

Implementasi Metode Advanced LSB dan AES pada Aplikasi Android untuk Menyisipkan Pesan pada Berkas Audio MP3

Jova Andres Riski Sirait
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jalan Ganesha 10, Bandung, Indonesia
jovaandrea8721@gmail.com

Abstrak—Dalam era digital yang berkembang dengan cepat, keamanan informasi menjadi aspek penting yang harus diperhatikan. Salah satu teknik yang sering digunakan untuk melindungi pesan adalah steganografi. *Advanced Least Significant Bit* (ALSB) sebagai salah satu metode steganografi memiliki keunggulan dari sisi kapasitas pesan yang dapat disisipkan dan lebih sulit untuk diekstrak daripada metode standarnya yaitu *Least Significant Bit* (LSB). Audio MP3 dapat digunakan sebagai media untuk menyisipkan pesan sebagai format audio yang paling populer dan luas penggunaannya. Penyisipan dilakukan dengan melewati bagian *header* dan *side information* pada setiap *frame* MP3 untuk menghindari *audio corrupt*. Sebagai upaya mengurangi *noise* yang muncul akibat penyisipan pesan, volume dari *frame* yang mengandung pesan diatur menjadi 0 yang menyebabkan efek *silent*. Untuk meningkatkan keamanan pesan, kriptografi dengan algoritma AES digunakan untuk mengenkripsi pesan, sehingga pesan yang berhasil diekstrak tidak dapat dibaca sebagai bentuk aslinya tanpa adanya kunci untuk dekripsi. Penyisipan pesan dengan metode tersebut dapat diimplementasikan menjadi sebuah aplikasi Android sebagai perangkat seluler yang paling banyak digunakan khususnya sebagai media pertukaran pesan. Kualitas audio diukur dengan metrik *Peak Signal to Noise Ratio* (PSNR), *Signal to Noise Ratio* (SNR), dan *Bit Error Rate* (BER). Nilai PSNR, SNR, dan BER masing-masing 48dB, 43dB, dan 0.2% menghasilkan efek *silent* dengan kemunculan yang tidak terlalu sering dengan kualitas yang tergolong baik, yaitu lebih dari 30dB. Selain itu, keberhasilan ekstraksi pesan juga ditentukan oleh masukan kunci yang benar sehingga pesan menjadi lebih aman.

Kata kunci— *Advanced Least Significant Bit*, AES, steganografi audio, audio MP3

I. PENDAHULUAN

Dengan pesatnya perkembangan teknologi komputer dan internet, semakin banyak data yang dihasilkan oleh setiap individu, dan segala jenis informasi multimedia tersebar di internet dalam berbagai situasi [1]. Kriptografi dan steganografi adalah teknik yang terkenal dan banyak digunakan untuk memanipulasi informasi untuk menyandikan dan menyembunyikan keberadaannya [2]. Kriptografi adalah seni mengubah teks yang dapat dibaca (teks biasa) menjadi teks yang tidak dapat dibaca (*chiphertext*) untuk menjamin privasi data

[13]. Salah satu algoritma kriptografi yang terkenal adalah *Advanced Encryption Standard* (AES) yang termasuk kriptografi simetris. Kriptografi simetris merupakan algoritma kriptografi yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi [12]. Steganografi merupakan teknik untuk menyisipkan pesan ke dalam suatu media yang tidak mencurigakan sehingga dapat dikirimkan ke penerima tanpa adanya pihak yang sadar akan keberadaan pesan tersebut [15]. Salah satu media yang dapat digunakan adalah berkas audio karena adanya redundansi yang membuat audio menjadi media yang baik untuk steganografi [3]. Beberapa metode yang biasa digunakan untuk steganografi adalah *Least Significant Bit* (LSB), *Phase Coding*, *Echo Hiding*, dan *Spread Spectrum* [14]. Sampai saat ini, audio dengan format MP3 menjadi salah satu format paling populer dan dijadikan preferensi dibandingkan format lain [4]. Kebanyakan penelitian saat ini masih menggunakan format WAV pada berbagai metode frekuensi dan *wavelet*, namun metode tersebut memerlukan komputasi yang cukup mahal dan rentan kehilangan informasi pada saat ekstraksi [5]. Pada pengukuran kualitas audio steganografi secara objektif, kriteria yang akan diuji difokuskan pada aspek *imperceptibility*, *fidelity*, *recovery*, dan *capacity* [16]. Beberapa penelitian sebelumnya telah berhasil memberikan hasil yang positif, namun belum melakukan simulasi untuk pertukaran pesan sehari-hari. Salah satu upaya yang dapat dilakukan adalah dengan mengimplementasikan skema penyisipan pesan menjadi sebuah aplikasi Android yang mudah digunakan. Android dipilih sebagai *platform* pengujian karena penggunaannya yang paling luas dari antara perangkat *mobile* lainnya [6]. Aplikasi yang dibangun pada penelitian ini diharapkan dapat memberikan alternatif pertukaran pesan dengan keamanan yang tinggi dan penggunaan yang praktis.

