *Spreading* pada karakter “UIN” dengan representasi biner dilakukan dengan mengganti setiap bit sesuai aturan *Spreading*, sehingga diperoleh hasil sebagai berikut:

U : 00001111000011110000111100001111

I : 00001111000000001111000000001111

N : 00001111000000001111111111110000

Proses *Spreading* yang sama diterapkan setiap karakter U, I dan N, sehingga setiap karakter yang semula memiliki panjang 8-bit berubah menjadi 32-bit.

Karena pesan “UIN” terdiri dari tiga karakter, maka total panjang pesan setelah proses *Spreading* menjadi 96-bit.

Tahap ini hanya berfungsi untuk melakukan perluasan bit pesan dan belum melibatkan proses pembangkitan maupun penerapan *PN Sequence*.

3. Pembangkitan *PN Sequence* Menggunakan LCG Berbasis Kata Kunci

Setelah proses *Spreading* selesai, tahap berikutnya adalah pembangkitan *PN Sequence* yang digunakan untuk memodulasi sinyal hasil *Spreading*. Pada penelitian ini, *PN Sequence* dibangkitkan menggunakan metode LCG.

LCG menghasilkan bilangan acak semu berdasarkan Persamaan (2.1).

Nilai awal (*seed*) LCG diperoleh dari kata kunci yang telah ditentukan. Pada penelitian ini, kata kunci yang digunakan adalah “kata”. Kata kunci tersebut dikonversi ke dalam bentuk biner menggunakan representasi ASCII 8-bit, kemudian diproses menggunakan operasi *XOR* secara berurutan untuk menghasilkan nilai *seed*. Nilai *seed* yang diperoleh kemudian digunakan sebagai nilai awal pada proses iterasi LCG.

Proses pembentukan *seed* dari kata kunci “kata” adalah sebagai berikut:

Langkah 1	: $k \oplus a$
k	:01101011
a	:01100001
XOR	: 00001010
Langkah 2	: Hasil $\oplus t$
Hasil	:00001010
t	:01110100
XOR	:01111110
Langkah 3	: Hasil $\oplus a$
Hasil	:01111110
a	:01100001
XOR	:00011111

Hasil akhir operasi *XOR* tersebut menghasilkan nilai biner 00011111, yang kemudian dikonversi ke dalam bentuk desimal untuk digunakan sebagai *seed* awal (X_0) pada proses pembangkitan bilangan acak menggunakan LCG.

Nilai biner 00011111 dikonversi ke desimal sehingga diperoleh:

$$X_0 = 31$$

Nilai ini digunakan sebagai *seed* awal LCG.

Berdasarkan *seed* awal tersebut, bilangan acak semu dibangkitkan menggunakan persamaan LCG dengan parameter a , c , dan m yang telah ditentukan. Penentuan nilai parameter a (*multiplier*), c (*increment*), dan m (modulus) pada algoritma LCG dilakukan untuk memastikan bahwa generator memiliki periode penuh (*full period*). Berdasarkan Teorema Hull-Dobell, sebuah LCG akan memiliki periode penuh jika dan hanya jika memenuhi syarat-syarat matematis tertentu. Berikut adalah pembuktian kelayakan parameter yang digunakan dalam penelitian ini sebagai berikut:

$$a = 17$$

$$c = 43$$

$$m = 256$$

a. Syarat Modulus (m)

- i. Modulus dipilih sebesar $m = 256$ (2^8).
- ii. Pemilihan ini dilakukan agar setiap bilangan acak pada rentang 0–255 dapat direpresentasikan dalam bentuk biner 8-bit (1 byte).

b. Syarat Increment (c)

- i. Syarat: c dan m harus relatif prima, yaitu $\text{FPB}(c, m) = 1$.

- ii. Faktor prima dari 256 adalah {2}, sedangkan faktor prima dari 43 adalah {43}. Karena tidak memiliki faktor prima yang sama, maka diperoleh $\text{FPB}(43,256) = 1$.
- c. Syarat Multiplier (a)
- i. Syarat A: ($a - 1$) harus dibagi oleh semua faktor prima dari m .
- 1) $a - 1 = 17 - 1 = 16$
 - 2) Faktor prima dari $m(256)$ adalah 2.
 - 3) Uji pembagian: $\frac{16}{2} = 8$ (Habis dibagi tanpa sisa)
- ii. Syarat B: Jika m adalah kelipatan 4, maka ($a - 1$) juga harus habis dibagi 4.
- 1) $m = 256$ merupakan kelipatan 4 ($\frac{256}{4} = 64$).
 - 2) $(a - 1) = 16$.
 - 3) Uji pembagian: $\frac{16}{4} = 4$ (Habis dibagi tanpa sisa).

Melalui pemenuhan ketiga syarat di atas, maka parameter $a = 17, c = 43$, dan $m = 256$ secara matematis memungkinkan deret bilangan acak semu memiliki periode yang panjang sebelum mengalami pengulangan. Hal ini krusial untuk menghasilkan *PN Sequence* yang memiliki distribusi yang baik untuk menyamarkan pesan teks ke dalam spektrum frekuensi audio. Proses pembangkitan bilangan acak semu dilakukan secara iteratif menggunakan persamaan LCG.

Iterasi ke-1:

$$X_1 = (17 \times 31 + 43) \text{ mod } 256$$

$$X_1 = (527 + 43) \text{ mod } 256 = 570 \text{ mod } 256 = 58$$

Iterasi ke-2:

$$X_2 = (17 \times 58 + 43) \bmod 256$$

$$X_2 = (986 + 43) \bmod 256 = 1029 \bmod 256 = 5$$

Iterasi ke-3:

$$X_3 = (17 \times 5 + 43) \bmod 256$$

$$X_3 = (85 + 43) \bmod 256 = 128$$

Iterasi ke-4:

$$X_4 = (17 \times 128 + 43) \bmod 256$$

$$X_4 = (2176 + 43) \bmod 256 = 2219 \bmod 256 = 171$$

Nilai hasil LCG kemudian dikonversi ke dalam bentuk biner 8-bit untuk membentuk *PN Sequence*.

Tabel 4.2 Representasi Nilai LCG dalam Bentuk Biner 8-bit

Iterasi	Nilai Desimal	Biner 8-bit
X_1	58	00111010
X_2	5	00000101
X_3	128	10000000
X_4	171	10101011
...
X_{12}	195	11000011

Berdasarkan hasil pembangkitan menggunakan algoritma LCG yang ditunjukkan pada Tabel 4.2, setiap bilangan desimal hasil LCG dikonversi ke dalam bentuk biner 8-bit. Seluruh bit dari hasil konversi tersebut kemudian digabungkan secara berurutan membentuk rangkaian bit kontinu. Rangkaian bit inilah yang digunakan sebagai *PN Sequence* pada proses modulasi DSSS. Proses iterasi LCG dilakukan berulang hingga panjang *PN Sequence* sama dengan panjang sinyal hasil *Spreading*, yaitu 96-bit. maka diperlukan sebanyak 12 iterasi untuk memenuhi kebutuhan panjang tersebut. Berikut adalah rangkaian bit *PN Sequence* yang dihasilkan:

00111010 00000101 10000000 10101011
 10000110 00010001 01001100 00110111
 11010010 00011101 00011000 11000011

a. Hubungan *PN Sequence* dengan Modulasi DSSS

PN Sequence yang dihasilkan dari proses LCG ini selanjutnya digunakan pada tahap modulasi DSSS, yaitu dengan mengoperasikan XOR antara *PN Sequence* dan sinyal hasil *Spreading* pesan. Dengan demikian, *PN Sequence* berfungsi sebagai sinyal pengacak yang menyamarkan pola bit pesan.

b. Modulasi DSSS menggunakan Operasi XOR

Modulasi DSSS dilakukan dengan operasi XOR antara sinyal hasil *Spreading* dan *PN Sequence*.

i. Modulasi karakter U

Spreading U:

00001111000011110000111100001111

PN Sequence:

00111010000001011000000010101011

XOR:

00001111 00001111 00001111 00001111

00111010 00000101 10000000 10101011 \oplus

00110101 00001010 10001111 10100100

ii. Modulasi Karakter I

Spreading I:

00001111000000001111000000001111

PN Sequence:

10000110000100010100110000110111

XOR:

00001111 00000000 11110000 00001111

10000110 00010001 01001100 00110111 \oplus

10001001 00010001 10111100 0011100

iii. Modulasi karakter N

Spreading N:

00001111000000001111111111110000

PN Sequence:

110100100001110100011000110000110

XOR:

00001111 00000000 11111111 11110000

11010010 00011101 00011000 11000011 \oplus

11011101 00011101 11100111 00110011

Hasil Modulasi DSSS pesan 'UIN' adalah:

00110101 00001010 10001111 10100100

10001001 00010001 10111100 00111000

11011101 00011101 11100111 00110011

Total panjang sinyal DSSS adalah 96-bit. Sinyal inilah yang digunakan sebagai data pesan pada tahap penyisipan ke dalam media audio.

PN Sequence yang dihasilkan bersifat deterministik, sehingga pada proses ekstraksi pesan, *PN Sequence* yang sama dapat dibangkitkan

kembali dengan menggunakan kata kunci dan parameter LCG yang sama.

2. Membuat *Header* 72 – bit

Pada tahap ini dilakukan pembentukan *Header* sepanjang 72-bit yang digunakan untuk menyimpan informasi penting dalam proses ekstraksi pesan. *Header* terdiri dari beberapa bagian, yaitu panjang pesan, posisi penyisipan, nilai parameter N-LSB, serta parameter LCG. Informasi tersebut diperlukan agar proses ekstraksi dapat dilakukan dengan tepat.

Proses Pembentukan *Header* dilakukan sebagai berikut:

a. Konversi jumlah bit pesan ke biner (16-bit)

Panjang pesan sebesar 96 dikonversi ke bentuk biner:

96 → 00000000 01100000

b. Representasi posisi penyisipan (8-bit)

Posisi penyisipan ditengah direpresentasikan dengan kode:

01 → 00000001

c. Konversi nilai N-LSB (8-bit)

Nilai parameter N-LSB sebesar 2 dikonversi ke biner:

2 → 00000010

d. Parameter LCG (32-bit)

Parameter LCG yang digunakan adalah:

Seed (31) → 00011111

a (17) → 00010001

c (43) → 00101011

m (256) → 00000001 00000000

- e. Gabung menjadi satu

Sehingga diperoleh *Header* sebagai berikut:

```
00000000 01100000 00000001 00000010 00011111 00010001
00101011 00000001 00000000
```

Header yang telah terbentuk ini kemudian digunakan sebagai informasi awal yang akan disisipkan ke dalam media audio sebelum proses penyisipan data pesan dilakukan.

4.2 Proses Penyisipan Pesan Menggunakan Metode 2-LSB

Pada subbab ini dibahas proses penyisipan sinyal pesan hasil modulasi DSSS ke dalam media audio menggunakan metode *2-Least Significant Bit (2-LSB)*. Proses penyisipan dilakukan dengan cara mengganti dua bit paling rendah dari setiap sampel audio dengan bit-bit pesan hasil modulasi DSSS. Seluruh tahapan dilakukan secara manual untuk menunjukkan prinsip kerja metode secara jelas.

Media audio yang digunakan berupa file audio .wav dengan format PCM, sehingga nilai amplitudo sampel dapat direpresentasikan dalam bentuk bilangan bulat dan biner. Format ini memudahkan proses pengolahan dan penyisipan data pada tingkat bit.