II. PENELITIAN TERKAIT

A. *Advanced Least Significant Bit* (ALSB) Modification

Metode LSB standar memiliki kelemahan yaitu ekstraksi pesan yang cukup mudah karena penyisipan dilakukan secara berurutan pada bit terakhir. Selain itu, kapasitas pesan yang dapat disisipkan juga terbatas. Penelitian [11] mengembangkan metode ALSB untuk meningkatkan kapasitas pesan dan

keamanannya dibandingkan metode LSB standar. ALSB menyisipkan pesan pada posisi acak dari salah satu 100 byte pertama pada audio. Pada posisi acak tempat penyisipan pesan, nilai dari LSB dan MSB akan dicek terlebih dahulu. Jika nilai LSB dan MSB adalah sama, maka 4 bit dari pesan akan disisipkan pada bit kedua sampai bit kelima audio. Sebaliknya jika nilai LSB dan MSB berbeda, maka hanya 2 bit dari pesan yang akan disisipkan pada bit kedua dan ketiga audio. Pengacakan posisi penyisipan dan variasi nilai LSB dan MSB menyebabkan pesan menjadi lebih aman dan sulit terdeteksi. Berikut adalah algoritma penyisipan yang dikembangkan pada penelitian [11]:

Algoritma 1: Advanced Least Significant Bit (ALSB)

// A adalah audio awal, P adalah pesan, A' adalah audio
// hasil steganografi

Input A, P

Output A'

```
// duplikasi berkas audio
1: for i = 1 to sizeof (H) do
2:   A'[i] = A[i]

// pilih posisi awal penyisipan dari 100-byte paling awal
// secara acak
3: Initial_position = random (1 to 100)

// metode ALSB
4: for i = Initial_position to sizeof (A') do
5:   {baca nilai LSB dan MSB A'[i]}
6:   if LSB+MSB = 00 or 11 then
7:     {sisipkan 4 bit P pada bit kedua sampai bit kelima
8:     A'[i]}
9:   else
10:    {sisipkan 2 bit P pada bit kedua dan ketiga A'[i]}
11: end if
```

Pada pengujian yang telah dilakukan pada [11], terlihat bahwa metode ALSB berhasil meningkatkan kapasitas pesan dibandingkan dengan metode LSB standar serta meningkatkan keamanan dengan mempersulit proses ekstraksi.

B. Audio MP3

Salah satu format audio yang paling populer adalah MP3 yang di-encode dengan MPEG-1 atau MPEG-2 Layer 3. Desain tersebut dibuat oleh The Moving Picture Experts Group pada tahun 1991 dan merupakan tipe *lossy compression*. MP3 merupakan gabungan dari banyak *frame* yang tiap *frame*-nya terdiri dari sekitar 26 milidetik audio. Berkas MP3 biasanya diawali dan diakhiri dengan *information tag*, yaitu ID3v2 Tag pada awal berkas dan ID3v1 Tag pada akhir berkas. Setiap *frame*-nya terdiri atas *header* sebanyak 4-byte, CRC sebanyak 2-byte, *side information* sebanyak 17-byte pada audio mono dan 32-byte pada audio stereo dan *main data* dengan panjang yang bervariasi. Struktur berkas MP3 dapat dilihat pada Gambar II.1.

Bagian *frame header* terdiri dari cukup banyak komponen, yaitu *Frame Sync*, *MPEG Version*, *Bit Rate*, *Channel Mode*, *Copyright*, dan lain-lain. Bagian *side information* berisi banyak informasi yang akan digunakan player untuk mengolah data MP3 menjadi sinyal suara yang dapat didengar. Bagian *main data* berisi data terkompresi dari audio.



Gambar II.1 Struktur berkas MP3 [17]

C. Steganografi Audio dengan Metode Least Significant Bit (LSB) pada berkas MP3 dan WAV

Implementasi metode LSB untuk steganografi audio MP3 dan WAV dan menjadi sebuah aplikasi Android telah diteliti sebelumnya pada penelitian [7]. Metode LSB dipilih karena mudah untuk diimplementasikan, tidak membutuhkan perhitungan yang kompleks, dan tidak membutuhkan sumber daya yang banyak. Penelitian ini menggunakan dua format audio yang berbeda, yaitu MP3 dan WAV. Pengukuran kualitas audio yang dihasilkan pada skema ini menggunakan PSNR. Jika nilai PSNR kurang dari 30dB, kualitas audio hasil steganografi dapat dikatakan buruk [7]. Berdasarkan hasil pengujian pada penelitian [7], terjadi perbedaan kualitas yang cukup jauh antara steganografi pada audio MP3 dan audio WAV. Audio WAV memberikan hasil PSNR yang jauh lebih baik sehingga lebih cocok digunakan pada skema ini. Sehingga, metode steganografi lain diperlukan untuk peningkatan kualitas audio khususnya pada media berkas MP3.

D. Penyembunyian Pesan Teks pada Audio MP3 dengan Modifikasi Metode Least Significant Bit (LSB)

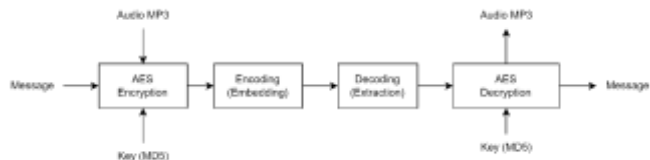
Pada algoritma LSB biasa, untuk mendapatkan pesan pada berkas audio dapat dilakukan dengan mengambil setiap bit terakhir secara linear. Hal ini tentunya memudahkan ekstraksi pesan jika keberadaannya telah diketahui. Penelitian [8] melakukan modifikasi algoritma LSB untuk meningkatkan keamanan pesan dari pihak luar. Algoritma ini menyisipkan bit pesan pada posisi tertentu berdasarkan nilai MSB yang akan dikonversi menjadi nilai desimal. Dua desimal pertama kemudian dipilih dan dijadikan posisi untuk menyisipkan bit pesan.

Kualitas audio hasil steganografi pada penelitian [8] diukur dengan *Signal to Noise Ratio* (SNR) dan *Peak Signal to Noise Ratio* (PSNR) dan pengujian dilakukan pada berkas audio MP3 dengan ukuran yang berbeda. Dari hasil pengujian yang dilakukan pada penelitian [8], algoritma yang dikembangkan menghasilkan peningkatan kualitas audio dibandingkan metode LSB biasa. Keamanan pesan juga meningkat akibat dari posisi

penyisipan yang acak sehingga lebih sulit untuk dideteksi dan diekstrak. Akan tetapi, peningkatan kualitas yang dihasilkan kurang signifikan.

E. Implementasi Enkripsi AES dan Fungsi Hash MD5 pada Steganografi Audio MP3

Steganografi pada audio dapat dideteksi dengan melakukan komparasi antara audio yang dicurigai terdapat pesan dengan audio aslinya. Oleh karena itu, AES dibutuhkan sebagai metode enkripsi pesan agar pesan yang terdeteksi tidak dapat dibaca informasinya dengan jelas [9]. Kunci AES dapat dibangkitkan dengan menggunakan fungsi *hash* MD5 untuk menghasilkan kunci yang acak.