1. Penyiapan Data Audio

Posisi awal penyisipan pesan pada file audio ditentukan dengan membagi sinyal audio menjadi dua bagian yang sama besar berdasarkan jumlah total sampel. Penyisipan dilakukan pada bagian kedua sinyal audio, yaitu mulai dari titik tengah audio, untuk menghindari bagian awal audio yang umumnya mengandung nilai nol atau transisi awal sinyal.

Secara matematis, posisi awal penyisipan ditentukan dengan persamaan:

$$k = \frac{N}{2} \quad (4.2)$$

dengan k sebagai indeks awal penyisipan dan N sebagai jumlah total sampel audio.

Berdasarkan hasil pembacaan file audio yang digunakan pada penelitian ini, diperoleh jumlah total sampel audio sebesar $N = 544.183$ sampel. Dengan demikian, posisi awal penyisipan pesan dihitung sebagai berikut:

$$k = \frac{544.183}{2} = 272.091,5$$

Karena indeks sampel audio harus berupa bilangan bulat, maka nilai tersebut dibulatkan menjadi 272091. Dengan demikian, proses penyisipan pesan dimulai dari sampel audio ke-272091 hingga seluruh sinyal hasil modulasi *Spread Spectrum* berhasil disisipkan ke dalam sinyal audio. Penentuan posisi awal bertujuan agar proses penyisipan berlangsung secara terstruktur. Panjang sinyal pesan hasil modulasi DSSS adalah 96-bit. Karena metode penyisipan yang digunakan adalah 2-LSB, maka setiap satu sampel audio digunakan untuk menyisipkan dua bit pesan, sehingga jumlah sampel audio yang dibutuhkan untuk menyisipkan seluruh pesan adalah sebanyak:

$$96 \div 2 = 48 \text{ sampel}$$

Sebanyak 48 sampel audio digunakan dalam proses penyisipan. Berdasarkan hasil pembacaan data audio, nilai amplitudo dari 48 sampel audio tersebut adalah sebagai berikut:

$$\begin{bmatrix} 2466 & 3322 & 3090 & 1479 & 443 & 1069 & 2447 & 3988 \\ 3815 & 4089 & 3072 & 2926 & 3253 & 3441 & 2688 & 735 \\ -1123 & -1620 & -1101 & -193 & 1268 & 2613 & 2675 & 2151 \\ 2065 & 1352 & -729 & -2311 & -1802 & -430 & -292 & -1399 \\ -1770 & -745 & 136 & 204 & 454 & 1408 & 2019 & 1196 \\ -368 & -1377 & -1453 & -548 & 915 & 1192 & -903 & -3227 \end{bmatrix}$$

Sampel audio yang digunakan dalam proses penyisipan memiliki nilai amplitudo yang bervariasi sesuai dengan karakteristik sinyal audio asli. Metode 2-LSB diterapkan dengan memodifikasi dua bit paling rendah dari representasi biner PCM 16-bit pada setiap sampel audio. Proses ini tidak memengaruhi bit-bit signifikan lainnya sehingga karakteristik utama sinyal audio tetap terjaga dan distorsi yang ditimbulkan bersifat minimal. Sebagai contoh, salah satu sampel audio dengan nilai amplitudo 2466 direpresentasikan dalam format PCM 16-bit sebagai 0000100110100010.

2. Penyiapan Bit Pesan Hasil Modulasi DSSS

Bit pesan yang akan disisipkan merupakan hasil modulasi DSSS dari pesan rahasia “UIN” dengan panjang total 96-bit, yaitu:

```
00110101 00001010 10001111 10100100
10001001 00010001 10111100 00111000
11011101 00011101 11100111 00110011
```

Bit-bit pesan tersebut dibaca secara berurutan dan dikelompokkan menjadi dua bit untuk setiap sampel audio sesuai dengan metode 2-LSB. Pengelompokan ini dilakukan untuk menyesuaikan jumlah bit yang disisipkan pada setiap sampel audio.

3. Penyisipan *Header*

Pada tahap ini dilakukan penyisipan *Header* ke dalam sampel audio menggunakan metode 2-LSB. *Header* yang digunakan adalah:

```
00000000 01100000 00000001 00000010 00011111 00010001
00101011 00000001 00000000
```

Karena metode yang digunakan adalah 2-LSB, maka setiap sampel audio akan menyimpan 2-bit data. Oleh karena itu, *Header* sepanjang 72-bit akan membutuhkan:

$$\frac{72}{2} = 36 \text{ sampel audio}$$

Sebanyak 36 sampel audio digunakan dalam proses penyisipan. Berdasarkan hasil pembacaan data audio, nilai amplitudo dari 36 sampel audio tersebut adalah sebagai berikut:

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

Selanjutnya yaitu proses penyisipan *Headernya* sebagai berikut:

Sampel ke – 1

Sampel asli : 0 → 0000000000000000

Bit pesan : 00

Sampel stego : 0000000000000000 → 0

Sampel ke – 2

Sampel asli : 0 → 0000000000000000

Bit pesan : 00

Sampel stego : 0000000000000000 → 0

Sampel ke – 3

Sampel asli : 0 → 0000000000000000

Bit pesan : 00

Sampel stego : 0000000000000000 → 0

Proses penyisipan ini dilanjutkan secara berurutan hingga seluruh 36 sampel audio digunakan dan seluruh 72-bit *Header* berhasil disisipkan ke dalam media audio.

4. Hasil penyisipan *Header*

Setelah proses penyisipan *Header* selesai, diperoleh perubahan nilai amplitudo yang sangat kecil, yaitu berada pada rentang 0 hingga 3 dari nilai sampel audio asli. Hal ini terjadi karena penyisipan dilakukan menggunakan metode 2-LSB yang hanya memodifikasi dua bit paling rendah pada setiap sampel audio. Oleh karena itu, perubahan yang dihasilkan tidak memberikan pengaruh terhadap kualitas audio.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 & 1 & 3 & 3 & 0 & 1 & 0 & 1 \\ 0 & 2 & 2 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Setelah penyisipan *Header* selesai, tahap selanjutnya adalah menyisipkan bit pesan ke dalam sampel audio sesuai dengan posisi yang telah ditentukan.

5. Proses Penyisipan Bit Pesan ke dalam Sampel Audio

Proses penyisipan dilakukan secara berurutan dengan tahapan sebagai berikut:

- a. Mengambil dua bit pesan hasil modulasi DSSS
- b. Menghapus dua bit paling rendah (2-LSB) dari sampel audio
- c. Mengganti dua bit tersebut dengan dua bit pesan

Karena sampel audio yang digunakan memiliki nilai amplitudo yang bervariasi, maka proses penyisipan hanya memodifikasi dua bit paling rendah dari setiap sampel tanpa mengubah bit-bit signifikan lainnya. Dengan demikian, karakteristik utama sinyal audio tetap terjaga dan

perubahan amplitudo yang terjadi bersifat minimal. Hal ini membantu mempertahankan kualitas audio setelah proses penyisipan pesan.

Proses Penyisipan:

Sampel ke – 1	:
Sampel asli	: 2466 → 0000100110100010
Bit pesan	: 00
Sampel stego	: 0000100110100000 → 2464
Sampel ke – 2	:
Sampel asli	: 3322 → 0000110011111010
Bit pesan	: 11
Sampel stego	: 0000110011111011 → 3323
Sampel ke – 3	:
Sampel asli	: 3090 → 0000110000010010
Bit pesan	: 01
Sampel stego	: 0000110000010001 → 3089
Sampel ke – 4	:
Sampel asli	: 1479 → 0000010111000111
Bit pesan	: 10
Sampel stego	: 0000010111000110 → 1478

Proses penyisipan ini dilanjutkan secara berurutan hingga seluruh 48 sampel audio digunakan dan seluruh 96 –bit pesan berhasil disisipkan ke dalam media audio. Dengan demikian, seluruh informasi pesan berhasil ditanamkan ke dalam media audio tanpa mengganggu karakteristik utama sinyal audio.

6. Hasil Penyisipan Pesan

Setelah proses penyisipan selesai, diperoleh audio stego dengan perubahan nilai amplitudo yang sangat kecil, yaitu hanya mengalami selisih sebesar 0 hingga 3 dari nilai sampel audio asli. Perubahan ini terjadi karena proses penyisipan hanya memodifikasi dua bit paling rendah ($2 - LSB$) pada setiap sampel audio. sehingga kualitas audio secara auditori tetap terjaga.

Berdasarkan hasil penyisipan menggunakan metode 2-LSB pada 48 sampel audio, diperoleh data sampel audio stego yang telah mengandung bit pesan hasil modulasi *Spread Spectrum* sebagai berikut:

$$\begin{bmatrix} 2464 & 3323 & 3089 & 1478 & 444 & 1068 & 2446 & 3988 \\ 4815 & 4089 & 3072 & 2926 & 3253 & 3441 & 2688 & 735 \\ -1124 & -1620 & -1103 & -193 & 1268 & 2612 & 2675 & 2152 \\ 2064 & 1352 & -728 & -2308 & -1804 & -428 & -292 & -1400 \\ -1772 & -744 & 136 & 204 & 452 & 1408 & 2020 & 1196 \\ -368 & -1376 & -1452 & -548 & 916 & 1192 & -904 & -3228 \end{bmatrix}$$

Dengan demikian, hasil penyisipan menunjukkan bahwa metode 2-LSB mampu menyisipkan pesan ke dalam media audio dengan perubahan amplitudo yang minimal serta tetap menjaga kualitas audio, sehingga audio stego yang dihasilkan tetap menyerupai audio asli secara auditori.

4.3 Proses Ekstraksi

Setelah proses penyisipan pesan menggunakan metode *Spread Spectrum* dan 2-LSB selesai, tahap selanjutnya adalah ekstraksi pesan dari audio *stego* untuk mendapatkan kembali pesan asli. Proses ekstraksi dilakukan dengan urutan yang terbalik dari proses penyisipan, yaitu dimulai dari pengambilan bit-bit tersembunyi dari audio *stego*, kemudian melakukan *DeSpreading* menggunakan *PN Sequence* yang sama, dan akhirnya mengkonversi kembali ke bentuk pesan teks.

1. Ekstraksi Bit dari Audio Stego dengan Metode 2-LSB

a. Membaca *Header*

Sebanyak 36 sampel pertama dari audio stego diambil sesuai dengan jumlah sampel yang digunakan dalam proses penyisipan. Data amplitudo sampel pertama audio stego adalah sebagai berikut:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 & 1 & 3 & 3 & 0 & 1 & 0 & 1 \\ 0 & 2 & 2 & 3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

b. Konversi Sampel ke Biner dan Ekstraksi 2 – LSB

Setiap sampel dikonversi ke dalam representasi biner 16-bit, kemudian dua bit paling rendah (*Least Significant Bits*) diambil. Contoh proses untuk beberapa sampel awal:

Sampel ke – 1:

$$1 \rightarrow 0000000000000000 \rightarrow \text{LSB: } 00$$

Sampel ke – 2:

$$2 \rightarrow 0000000000000000 \rightarrow \text{LSB: } 00$$

Sampel ke – 3:

$$0 \rightarrow 0000000000000000 \rightarrow \text{LSB: } 00$$

Proses ini dilanjutkan hingga sampel ke – 36 yang akan menghasilkan deretan bit sebagai berikut:

00000000 01100000 00000001 00000010 00011111 00010001
00101011 00000001 00000000

Deretan bit ini memiliki panjang 72-bit, sesuai dengan panjang *Header* yang disisipkan, yang terdiri dari informasi panjang pesan, posisi penyisipan, nilai parameter N-LSB, serta bit cadangan.

Jika dikonversi ke dalam bentuk bilangan integer, maka diperoleh:

00000000 01100000 = 96 (panjang bit pesan)

00000001 = 01 (kode posisi penyisipan)

00000010 = 2 (nilai N – LSB)

00011111 = 31 (*seed* LCG)

00010001 = 17 (parameter a)

00101011 = 43 (parameter c)

00000001 00000000 = 256 (parameter m)

Kode posisi penyisipan bernilai 01 menunjukkan bahwa data pesan disisipkan pada bagian tengah audio. Kode ini digunakan oleh sistem untuk menentukan lokasi pengambilan data pesan pada saat proses ekstraksi berlangsung. Dengan demikian, proses ekstraksi dapat dilakukan sesuai dengan posisi penyisipan yang digunakan sebelumnya.

Selain itu, parameter LCG yang diperoleh dari *Header* digunakan kembali untuk membangkitkan deret bilangan semu yang sama seperti pada saat proses penyisipan, sehingga proses *DeSpreading* dapat dilakukan secara tepat dan pesan asli dapat direkonstruksi dengan benar.

c. Pengambilan Sampel Audio Stego

Setelah *Header* dibaca, proses ekstraksi dilanjutkan dengan mengambil 48 sampel audio stego pada bagian tengah sesuai informasi posisi penyisipan, berdasarkan panjang pesan 96-bit dan penggunaan metode 2-LSB. Data amplitudo sampel audio stego adalah sebagai berikut:

2464	3323	3089	1478	444	1068	2446	3988
4815	4089	3072	2926	3253	3441	2688	735
-1124	-1620	-1103	-193	1268	2612	2675	2152
2064	1352	-728	-2308	-1804	-428	-292	-1400
-1772	-744	136	204	452	1408	2020	1196
-368	-1376	-1452	-548	916	1192	-904	-3228

d. Konversi Sampel ke Biner dan Ekstraksi 2 – LSB

Setiap sampel dikonversi ke dalam representasi biner 16-bit, kemudian dua bit paling rendah (*Least Significant Bits*) diambil. Contoh proses untuk beberapa sampel awal:

Sampel ke – 1:

$$2464 \rightarrow 0000100110100000 \rightarrow \text{LSB: } 00$$

Sampel ke – 2:

$$3323 \rightarrow 0000110011111011 \rightarrow \text{LSB: } 11$$

Sampel ke – 3:

$$3089 \rightarrow 0000110000010001 \rightarrow \text{LSB: } 01$$

Sampel ke – 4:

$$1478 \rightarrow 0000010111000110 \rightarrow \text{LSB: } 10$$

Sampel ke – 5:

$$444 \rightarrow 0000000110111100 \rightarrow \text{LSB: } 00$$

Proses ini dilanjutkan hingga sampel ke – 48 yang akan menghasilkan deretan bit sebagai berikut:

00110101 00001010 10001111 10100100

10001001 00010001 10111100 00111000

11011101 00011101 11100111 00110011

Deretan bit ini memiliki panjang 96 bit, sesuai dengan panjang sinyal DSSS hasil penyisipan.

1. *DeSpreading Menggunakan PN Sequence*

a. *Pembangkitan Kembali PN Sequence*

PN Sequence dibangkitkan kembali menggunakan LCG dengan parameter dan kata kunci yang sama seperti pada proses penyisipan, yaitu:

Kata kunci: "kata"

Seed hasil XOR karakter "kata": 00011111 (desimal 31)

Parameter LCG: $a = 17, c = 43, m = 256$

Hasil pembangkitan *PN Sequence* (96 bit) adalah:

00111010 00000101 10000000 10101011

10000110 00010001 01001100 00110111

11010010 00011101 00011000 11000011

b. *Operasi XOR untuk DeSpreading*

Bit hasil ekstraksi (96- bit) dibagi menjadi 3 blok masing-masing 32-bit sesuai dengan pengelompokan pada proses penyisipan. Setiap blok di operasikan XOR dengan *PN Sequence* yang sesuai:

Blok 1:

00110101 00001010 10001111 10100100 \oplus

00111010 00000101 10000000 10101011 =

00001111 00001111 00001111 00001111

Blok 2:

10001001 00010001 10111100 00111000 \oplus

10000110 00010001 01001100 00110111 =

00001111 00000000 11110000 00001111

Blok 3:

11011101 00011101 11100111 00110011 \oplus

11010010 00011101 00011000 11000011 =

00001111 00000000 11111111 11110000

Hasil XOR tersebut merupakan sinyal tersebar (*spread signal*) untuk masing-masing karakter pesan.

2. *DeSpreading Chip expansion* (Faktor 4)

Setiap blok hasil XOR (32-bit) merupakan hasil *Spreading* dengan *Spreading factor* sebesar 4 sesuai parameter yang digunakan pada proses penyisipan. Nilai parameter tersebut digunakan kembali pada proses ekstraksi sehingga proses *DeSpreading* dapat dilakukan dengan mengelompokkan setiap 4 bit menjadi 1 bit berdasarkan aturan:

0000 \rightarrow *bit 0*

1111 \rightarrow *bit 1*

Karakter U (blok 1):

0000 1111 0000 1111 0000 1111 0000 1111

Hasil pengelompokan : 0,1, 0, 1, 0, 1, 0, 1

Biner : 01010101

(ASCII 85 = 'U')

Karakter I (blok 2):

0000 1111 0000 0000 1111 0000 0000 1111

Hasil pengelompokan : 0, 1, 0, 0, 1, 0, 0, 1

Biner : 01001001

(ASCII 73 = 'I')

Karakter N (blok 3):

0000 1111 0000 0000 1111 1111 1111 0000

Hasil pengelompokan : 0, 1, 0, 0, 1, 1, 1, 0

Biner : 01001110

(ASCII 78 = 'N')

3. Hasil Akhir Ekstraksi

Pesan yang berhasil diekstraksi dari audio stego adalah tiga karakter: U, I, N, yang membentuk pesan teks "UIN". Hasil ini sama persis dengan pesan asli yang disisipkan pada proses penyisipan.

4.4 Evaluasi Kualitas Audio dengan *Spectral Distortion*

SD digunakan untuk mengukur tingkat perbedaan spektrum frekuensi antara sinyal audio asli dan sinyal audio stego. Rumus SD adalah:

$$SD = \sqrt{\frac{1}{k} \sum_{k=1}^K \left[10 \log_{10} \left(\frac{S_o(k)}{S_s(k)} \right) \right]^2} \quad (4.3)$$

Dimana:

K : Jumlah komponen frekuensi (bit frekuensi).

$S_o(k)$: Nilai spektrum amplitudo sinyal audio asli pada frekuensi ke k .

$S_s(k)$: Nilai spektrum amplitudo sinyal audio stego pada frekuensi ke k .

1. Perhitungan SD

Untuk menyederhanakan perhitungan manual, akan dihitung SD menggunakan 4 sampel pertama sebagai representasi dengan 3 komponen frekuensi ($K = 3$). Data amplitudo dalam domain frekuensi diperoleh melalui Transformasi *Fourier Diskrit* (DFT) sederhana. Pendekatan ini digunakan agar proses perhitungan dapat dilakukan secara lebih mudah.

- a. Data Amplitudo Audio Asli (S_o) dan Audio Stego (S_s)

Tabel 4.3 merupakan data empat sampel pada audio asli dan stego.

Tabel 4.3 Data 4 Sampel Pertama Audio Asli dan Audio Stego

Sampel ke-	Audio Asli ($x[n]$)	Audio Stego ($y[n]$)
1	2466	2464
2	3322	3323
3	3090	3089
4	1479	1477

- b. Transformasi ke Domain Frekuensi (DFT 4 titik)

Rumus DFT :

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i\left(\frac{2\pi}{N}\right)kn}, \text{ untuk } n = 0,1,2,3 \quad (4.3)$$

Untuk $N = 4$, faktor putar $e^{\frac{-i2\pi}{4}} = e^{\frac{-i\pi}{2}} = -i$

Untuk sinyal asli $x[n] = [2466, 3322, 3090, 1479]$:

$K = 0$:

$$X[0] = 2466 + 3322 + 3090 + 1479 = 10357$$

$|X[0]| = 10357$ (Tidak digunakan dalam perhitungan SD)

$K = 1$:

$$X[1] = 2466(1) + 3322(-i) + 3090(-1) + 1479(i)$$

$$X[1] = (2466 - 3090) + (-3322 + 1479)i$$

$$X[1] = -624 - 1843i$$

Magnitudo:

$$|X[1]| = \sqrt{(-624)^2 + (-1843)^2}$$

$$|X[1]| = \sqrt{389376 + 3396649}$$

$$|X[1]| = \sqrt{3786.025}$$

$$|X[1]| = 1945,77$$

$K = 2$:

$$X[2] = 2466(1) + 3322(-1) + 3090(1) + 1479(-1)$$

$$X[2] = (2466 + 3090) - (3322 + 1479)$$

$$X[2] = 5556 - 4801$$

$$X[2] = 755$$

$K = 3$:

$$X[3] = 2466(1) + 3322(i) + 3090(-1) + 1479(-i)$$

$$X[3] = (2466 - 3090) + (3322 - 1479)i$$

$$X[3] = -624 + -1843i$$

Magnitudo:

$$|X[3]| = \sqrt{(-624)^2 + (-1843)^2}$$

$$|X[3]| = \sqrt{389376 + 3396649}$$

$$|X[3]| = \sqrt{3786025}$$

$$|X[3]| = 1945,77$$

Untuk sinyal stego $y[n] = [2464,3323,3089,1477]$:

$K = 1$:

$$Y[1] = 2464(1) + 3323(i) + 3089(-1) + 1477(-i)$$

$$Y[1] = (2464 - 3089) + (3323 - 1477)i$$

$$Y[1] = -625 + (1846)i$$

Magnitudo:

$$|Y[1]| = \sqrt{(-625)^2 + (1846)^2}$$

$$|Y[1]| = \sqrt{390625 + 3407716}$$

$$|Y[1]| = 1948,93$$

$K = 2:$

$$Y[2] = 2464(1) + 3323(-1) + 3089(1) + 1477(-1)$$

$$Y[2] = (2464 + 3089) - (3323 + 1477)$$

$$Y[2] = 5553 - 4800$$

$$Y[2] = 753$$

$K = 3:$

$$Y[3] = 2464(1) + 3323(i) + 3089(-1) + 1477(-i)$$

$$Y[3] = (2464 - 3089) + (3323 - 1477)i$$

$$Y[3] = -625 + 1846i$$

Magnitudo:

$$|Y[3]| = \sqrt{(-625)^2 + (1846)^2}$$

$$|Y[3]| = \sqrt{390625 + 3407716}$$

$$|Y[3]| = \sqrt{3798341}$$

$$|Y[3]| = 1948,93$$

c. Rekapitulasi *Spectrum* Amplitudo

Tabel 4.4 merupakan rekapitulasi *spectrum* amplitudo dari contoh perhitungan sebelumnya

Tabel 4.4 Rekapitulasi *Spectrum* Amplitudo

Frekuensi (k)	$S_o(k)$ Audio Asli	$S_s(k)$ Audio Stego
1	1945,77	1948,93
2	755	753
3	1945,77	1948,93