Gambar II.2 Skema steganografi dan enkripsi AES

Terdapat 4 proses utama yaitu enkripsi, encoding, decoding, dan dekripsi. Proses enkripsi dimulai dengan membangkitkan kunci enkripsi menggunakan *hash* MD5 dan kunci tersebut digunakan untuk enkripsi pesan. Pada proses encoding, audio MP3 diproses untuk menentukan bagian yang akan disisipkan pesan. *Header* dan *side information* akan diabaikan karena merupakan bagian penting dari berkas MP3. Bagian yang akan disisipkan pesan adalah *frame* yang memiliki nilai yang homogen pada data audionya. Bagian tersebut biasanya tidak menghasilkan suara yang dapat terdengar. Pada saat penyisipan, kode tertentu ikut disisipkan agar dapat dikenali pada saat ekstraksi dan proses ini akan menghasilkan audio yang telah mengandung pesan. Proses selanjutnya adalah decoding, yaitu proses ekstraksi pesan dari sebuah audio MP3 yang telah disisipkan pesan sebelumnya. Proses ini melakukan pengecekan dari kode tertentu untuk menentukan apakah ada pesan di dalamnya. Jika kode terdeteksi, mak pesan akan diambil dari setiap *frame* yang mengandung kode tersebut. Hasil dari proses tersebut adalah pesan yang terenkripsi, yang selanjutnya akan menjadi masukan pada proses dekripsi untuk menghasilkan pesan yang dapat dibaca. Skema penyisipan pesan pada penelitian [9] dapat dilihat pada Gambar II.2.

Berdasarkan hasil pengujian pada penelitian [9], AES berhasil diimplementasikan sebagai lapisan keamanan tambahan pada steganografi audio dan pesan yang disisipkan dengan berbagai jenis format berhasil diekstrak tanpa adanya perubahan pada pesan.

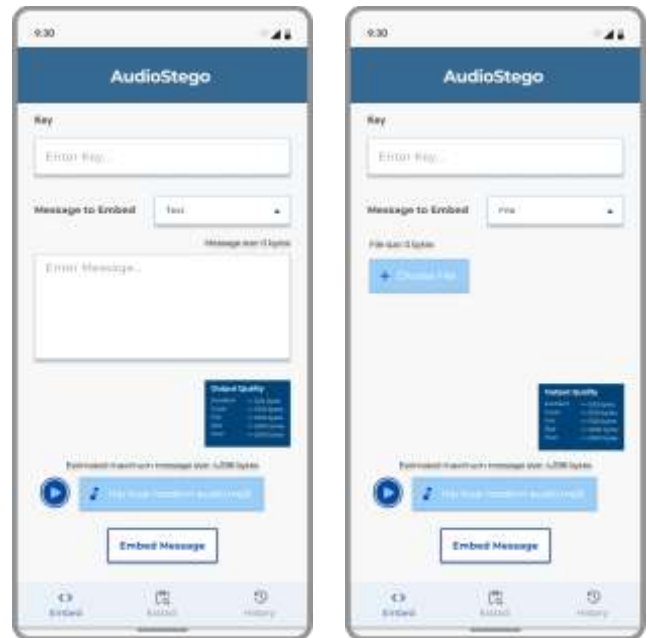
III. RANCANGAN SOLUSI

Skema penyisipan pesan melalui berkas audio mencakup dua bagian utama, yaitu bagian kriptografi dan bagian steganografi. Algoritma kriptografi yang akan digunakan adalah *Advanced Encryption Standard* (AES) karena menghasilkan keamanan yang lebih tinggi dari algoritma enkripsi lain [3] serta masih menjadi algoritma enkripsi yang paling efisien [10]. Algoritma steganografi yang akan diimplementasikan adalah *Advanced*