2. Perhitungan Rasio *Spectrum* $\frac{S_o(k)}{S_s(k)}$

Untuk $k = 1:$

$$\frac{S_o(1)}{S_s(1)} = \frac{1945,77}{1948,93} = 0,99838$$

Untuk $k = 2$:

$$\frac{S_o(2)}{S_s(2)} = \frac{755}{753} = 1,00266$$

Untuk $k = 3$:

$$\frac{S_o(3)}{S_s(3)} = \frac{1945,77}{1948,93} = 0,99838$$

3. Perhitungan $10 \log_{10} \left(\frac{S_o(k)}{S_s(k)} \right)$

Untuk $k = 1$:

$$10 \log_{10}(0,99838)$$

Langkah perhitungan:

$$\log_{10}(0,99838) = \log_{10}(9,9838 \times 10^{-1})$$

$$\log_{10}(0,99838) = \log_{10}(9,9838) + \log_{10}(10^{-1})$$

$$\log_{10}(0,99838) = 0,9993 + (-1)$$

$$\log_{10}(0,99838) = -0,0007$$

Maka:

$$10 \log_{10}(0,99838) = 10 \times (-0,0007)$$

$$= -0,007 \text{ dB}$$

Untuk $k = 2$:

$$10 \log_{10}(1,00266)$$

$$\log_{10}(1,00,266) = 0,001155$$

Maka:

$$10 \log_{10}(1,00266) = 10 \times 0,001155 = 0,01155 \text{ dB}$$

Untuk $k = 3$:

$$10 \log_{10}(0,99838) = -0,007 \text{ (sama dengan } k = 1)$$

4. Perhitungan Kuadrat Selisih

Untuk $k = 1$:

$$\left[10 \log_{10} \left(\frac{S_o(1)}{S_s(1)} \right) \right]^2 = [-0,007]^2 = 0,000049$$

Untuk $k = 2$:

$$\left[10 \log_{10} \left(\frac{S_o(2)}{S_s(2)} \right) \right]^2 = [0,01155]^2 = 0,0001334$$

Untuk $k = 3$:

$$\left[10 \log_{10} \left(\frac{S_o(3)}{S_s(3)} \right) \right]^2 = [-0,007]^2 = 0,000049$$

5. Penjumlahan dan Rata-rata

Jumlah kuadrat:

$$\sum_{k=1}^3 \left[10 \log_{10} \left(\frac{S_o(k)}{S_s(k)} \right) \right]^2 = 0,000049 + 0,0001334 + 0,000049$$

$$\sum_{k=1}^3 \left[10 \log_{10} \left(\frac{S_o(k)}{S_s(k)} \right) \right]^2 = 0,0002314$$

Rata-rata:

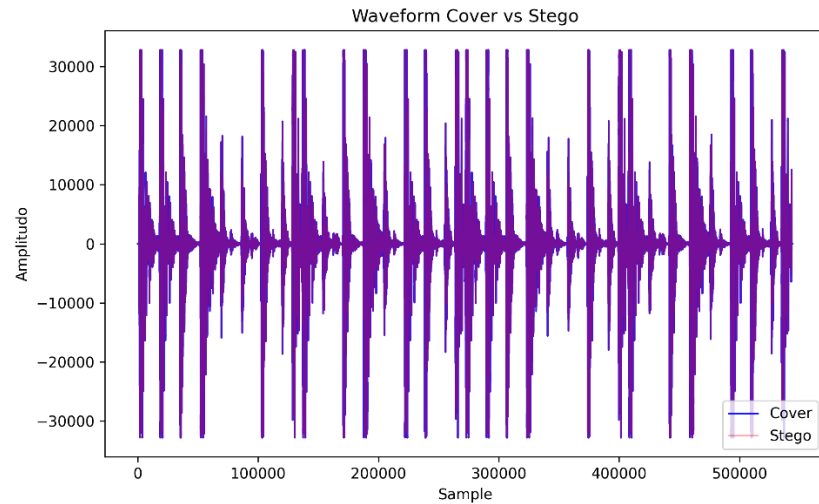
$$\frac{1}{K} \sum_{k=1}^3 \left[10 \log_{10} \left(\frac{S_o(k)}{S_s(k)} \right) \right]^2 = \frac{0,0002314}{3} = 0,00007713$$

6. Hasil Akhir *Spectral Distortion*

$$SD = \sqrt{\frac{1}{K} \sum_{k=1}^K \left[10 \log_{10} \left(\frac{S_o(k)}{S_s(k)} \right) \right]^2}$$

$$SD = \sqrt{0,00007713}$$

$$SD = 0,00878$$



Gambar 4.1 Grafik Penyisipan di tengah

Nilai tersebut menunjukkan bahwa perbedaan spektrum antara audio asli dan audio hasil penyisipan relatif sangat kecil, sehingga perubahan kualitas audio akibat proses penyisipan dapat dikatakan minimal. Hal ini juga terlihat pada Gambar 4.1, di mana pola spektrum frekuensi audio cover dan audio stego tampak hampir serupa. Secara umum, nilai *Spectral Distortion* (SD) berada pada rentang ≥ 0 , di mana nilai yang mendekati 0 (misalnya 0 – 1) menunjukkan perbedaan spektrum yang sangat kecil sehingga kualitas audio tergolong sangat baik. Nilai pada rentang sedang (sekitar 1 – 3) menunjukkan adanya sedikit perubahan namun masih dapat diterima, sedangkan nilai SD yang lebih besar (di atas 3) menunjukkan distorsi yang signifikan sehingga kualitas audio mengalami penurunan yang cukup jelas.

Selanjutnya, untuk mengetahui pengaruh lokasi penyisipan pesan terhadap kualitas audio, dilakukan pengujian dengan memvariasikan posisi penyisipan menggunakan metode *Spread Spectrum* dan 2-LSB. Hasil pengujian tersebut disajikan pada Tabel 4.5.

Tabel 4.5 Pengaruh Lokasi Penyisipan pada Nilai *Spectral Distortion*

No	Lokasi penyisipan	Nilai SD	Jumlah Sampel
1.	Awal Audio	2.1327 dB	84
2.	Tengah Audio	0.9975 dB	84
3.	Akhir audio	1.9660 dB	84

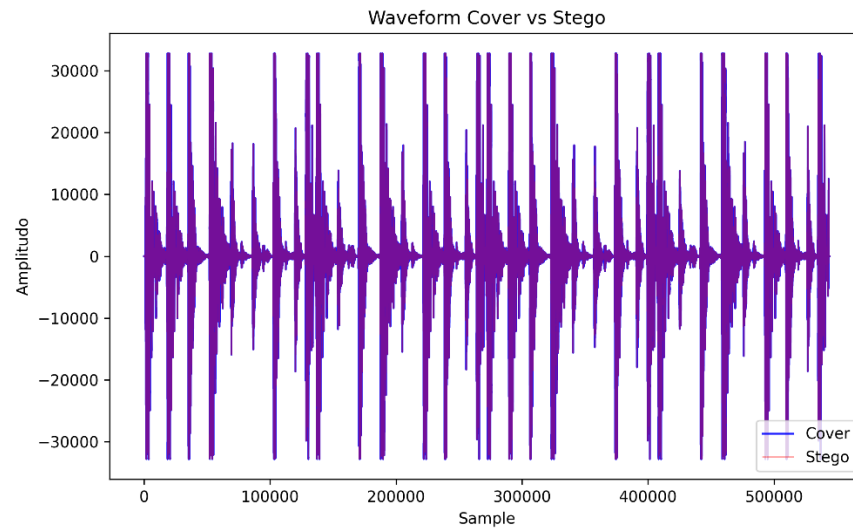
Berdasarkan Tabel 4.5 dapat dilihat bahwa nilai SD berbeda pada setiap lokasi penyisipan pesan. Nilai SD paling kecil diperoleh ketika pesan disisipkan pada bagian tengah audio, yaitu sebesar 0,9975 dB. Nilai tersebut menunjukkan bahwa perubahan spektrum frekuensi yang terjadi sangat kecil sehingga kualitas audio tergolong sangat baik.

penyisipan pesan pada bagian awal dan akhir audio menghasilkan nilai SD yang lebih tinggi dibandingkan bagian tengah audio, namun masih berada pada rentang yang dapat diterima. Hal ini menunjukkan bahwa penyisipan pada bagian tengah audio menghasilkan distorsi yang lebih kecil sehingga lebih optimal digunakan untuk proses penyisipan pesan. Visualisasi grafik hasil pengujian disajikan pada Lampiran 6.

Selain menganalisis pengaruh lokasi penyisipan, pengujian juga dilakukan untuk mengetahui pengaruh jumlah bit yang digunakan dalam metode N-LSB terhadap nilai SD. Pada pengujian ini digunakan pesan dengan ukuran yang cukup panjang dan proses penyisipan dilakukan pada seluruh bagian audio untuk melihat perubahan kualitas sinyal secara lebih menyeluruh. Nilai N-LSB kemudian divariasikan dari 2 hingga 5 bit pada setiap sampel audio guna mengamati sejauh mana peningkatan jumlah bit yang dimodifikasi mempengaruhi perubahan spektrum frekuensi audio. Hasil pengujian tersebut selanjutnya dihitung menggunakan metode *Spectral Distortion* dan disajikan pada tabel 4.6.

Tabel 4.6 Pengaruh Nilai N-LSB terhadap *Spectral Distortion*

No	Nilai N-LSB	Nilai SD	Jumlah Sampel
1.	2-LSB	3.6765 dB	2532
2.	3-LSB	4.0942 dB	1688
3.	4-LSB	4.4915 dB	1266
4.	5-LSB	4.6938 dB	1013

**Gambar 4.2** Grafik SD pada 2-LSB

Berdasarkan Tabel 4.6 dan Gambar 4.2, terlihat bahwa nilai Spectral Distortion cenderung meningkat seiring dengan bertambahnya nilai N-LSB yang digunakan dalam proses penyisipan. Pada penggunaan 2-LSB, nilai SD sebesar 3,6765 dB, kemudian meningkat menjadi 4,0942 dB pada 3-LSB, 4,4915 dB pada 4-LSB, dan mencapai 4,6938 dB pada 5-LSB. Adapun visualisasi hasil pengujian lengkap untuk variasi 3-LSB, 4-LSB, dan 5-LSB disajikan pada bagian lampiran 6.

Peningkatan nilai SD menunjukkan bahwa semakin banyak bit yang digunakan, semakin besar perubahan pada sinyal audio.. Akibatnya, tingkat distorsi pada spektrum frekuensi audio akan meningkat. Oleh karena itu, nilai N-LSB yang lebih kecil cenderung lebih baik dalam menjaga kualitas audio.

4.5 Integrasi Nilai-Nilai Islam dalam Keamanan Informasi

Integrasi nilai-nilai Islam dalam penelitian ini bertujuan untuk menunjukkan bahwa penerapan sistem keamanan informasi tidak hanya berorientasi pada aspek teknis, tetapi juga selaras dengan prinsip etika dan moral dalam Islam. Analisis integrasi dilakukan dengan mengaitkan nilai amanah yang bersumber dari Al-Qur'an dengan hasil penyisipan pesan, proses ekstraksi, serta evaluasi kualitas audio pada sistem steganografi audio yang diimplementasikan.

Konsep amanah dapat dimaknai sebagai tanggung jawab dalam menjaga kerahasiaan dan keutuhan informasi yang dipercayakan. Dalam konteks penelitian ini, pesan rahasia yang disisipkan ke dalam media audio dapat dipandang sebagai bentuk amanah digital yang perlu dilindungi dari akses pihak yang tidak berwenang. Berdasarkan hasil pengujian, pesan yang disisipkan menggunakan metode *Spread Spectrum* dan N-LSB dapat disembunyikan ke dalam sinyal audio dengan perubahan karakteristik yang relatif kecil. Hal ini mengindikasikan bahwa sistem memiliki kemampuan dalam menjaga kerahasiaan pesan sehingga tidak mudah dikenali secara langsung.

Selain kerahasiaan, keutuhan pesan juga menjadi bagian penting dari amanah. Hasil proses ekstraksi menunjukkan bahwa pesan yang diperoleh kembali sesuai dengan pesan asli yang disisipkan pada kondisi pengujian yang dilakukan. Temuan ini mengindikasikan bahwa sistem steganografi yang dibangun dapat menjaga informasi agar tidak mengalami perubahan yang signifikan selama proses penyimpanan maupun transmisi.

Hasil evaluasi menunjukkan bahwa perbedaan spektrum frekuensi antara audio asli dan audio stego relatif kecil, sehingga proses penyisipan pesan tidak

memberikan pengaruh signifikan terhadap kualitas audio. Dengan demikian, sistem steganografi audio yang dikembangkan mampu menjaga kualitas media sekaligus mempertahankan kerahasiaan dan keutuhan pesan sesuai dengan prinsip amanah.

Perlindungan informasi dari akses tidak sah serta keberhasilan pemulihan pesan menunjukkan bahwa sistem bekerja secara bertanggung jawab dalam mengelola informasi digital. Dengan demikian, penerapan metode *Spread Spectrum* dan N-LSB dalam penelitian ini tidak hanya efektif secara teknis, tetapi juga selaras dengan nilai-nilai Islam dalam menjaga kepercayaan, tanggung jawab, dan etika dalam keamanan informasi.

Selain itu, keamanan informasi dalam penelitian ini juga dapat ditinjau dari perspektif prinsip dasar dalam mengemban amanah, yaitu kejujuran, tanggung jawab, keadilan, profesionalitas, dan menjaga kerahasiaan. Kejujuran berkaitan dengan penyampaian informasi tanpa manipulasi, tanggung jawab tercermin dari kemampuan sistem dalam menjaga dan mengembalikan pesan, keadilan ditunjukkan melalui pembatasan akses hanya kepada pihak yang berhak, profesionalitas terlihat dari penerapan metode yang tepat, serta menjaga kerahasiaan menjadi aspek utama dalam proses penyembunyian pesan. Dengan demikian, sistem yang dibangun tidak hanya memenuhi aspek teknis, tetapi juga mendukung penerapan nilai amanah dalam pengelolaan informasi digital.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, sistem steganografi audio menggunakan kombinasi metode *Spread Spectrum* dan N-LSB berhasil diimplementasikan pada file audio berformat WAV untuk proses penyisipan dan ekstraksi pesan teks. Proses penyisipan dilakukan dengan menyebarkan bit pesan menggunakan metode *Spread Spectrum* sebelum disisipkan ke dalam sampel audio menggunakan metode N-LSB. Hasil pengujian menunjukkan bahwa nilai SD berbeda pada setiap variasi lokasi penyisipan dan nilai N-LSB yang digunakan. Pada pengujian lokasi penyisipan, nilai SD terkecil diperoleh pada bagian tengah audio sebesar 0,9975 dB. Sedangkan pada pengujian variasi N-LSB, nilai SD cenderung meningkat seiring bertambahnya jumlah bit yang digunakan dalam proses penyisipan. Berdasarkan hasil evaluasi kualitas audio menggunakan SD, proses penyisipan menghasilkan perubahan spektrum frekuensi yang relatif kecil sehingga kualitas audio stego tetap mendekati audio asli dalam batas pengujian yang dilakukan. Pada proses ekstraksi, pesan dapat diperoleh kembali dengan membaca *Header 72-bit* serta menggunakan parameter yang sama seperti pada proses penyisipan.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, berikut beberapa saran yang dapat dipertimbangkan untuk penelitian lanjutan maupun pengembangan sistem:

1. Penelitian selanjutnya disarankan untuk mengembangkan sistem steganografi audio agar dapat diterapkan pada berbagai jenis dan format audio selain audio mono, seperti audio stereo maupun format terkompresi (misalnya MP3), sehingga cakupan penerapan metode menjadi lebih luas.
2. Disarankan agar nilai N pada metode *Least Significant Bit* (LSB) dapat diubah atau diatur secara fleksibel, sehingga pengguna dapat menyesuaikan antara kapasitas penyisipan pesan dan kualitas audio stego yang dihasilkan.
3. Penelitian selanjutnya dapat mengembangkan pola penyisipan pesan pada sampel audio dengan menggunakan posisi penyisipan yang tidak dimulai dari titik awal (indeks 0) dan tidak bersifat statis. Penggunaan pola acak atau penyisipan secara berselang dapat membantu meningkatkan keamanan data serta mengurangi kemungkinan terdeteksinya pesan tersembunyi melalui analisis statistik.
4. Selain itu, sistem steganografi audio dapat dikembangkan dengan menambahkan mekanisme pengujian ketahanan (*robustness*) terhadap manipulasi sinyal audio, seperti kompresi, penambahan *noise*, atau perubahan amplitudo.
5. Penelitian selanjutnya dapat menguji penggunaan nilai *Spreading factor* selain 4 untuk mengetahui pengaruhnya terhadap tingkat keamanan, kapasitas penyisipan, serta kualitas audio hasil steganografi.
6. Penelitian selanjutnya juga dapat menambahkan parameter evaluasi kualitas audio yang lebih beragam, tidak hanya satu metrik, agar analisis kualitas audio stego menjadi lebih objektif dan komprehensif.

DAFTAR RUJUKAN

- Al 'Azhim Abadi, A. T. M. S. H. (2009). *Aunul ma'bud: Syarah sunan abu daud* (Jil. 3). (A. Taslim, Penerj.). Pustaka Azzam.
- Al-Albani, M. N. (1995). *Silsilah al-aḥādīth al-ṣaḥīḥah* (Vol. 10). Maktabah al-Ma'ārif.
- Alarood, A. A., Al-Dhaqm, A., Al-Hadhrami, T., & Alshahrani, M. (2022). Audio steganography method using least significant bit (LSB) encoding technique. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 13(1).
- Almara'beh, H. (2016). Steganography techniques: Data security using audio and video. *International Journal of Advanced Research in Computer Science and Software Engineering*, 6(2), 45–50.
- Alqahtani, A., Alzahrani, A., & Alghamdi, M. (2020). Digital audio steganography: A systematic review. *IEEE Access*, 8, 114966-114983.
- Alsolami, E. A. (2022). Audio steganography method using least significant bit (LSB) encoding technique. *Journal of Theoretical and Applied Information Technology*, 100(12).
- Assyahid, M. M., Rihartanto, R., & Utomo, D. S. B. (2018). Implementasi steganografi pesan teks ke dalam audio dengan metode *Spread Spectrum*. *Juristek*, 3(2), 27–34.
- Az-Zuhaili, W. (2011). *Tafsir al-Wajiz fi tafsir al-kitab al-'aziz*. Dār al-Fikr.
- Dixit, A., & Sharma, S. (2020). Audio steganography techniques for secure data communication: A review. *International Journal of Computer Applications*.
- Hamdi, A. A., Alghamdi, F., Alharbi, S., & Alotaibi, M. (2025). Improving audio steganography transmission over various wireless channels.
- Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T., & Jung, K. H. (2018). Image steganography in spatial domain: A survey. *The Visual Computer*, 34(1), 45-66.
- Ibn Katsir, I. (2000). *Tafsir al-Qur'an al-'Azhim* (Juz 2). Dār al-Kutub al-'Ilmiyyah.
- Kementerian Agama Republik Indonesia. (2022). *Al-Qur'an dan terjemahannya*. <https://quran.kemenag.go.id>
- Luthfi, A., & Jamarun, N. (2024). Evaluasi kinerja improved *Spread Spectrum* steganography pada advanced audio coding.
- Munir, R. (2004). *Steganografi dan watermarking*. Departemen Teknik Informatika Institut Teknologi Bandung.
- Munir, R. (2021). *Pengantar kriptografi* (pp. 1–57). <https://informatika.stei.itb.ac.id/~rinaldi.munir/>

- NU Online. (2025). *Tafsir Al-Wajiz: Surah Al-Mu'minin ayat 8*. <https://www.nu.or.id/>
- Nursobah, N., Lailiyah, S., & Kurnia, A. (2020). Implementasi steganografi pesan teks ke dalam file audio (.mp3) dengan algoritma advanced encryption standard dan least significant bit. *Jurnal IT*, 10(2), 122–139. <https://doi.org/10.37639/jti.v10i2.152>
- Rupanshi, Preeti, & Vandana. (2014). Audio steganography by direct sequence Spread Spectrum. *International Journal of Computer Trends and Technology*, 13(2), 83–86.

LAMPIRAN

Lampiran 1 Kode Program Mengubah Audio ke Integer

```
import soundfile as sf
import numpy as np
# === PARAMETER ===
file_path = "sunflower.wav"
JUMLAH_SAMPEL_TAMPIL = 48 # jumlah sampel yang digunakan
# === BACA FILE AUDIO ===
data, sample_rate = sf.read(file_path, dtype='int16')
# === CEK MONO ===
if data.ndim != 1:
    raise ValueError("File audio ini bukan mono! Harus 1
channel (mono).")
# === INFORMASI AUDIO ===
print("=== INFORMASI AUDIO ===")
print(f>Nama file      : {file_path}")
print(f>Sample rate   : {sample_rate} Hz")
print(f>Tipe data     : {data.dtype}")
print(f>Jumlah sampel : {len(data)}")
print(f>Rentang nilai : {np.min(data)} sampai
{np.max(data)}")
# =====
# SIMPAN SELURUH DATA AUDIO KE TXT (BENTUK TABEL)
# =====
output_all = "audio_integer_tabel.txt"
with open(output_all, "w") as f:
    f.write("Index\tSample_Value\n") # Header tabel
    for i, val in enumerate(data):
        f.write(f">{i}\t{val}\n")
print(f">\n Data audio integer (format tabel) disimpan
di: {output_all}")
# =====
# AMBIL SAMPEL MULAI DARI TITIK TENGAH
# =====
mid_index = len(data) // 2
start = mid_index
end = mid_index + JUMLAH_SAMPEL_TAMPIL
if end > len(data):
    raise ValueError("Jumlah sampel melebihi panjang data
audio!")
sampel_tengah = data[start:end]
print(f">\n=== SAMPEL YANG DIGUNAKAN UNTUK ANALISIS /
PENYISIPAN ===")
print(f">Indeks awal (titik tengah) : {start}")
print(f">Indeks akhir           : {end - 1}")
```

```
print(f"Jumlah sampel          :  
{len(sampel_tengah)}")  
print("\nIndex\tNilai Sample")  
for i, val in enumerate(sampel_tengah, start=start):  
    print(f"{i}\t{val}")
```