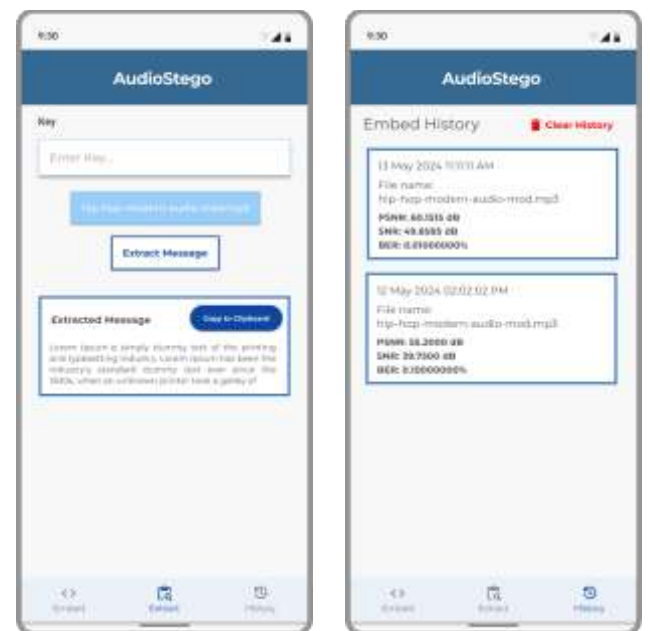
Least Significant Bit (ALSB) yang dikembangkan pada penelitian [11]. Skema penyisipan pesan tersebut kemudian akan diimplementasikan menjadi sebuah aplikasi yang berjalan pada perangkat Android.

A. Rancangan Antarmuka

Terdapat tiga halaman utama yang akan dibangun, yaitu halaman penyisipan, halaman ekstraksi, dan halaman riwayat penyisipan pesan. Rancangan antarmuka aplikasi yang akan dibangun dapat dilihat pada Gambar III.1 dan Gambar III.2.



Gambar III.1 Rancangan antarmuka halaman penyisipan



Gambar III.2 Rancangan antarmuka halaman ekstraksi dan halaman riwayat

Aplikasi terdiri dari tiga komponen utama pada setiap halamannya, yaitu TopBar, Page, dan BottomBar. TopBar terletak pada bagian paling atas dari tampilan aplikasi yang menunjukkan nama dari aplikasi. BottomBar terletak pada bagian paling bawah yang terdiri dari tiga navigasi yang akan mengarahkan pengguna ke tiga Page yang berbeda, yaitu EmbedPage, ExtractPage, dan HistoryPage.

B. Rancangan Alur Kerja Program

Terdapat dua fungsionalitas utama yang akan dibangun, yaitu fungsi untuk melakukan penyisipan pesan dan fungsi untuk ekstraksi pesan. Rancangan alur aplikasi android dapat dilihat pada Gambar III.3 dan Gambar III. Berikut adalah alur kerja aplikasi pada proses penyisipan:

- 1) Aplikasi menerima masukan berupa audio dengan format MP3, pesan berupa teks, gambar, audio, ataupun jenis berkas lainnya, serta kunci steganografi.
- 2) Audio akan disiapkan untuk proses penyisipan melalui dekomposisi sehingga audio akan terpisah menjadi kumpulan *frame* MP3 yang terdiri atas *header*, *side information*, dan *audio data*. Ukuran pesan maksimal yang dapat disisipkan akan dihitung untuk memperkirakan estimasi kualitas yang akan dihasilkan. Formula untuk menghitung ukuran pesan maksimal adalah sebagai berikut:

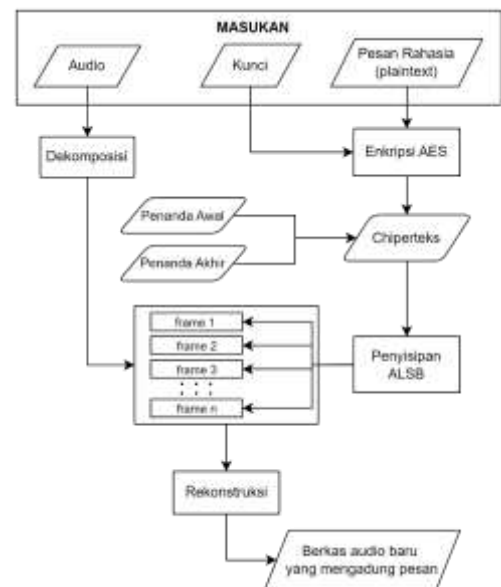
$$t = \sum_{i=1}^n \begin{cases} 4, MSB(b_i) = LSB(b_i) \\ 2, MSB(b_i) \neq LSB(b_i) \end{cases} \quad (1)$$

dengan t adalah ukuran pesan maksimal yang dapat disisipkan, n adalah total byte yang dapat disisipkan, MSB dan LSB masing-masing adalah nilai bit pertama dan bit terakhir, serta b_i adalah nilai byte ke- i .