Lampiran 2 Kode Program mengubah Integer ke Audio

```
import soundfile as sf
import numpy as np
print("=== MODIFIKASI 48 SAMPEL AUDIO (SINKRON) ===")
# =====
# INPUT USER
# =====
input_txt = input("Masukkan file TXT tabel audio: ")
audio_referensi = input("Masukkan file audio referensi
(.wav): ")
output_wav = input("Masukkan nama file audio output
(.wav): ")
if not output_wav.lower().endswith(".wav"):
    output_wav += ".wav"
# =====
# AMBIL SAMPLE RATE OTOMATIS
# =====
_, sample_rate = sf.read(audio_referensi)
# =====
# BACA TXT FORMAT TABEL
# =====
audio_int = []
with open(input_txt, "r") as f:
    lines = f.readlines()[1:] # skip Header
    for line in lines:
        idx, val = line.strip().split()
        audio_int.append(int(val))
audio_int = np.array(audio_int, dtype=np.int16)
print("Jumlah total sample :", len(audio_int))
# =====
# INPUT NILAI BARU (48 SAMPLE)
# =====
nilai_baru_str = input(
    "\nMasukkan 48 nilai sample baru (pisahkan koma):\n"
)
nilai_baru = np.array(
    list(map(int, nilai_baru_str.split(","))),
    dtype=np.int16)
if len(nilai_baru) != 48:
    raise ValueError("HARUS tepat 48 sample!")
# =====
# TENTUKAN TITIK TENGAH (SAMA)
# =====
start_index = len(audio_int) // 2
```

```

end_index = start_index + 48
# =====
# LAKUKAN PERUBAHAN
# =====
audio_int[start_index:end_index] = nilai_baru
# =====
# SIMPAN AUDIO
# =====
sf.write(output_wav, audio_int, sample_rate)
# =====
# VALIDASI (BUKTI BERUBAH)
# =====
print("\n=== VALIDASI ===")
print(f"Index diubah : {start_index} s.d. {end_index -
1}")
print("Isi setelah perubahan:")
print(audio_int[start_index:end_index])
print("\nFile audio berhasil disimpan:", output_wav)

```

Lampiran 3 Kode Program Penyisipan Pesan

```
import wave
import numpy as np
print("=== PENYISIPAN PESAN AUDIO N-LSB (FINAL) ===")
audio_in = input("File audio cover (.wav): ")
audio_out = input("Nama audio stego (.wav): ")
print("\nJenis pesan:")
print("1 = ASCII (teks)")
print("2 = BIT (biner langsung)")
mode = input("Pilihan: ")
message = input("Masukkan pesan: ")
N = int(input("Nilai N-LSB (1-8): "))
if N < 1 or N > 8:
    print("✘ N harus 1-8")
    exit()
print("\nPosisi penyisipan:")
print("1 = Depan")
print("2 = Tengah")
print("3 = Belakang")
pos = int(input("Pilihan: "))
# =====
# INPUT PARAMETER LCG
# =====
seed = int(input("Seed (0-255): "))
a = int(input("Multiplier a (0-255): "))
c = int(input("Increment c (0-255): "))
m = int(input("Modulus m (0-65535): "))
# =====
# BACA AUDIO
# =====
audio = wave.open(audio_in, 'rb')
params = audio.getparams()
frames = audio.readframes(audio.getnframes())
audio.close()
samples = np.frombuffer(frames, dtype=np.int16).copy()
total_samples = len(samples)
print("\nTotal sampel:", total_samples)
# =====
# KONVERSI PESAN
# =====
if mode == "1":
    message_bits = ''.join(format(ord(c), '08b') for c
in message)
elif mode == "2":
    message_bits = ''.join(c for c in message if c in
"01")
```

```

        if len(message_bits) != len(message):
            print("✘ Pesan biner hanya boleh 0 dan 1")
            exit()
    else:
        print("✘ Mode tidak valid")
        exit()
    msg_len = len(message_bits)
    # =====
    # HEADER 72 BIT
    # =====
    # 16 bit panjang pesan
    Header_len = format(msg_len, '016b')
    # 8 bit posisi
    if pos == 1:
        pos_code = "00"
    elif pos == 2:
        pos_code = "01"
    elif pos == 3:
        pos_code = "10"
    Header_pos = pos_code + "000000"
    # 8 bit N-LSB
    Header_n = format(N, '08b')
    # 40 bit LCG
    Header_seed = format(seed, '08b')
    Header_a = format(a, '08b')
    Header_c = format(c, '08b')
    Header_m = format(m, '016b')
    Header_lcg = Header_seed + Header_a + Header_c + Header_m
    # Gabungkan semua
    Header_bits = Header_len + Header_pos + Header_n +
    Header_lcg
    # =====
    # HITUNG TOTAL
    # =====
    total_bits = 72 + msg_len
    needed_samples = (total_bits + N - 1) // N
    print("Total bit disisipkan:", total_bits)
    if needed_samples > total_samples:
        print("✘ Kapasitas audio tidak cukup")
        exit()
    # =====
    # PENYISIPAN
    # =====
    mask = ~(1 << N) - 1
    # =====
    # 1. SISIPKAN HEADER
    # =====

```

```

bit_index = 0
sample_index = 0
while bit_index < 72:
    bits =
Header_bits[bit_index:bit_index+N].ljust(N,'0')
    samples[sample_index] = (samples[sample_index] &
mask) | int(bits,2)
    bit_index += N
    sample_index += 1
Header_samples = sample_index
# =====
# 2. TENTUKAN POSISI PESAN
# =====
if pos == 1:
    sample_index = Header_samples
elif pos == 2:
    sample_index = total_samples // 2
elif pos == 3:
    sample_index = total_samples - (msg_len // N) - 1
# =====
# 3. SISIPKAN PESAN
# =====
bit_index = 0
while bit_index < msg_len:
    bits =
message_bits[bit_index:bit_index+N].ljust(N,'0')
    samples[sample_index] = (samples[sample_index] &
mask) | int(bits,2)
    bit_index += N
    sample_index += 1
# =====
# SIMPAN AUDIO
# =====
out = wave.open(audio_out, 'wb')
out.setparams(params)
out.writeframes(samples.tobytes())
out.close()
print("\n✅ Penyisipan berhasil")
print("Bit disisipkan:", total_bits)
print("Sampel digunakan:", needed_samples)

```

Lampiran 4 Kode Program Ekstraksi Pesan

```
import wave
import numpy as np
print("=== EKSTRAKSI PESAN AUDIO N-LSB (FINAL) ===")
audio_in = input("File audio stego (.wav): ")
N = int(input("Masukkan nilai N-LSB (harus sama saat penyisipan): "))
if N < 1 or N > 8:
    print("✘ N harus 1-8")
    exit()

# =====
# BACA AUDIO
# =====
audio = wave.open(audio_in, 'rb')
frames = audio.readframes(audio.getnframes())
audio.close()
samples = np.frombuffer(frames, dtype=np.int16)
total_samples = len(samples)
print("\nTotal sampel:", total_samples)
# =====
# FUNGSI AMBIL N-LSB
# =====
def get_n_lsb(value, n):
    return format(value & ((1 << n) - 1), f'0{n}b')
# =====
# 1. EKSTRAK HEADER (72 BIT)
# =====
Header_bits = ""
bit_index = 0
sample_index = 0
while bit_index < 72 and sample_index < total_samples:
    bits = get_n_lsb(samples[sample_index], N)
    Header_bits += bits
    bit_index += N
    sample_index += 1
Header_bits = Header_bits[:72]
if len(Header_bits) < 72:
    print("✘ Gagal membaca Header")
    exit()
# =====
# 2. PARSING HEADER
# =====
msg_len = int(Header_bits[0:16], 2)
pos_code = Header_bits[16:18]
if pos_code == "00":
```

```

    pos = 1
elif pos_code == "01":
    pos = 2
elif pos_code == "10":
    pos = 3
else:
    print("✘ Posisi tidak valid")
    exit()
# 6 bit padding diabaikan
# LCG PARAMETER
seed = int(Header_bits[32:40], 2)
a = int(Header_bits[40:48], 2)
c = int(Header_bits[48:56], 2)
m = int(Header_bits[56:72], 2)
print("\n=== HEADER ===")
print("Panjang pesan :", msg_len)
print("Posisi          :", pos)
print("N-LSB           :", N)
print("Seed            :", seed)
print("a               :", a)
print("c               :", c)
print("m               :", m)
# =====
# 3. HITUNG KEBUTUHAN
# =====
needed_msg_samples = (msg_len + N - 1) // N
Header_samples = (72 + N - 1) // N
# =====
# 4. TENTUKAN POSISI
# =====
if pos == 1:
    sample_index = Header_samples
elif pos == 2:
    start_index = total_samples // 2
    if start_index + needed_msg_samples > total_samples:
        print("✘ Data melebihi batas audio (tengah)")
        exit()
    sample_index = start_index
elif pos == 3:
    sample_index = total_samples - needed_msg_samples
# =====
# 5. EKSTRAK PESAN
# =====
message_bits = ""
bit_index = 0

```

```

while bit_index < msg_len and sample_index <
total_samples:
    bits = get_n_lsb(samples[sample_index], N)
    message_bits += bits
    bit_index += N
    sample_index += 1
message_bits = message_bits[:msg_len]
# =====
# 6. KONVERSI KE TEKS
# =====
def binary_to_text(binary_str):
    chars = []
    for i in range(0, len(binary_str), 8):
        byte = binary_str[i:i+8]
        if len(byte) == 8:
            chars.append(chr(int(byte, 2)))
    return ''.join(chars)
try:
    message_text = binary_to_text(message_bits)
except:
    message_text = "(tidak bisa dikonversi)"
# =====
# OUTPUT
# =====
print("\n===== HASIL EKSTRAKSI =====")
print("Biner pesan :", message_bits)
print("Pesan teks  :", message_text)

```

Lampiran 5 Kode Program Evaluasi

```
import wave
import numpy as np
import matplotlib.pyplot as plt
# =====
# INPUT FILE AUDIO
# =====
print("=== EVALUASI AUDIO STEGANOGRAFI ===")
cover_file = input("Masukkan file audio cover (.wav): ")
stego_file = input("Masukkan file audio stego (.wav): ")
# =====
# FUNGSI BACA AUDIO WAV
# =====
def read_audio(filename):
    with wave.open(filename, 'rb') as wf:
        frames = wf.readframes(wf.getnframes())
        audio = np.frombuffer(frames, dtype=np.int16)
        fs = wf.getframerate()
    return audio, fs
# =====
# FUNGSI FFT SPEKTRUM
# =====
def compute_spectrum(signal, fs):
    spectrum = np.abs(np.fft.fft(signal))
    freqs = np.fft.fftfreq(len(signal), d=1/fs)
    return freqs[:len(freqs)//2],
    spectrum[:len(spectrum)//2]
# =====
# FUNGSI SPECTRAL DISTORTION
# =====
def spectral_distortion(cover, stego):
    cover = cover.astype(float)
    stego = stego.astype(float)
    eps = 1e-10
    sd = np.sqrt(
        np.mean(
            (20 * np.log10((np.abs(stego) + eps) /
                (np.abs(cover) + eps))) ** 2
        )
    )
    return sd
# =====
# LOAD AUDIO
# =====
```

```

cover, fs = read_audio(cover_file)
stego, _ = read_audio(stego_file)
# Samakan panjang
min_len = min(len(cover), len(stego))
cover = cover[:min_len]
stego = stego[:min_len]
# Selisih sinyal
difference = stego - cover
# =====
# HITUNG SPEKTRUM
# =====
freq_c, spec_c = compute_spectrum(cover, fs)
freq_s, spec_s = compute_spectrum(stego, fs)
freq_d, spec_d = compute_spectrum(difference, fs)
# =====
# HITUNG SPECTRAL DISTORTION
# =====
sd_value = spectral_distortion(cover, stego)
# =====
# VISUALISASI (MERAH & BIRU)
# =====
# Waveform Cover vs Stego
plt.figure()
plt.plot(cover, color='blue', label='Cover', alpha=0.7)
plt.plot(stego, color='red', label='Stego', alpha=0.7)
plt.title("Waveform Cover vs Stego")
plt.xlabel("Sample")
plt.ylabel("Amplitudo")
plt.legend()
plt.show()
# Waveform Selisih
plt.figure()
plt.plot(difference, color='black', linestyle='--',
label='Difference')
plt.title("Waveform Selisih (Stego - Cover)")
plt.xlabel("Sample")
plt.ylabel("Amplitudo")
plt.legend()
plt.show()
# Spektrum Cover vs Stego
plt.figure()
plt.plot(freq_c, spec_c, color='blue', label='Cover',
alpha=0.7)
plt.plot(freq_s, spec_s, color='red', label='Stego',
alpha=0.7)

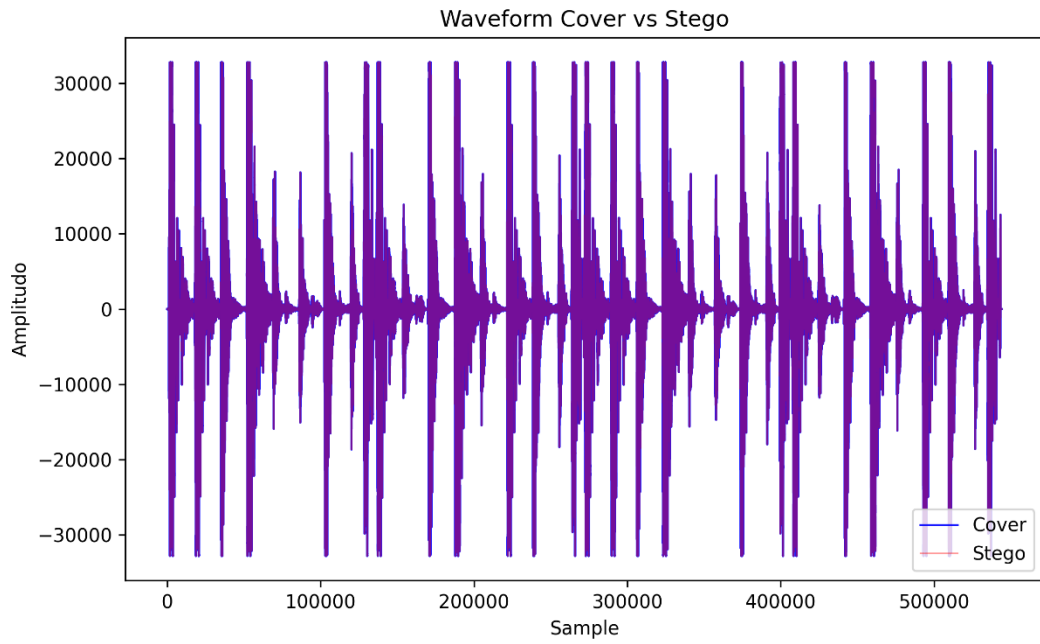
```

```

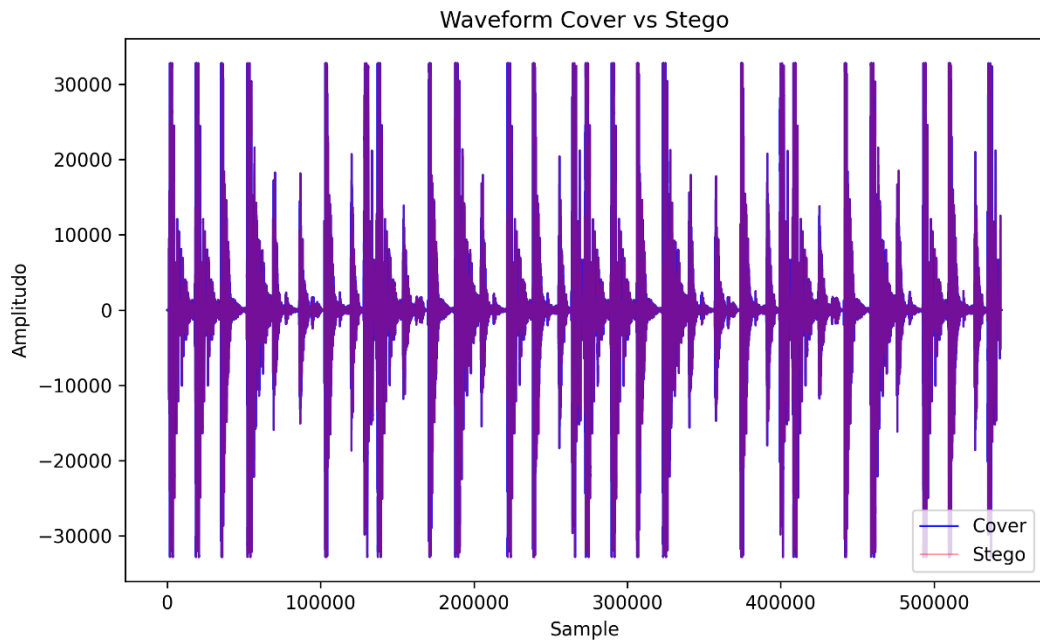
plt.title("Spektrum Frekuensi Cover vs Stego")
plt.xlabel("Frekuensi (Hz)")
plt.ylabel("Magnitudo")
plt.legend()
plt.show()
# Spektrum Selisih
plt.figure()
plt.plot(freq_d, spec_d, color='black', linestyle='--',
label='Difference')
plt.title("Spektrum Frekuensi Selisih")
plt.xlabel("Frekuensi (Hz)")
plt.ylabel("Magnitudo")
plt.legend()
plt.show()
# =====
# OUTPUT AKHIR
# =====
print("\n=== HASIL EVALUASI ===")
print(f"Spectral Distortion (SD): {sd_value:.4f} dB")