- 3) Pesan yang dimasukkan akan dienkripsi dengan kunci steganografi yang diberikan dan ditandai bagian awal dan akhirnya menggunakan rangkaian karakter tertentu. Pada penanda bagian awal pesan, akan ditambahkan informasi jenis pesan yang disisipkan agar memudahkan identifikasi pada proses ekstraksi pesan.
- 4) Pesan yang telah ditambahkan penanda awal dan akhir pesan kemudian disisipkan dengan metode *Advanced Least Significant Bit* (ALSB) ke dalam *audio data* pada tiap *frame* MP3. Penyisipan ke dalam *frame* MP3 dilakukan dengan mengabaikan total bytes dari *header* dan *side information* untuk memastikan pesan tidak menggantikan informasi penting di *header* dan *side information* yang dapat menyebabkan audio menjadi rusak.
- 5) *Frame* MP3 yang mengandung pesan akan menghasilkan *noise* dengan nada yang cukup tinggi sehingga cukup mengganggu ketika didengarkan. Untuk mengatasi hal tersebut, bagian dari *side information* yang mengatur volume dari *frame* yang

mengandung pesan diatur menjadi 0 untuk menghilangkan *noise* secara keseluruhan. Efek yang dihasilkan adalah adanya beberapa bagian kecil dari audio yang tidak menghasilkan suara (*silent*) selama beberapa milidetik dengan kemunculan sebanding dengan besar dari pesan yang disisipkan.

- 6) *Frame* MP3 yang telah disisipkan pesan kemudian dikonstruksi kembali menjadi bentuk audio MP3 yang dapat dijalankan pada pemutar audio dan disimpan sebagai sebuah berkas audio baru.
- 7) *Frame* MP3 yang telah modifikasi kemudian akan dibandingkan dengan *frame* MP3 orisinal untuk membandingkan kualitas audionya melalui pengukuran *Peak Signal to Noise Ratio* (PSNR), *Signal to Noise Ratio* (SNR), dan juga *Bit Error Rate* (BER)

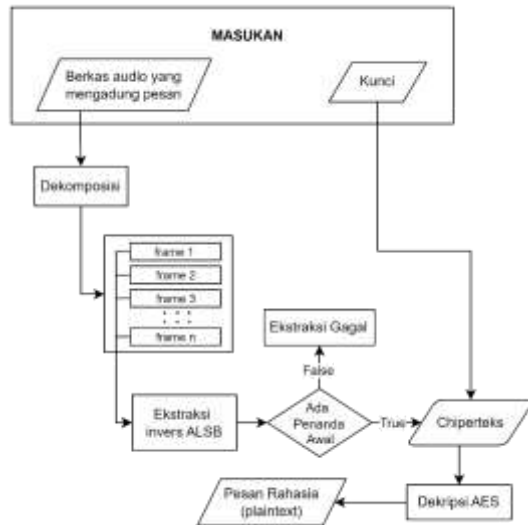


Gambar III.3 Alur penyisipan pesan pada aplikasi

Berikut adalah alur kerja aplikasi pada proses ekstraksi:

- 1) Aplikasi menerima masukan berupa audio dengan format MP3 yang telah disisipkan pesan serta kunci steganografi.
- 2) Audio akan disiapkan untuk proses penyisipan melalui dekomposisi sehingga audio akan terpisah menjadi kumpulan *frame* MP3 yang terdiri atas *header*, *side information*, dan *audio data*.
- 3) Setiap *frame* MP3 kemudian akan dilakukan ekstraksi bit-bit pesan dengan melakukan kebalikan dari metode metode *Advanced Least Significant Bit* (ALSB).
- 4) Bit-bit pesan tersebut kemudian akan dibentuk menjadi pesan berupa *byte array* dan akan berhenti apabila pesan telah mengandung penanda akhir pesan.