```

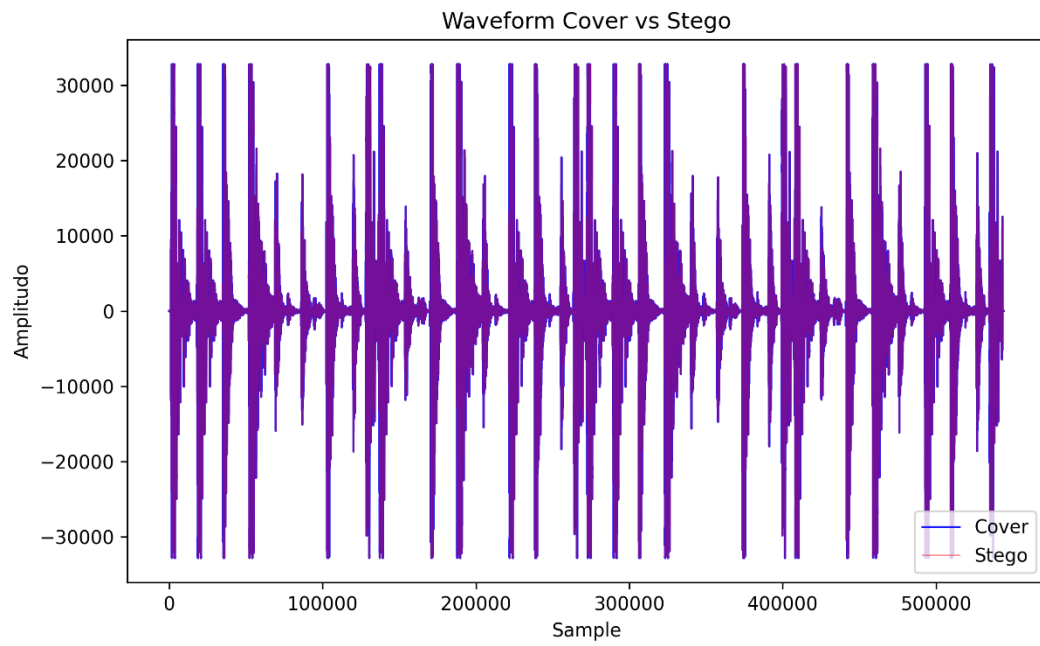
Lampiran 6 Gambar *Spectral Distortion*



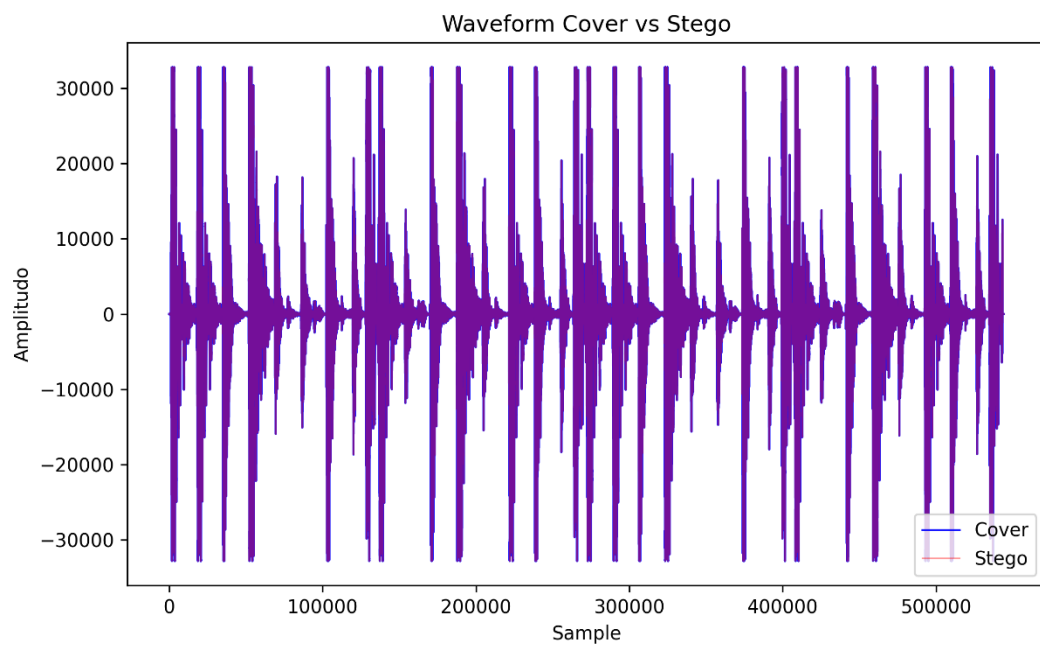
Grafik Penyisipan di awal



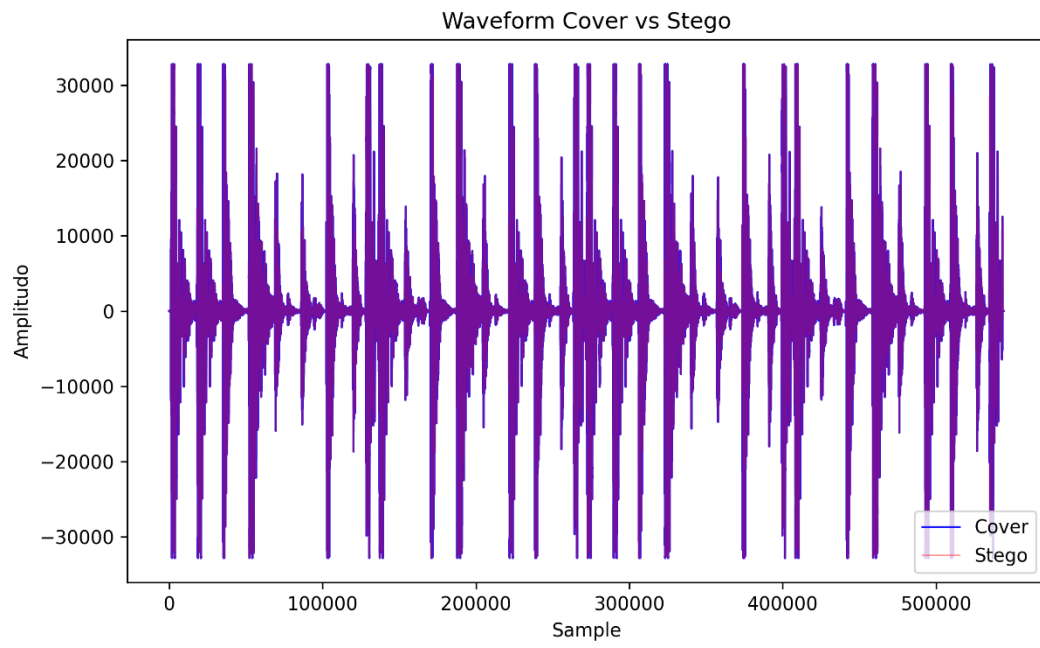
Grafik Penyisipan di akhir



Grafik SD pada 3-LSB



Grafik SD pada 4-LSB



Grafik SD pada 5-LSB

Lampiran 7 Program Penyisipan *Direct Sequence Spread Spectrum* (DSSS)

```
# =====
# 1. KUNCI → SEED (XOR)
# =====
def xor_key_to_seed(key, m):
    if key.isdigit():
        return int(key) % m
    seed = 0
    for ch in key:
        seed ^= ord(ch)
    return seed % m

# =====
# 2. TEXT → BINER 8 BIT
# =====
def text_to_binary(text):
    return ''.join(format(ord(c), '08b') for c in text)

# =====
# 3. SPREADING
# =====
def Spreading(binary, chip_0, chip_1):
    spreaded = ''
    for bit in binary:
        spreaded += chip_1 if bit == '1' else chip_0
    return spreaded

# =====
# 4. LCG → PN Sequence
# =====
def lcg_to_pn(seed, a, c, m, min_length):
    x = seed
    seq_decimal = []
    seq_bin = ''
    i = 1
    print("\n===== ITERASI LCG =====")
    while len(seq_bin) < min_length:
        prev_x = x
        x = (a * x + c) % m
        seq_decimal.append(x)
        bin8 = format(x, '08b')
        seq_bin += bin8
        # Tampilkan iterasi dan biner
    print(f"Iterasi ke-{i}:")
    print(f"X_{i} = ({a} × {prev_x} + {c}) mod {m}")
    print(f"          = ({a*prev_x} + {c}) mod {m} =
{a*prev_x + c} mod {m} = {x}")
    print(f"X_{i} dalam biner 8-bit: {bin8}\n")
    i += 1
```

```

        return seq_decimal, seq_bin[:min_length] # potong
sesuai panjang spreaded_msg
# =====
# 5. DSSS MODULATION (XOR)
# =====
def dsss_modulation(spreaded, pn_bits):
    return ''.join(
        '0' if spreaded[i] == pn_bits[i] else '1'
        for i in range(len(spreaded))
    )
# ===== PROGRAM UTAMA =====
# Input pengguna
message = input("Masukkan pesan          : ")
key      = input("Masukkan kunci LCG      : ")
chip_0   = input("Chip untuk bit 0 (misal 0000): ")
chip_1   = input("Chip untuk bit 1 (misal 1111): ")
a = int(input("Multiplier (a) : "))
c = int(input("Increment (c) : "))
m = int(input("Modulus (m) : "))
# 1. Pesan → biner 8-bit
binary_msg = text_to_binary(message)
# 2. Spreading
spreaded_msg = Spreading(binary_msg, chip_0, chip_1)
# 3. PN Sequence dari LCG
seed = xor_key_to_seed(key, m)
_, pn_bits = lcg_to_pn(seed, a, c, m, len(spreaded_msg))
# 4. Modulasi DSSS
dsss_signal = dsss_modulation(spreaded_msg, pn_bits)
# ===== OUTPUT =====
print("\n===== HASIL =====")
print("Pesan          :", message)
print("Biner pesan     :", binary_msg)
print("Hasil Spreading  :", spreaded_msg)
print("Rangkaian PN Sequence:", pn_bits)
print("Sinyal DSSS (XOR)  :", dsss_signal)

```

Lampiran 8 Program Ekstraksi *Direct Sequence Spread Spectrum* (DSSS)

```
# =====
# 0. XOR KEY → SEED (WAJIB TAMBAH INI)
# =====
def xor_key_to_seed(key, m):
    seed = 0
    for char in key:
        seed ^= ord(char)
    return seed % m

# =====
# 1. TEXT → BINER 8 BIT (untuk ekstraksi)
# =====
def binary_to_text(binary_str):
    chars = []
    for i in range(0, len(binary_str), 8):
        byte = binary_str[i:i+8]
        if len(byte) < 8:
            byte = byte.ljust(8, '0')
        chars.append(chr(int(byte, 2)))
    return ''.join(chars)

# =====
# 2. INVERSE SPREADING
# =====
def inverse_spreading(spreaded, chip_0, chip_1):
    chip_len = len(chip_0)
    bits = ''
    for i in range(0, len(spreaded), chip_len):
        chunk = spreaded[i:i+chip_len]
        if chunk == chip_0:
            bits += '0'
        elif chunk == chip_1:
            bits += '1'
        else:
            # Jika ada kombinasi lain, pilih logika
            terdekat (misal 1 jika lebih banyak 1)
            ones = chunk.count('1')
            zeros = chunk.count('0')
            bits += '1' if ones > zeros else '0'
    return bits

# =====
# 3. PN Sequence dari LCG (sama seperti embed)
# =====
def lcg_to_pn(seed, a, c, m, min_length):
    x = seed
    seq_bin = ''
```

```

    while len(seq_bin) < min_length:
        x = (a * x + c) % m
        seq_bin += format(x, '08b')
    return seq_bin[:min_length]
# =====
# 4. DSSS DEMODULATION
# =====
def dsss_demodulation(dsss_signal, pn_bits):
    return ''.join(
        '0' if dsss_signal[i] == pn_bits[i] else '1'
        for i in range(len(dsss_signal))
    )
# ===== PROGRAM EKSTRAKSI =====
dsss_signal = input("Masukkan sinyal DSSS (0/1) : ")
key = input("Masukkan kunci LCG : ")
chip_0 = input("Chip untuk bit 0 : ")
chip_1 = input("Chip untuk bit 1 : ")
a = int(input("Multiplier (a) : "))
c = int(input("Increment (c) : "))
m = int(input("Modulus (m) : "))
# Hitung seed
seed = xor_key_to_seed(key, m)
# Hasilkan PN Sequence sesuai panjang sinyal DSSS
pn_bits = lcg_to_pn(seed, a, c, m, len(dsss_signal))
# Demodulasi DSSS → spreaded
spreaded_signal = dsss_demodulation(dsss_signal, pn_bits)
# Inverse Spreading → dapat bit asli
original_bits = inverse_Spreading(spreaded_signal,
chip_0, chip_1)
# Konversi ke teks
original_text = binary_to_text(original_bits)
print("\n===== HASIL EKSTRAKSI =====")
print("Biner pesan hasil ekstraksi :", original_bits)
print("Pesan asli : ", original_text)

```

Lampiran 9 Data Audio

https://drive.google.com/drive/folders/1LE_iCk0dwSQVLcD2jNMs6JFe2nTITffU?usp=sharing

RIWAYAT HIDUP



Muhammad Agus Khifdon Yuwafi, lahir di Lamongan pada tanggal 06 Agustus 2004. Penulis berdomisili di Desa Wanar, Kecamatan Pucuk, Kabupaten Lamongan. Penulis merupakan anak kedua dan terakhir, putra dari pasangan Bapak Purwanto dan Ibu Ainul Fitriyah. Perjalanan pendidikan penulis di mulai di sekolah dasar di MI Salafiyah Wanar dan lulus pada tahun 2016. Pendidikan menengah pertama ditempuh di MTs Mambaul Ulum Wanar dan lulus pada tahun 2019. Penulis melanjutkan pendidikan ke jenjang menengah atas di SMA Ma'arif Nu Mambaul Ulum dan lulus pada tahun 2022. Pada jenjang ini, penulis aktif berorganisasi sebagai anggota OSIS, yang menjadi wadah pengembangan kepemimpinan dan tanggung jawab sosial. Selanjutnya penulis melanjutkan pendidikan Strata 1 di Universitas Islam Negeri Maulana Malik Ibrahim Malang dan memilih Program Studi Matematika Fakultas Sains dan Teknologi. Selama menempuh pendidikan di perguruan tinggi, penulis aktif berpartisipasi dalam kegiatan organisasi kemahasiswaan. Penulis bergabung sebagai anggota Himpunan Mahasiswa Program Studi (HMPS) “Integral Matematika” Divisi Penalaran Keagamaan selama dua periode, yaitu tahun 2023–2024. Pada periode kedua, penulis dipercaya untuk mengemban amanah sebagai Ketua Divisi Penalaran Keagamaan, yang berperan dalam mengoordinasikan program kerja, mengelola kegiatan keilmuan dan keagamaan, serta meningkatkan partisipasi anggota dalam kegiatan divisi. Selain itu, penulis juga aktif sebagai pengurus di Ma'had Universitas Islam Negeri (UIN) Malang dengan jabatan sebagai Musyrif, yang bertugas membimbing, mengawasi, serta mendampingi mahasiswa dalam kegiatan pembinaan keasramaan dan penguatan karakter religius.



BUKTI KONSULTASI SKRIPSI

Nama : Muhammad Agus Khifdon Yuwafi
NIM : 220601110012
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Penyisipan Pesan dalam File Audio dengan Metode Spread Spectrum dan N-LSB
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Dr. Ach. Nashichuddin, M.A

No	Tanggal	Hal	Tanda Tangan
1.	15 September 2025	Konsultasi Bab I, II, dan III	1.
2.	17 September 2025	Konsultasi Bab I, II, dan III	2.
3.	24 September 2025	Konsultasi Bab I, II, dan III	3.
4.	3 November 2025	Konsultasi Bab I, II, dan III	4.
5.	10 November 2025	Konsultasi Bab I, II, dan III	5.
6.	12 November 2025	Konsultasi Kajian Agama Bab I-II & Konsultasi Bab I-II-III	6.
7.	13 November 2025	Konsultasi Kajian Agama Bab I dan II	7.
8.	18 November 2025	ACC Kajian Agama Bab I dan II	8.
9.	24 November 2025	ACC Bab I, II, dan III	9.
10.	25 November 2025	ACC Seminar Proposal	10.
11.	19 Januari 2026	Konsultasi Revisi Seminar Proposal	11.
12.	4 Februari 2026	Konsultasi Bab IV dan V	12.
13.	23 Februari 2026	Konsultasi Bab IV dan V	13.
14.	24 Februari 2026	Konsultasi Kajian Agama Bab IV	14.



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

15.	2 Maret 2026	Konsultasi Bab IV dan V	15. <i>[Signature]</i>
16.	11 Maret 2026	ACC Bab IV dan V	16. <i>[Signature]</i>
17.	12 Maret 2026	ACC Seminar Hasil	17. <i>[Signature]</i>
18.	14 April 2026	Konsultasi Revisi Seminar Hasil	18. <i>[Signature]</i>
19.	22 April 2026	Konsultasi Revisi Seminar Hasil	19. <i>[Signature]</i>
20.	27 April 2026	ACC Sidang Skripsi	20. <i>[Signature]</i>
21.	7 Mei 2026	ACC Keseluruhan	21. <i>[Signature]</i>



Malang, 7 Mei 2026,
Ketua Program Studi Matematika

[Signature]
D. Fachrur Rozi, M.Si.
NIP. 19800527 200801 1 012