- 5) Keseluruhan pesan yang telah diekstrak kemudian dipisahkan dari penanda awal dan penanda akhir sehingga hanya menyisakan pesan aslinya. Penanda awal kemudian dicek informasi jenis pesan tersebut.
- 6) Pesan yang tersisa kemudian akan didekripsi menggunakan kunci yang dimasukkan. Apabila jenis pesan berupa teks, pesan akan ditampilkan. Apabila jenis pesan berupa berkas, pesan akan disimpan sebagai berkas baru.



Gambar III.4 Alur ekstraksi pesan pada aplikasi

IV. IMPLEMENTASI

Implementasi dilakukan dengan menggunakan perangkat lunak Android Studio Koala 2024 dan aplikasi dijalankan pada Emulator Android versi 14. Terdapat beberapa pustaka yang digunakan untuk mendukung pembuatan dari aplikasi Android yang dibuat.

Pustaka `net.addradio.codec:mpeg-audio-streams` merupakan pustaka yang digunakan untuk melakukan dekomposisi audio dengan tipe MPEG menjadi kumpulan frame serta dapat melakukan konstruksi kembali frame-frame tersebut menjadi audio yang dapat didengarkan.

Pustaka `com.arthenica:ffmpeg-kit-full` merupakan pustaka yang digunakan untuk melakukan proses konversi format audio. Karena proses penyisipan dan ekstraksi dirancang untuk masukan audio dengan format MP3, pustaka ini dimanfaatkan untuk menambah jenis format audio yang dapat diterima aplikasi dengan mengonversinya menjadi format MP3.

Pustaka `android.room` merupakan pustaka yang digunakan untuk menyimpan informasi metrik audio dari hasil penyisipan sehingga pengguna dapat melihat riwayat hasil penyisipan yang telah dilakukan.

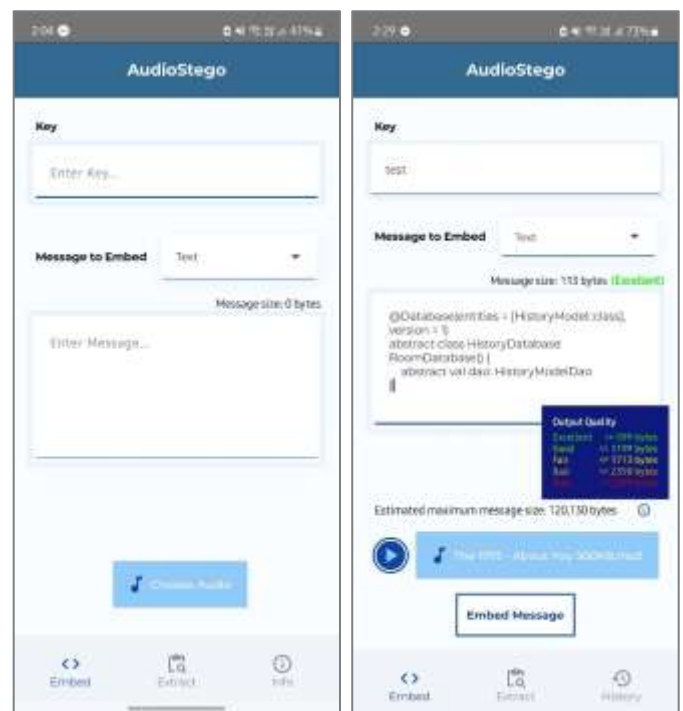
Pustaka `androidx.compose`, `androidx.navigation`, dan `com.google.accompanist:accompanist-permissions` merupakan pustaka utilitas yang digunakan dalam mengembangkan aplikasi pada sistem operasi Android.

Fungsinya mencakup konstruksi tampilan aplikasi, pengaturan perpindahan antar halaman pada aplikasi, serta pengaturan akses aplikasi seperti misalnya meminta izin untuk mengakses berkas foto, audio, dan lain-lain.

A. Implementasi Antarmuka

Dari rancangan antarmuka yang telah dibuat, antarmuka aplikasi diimplementasikan menjadi tiga halaman utama, yaitu *EmbedPage*, *ExtractPage*, dan *HistoryPage* yang semuanya dituliskan pada modul *MainActivity*. Hasil implementasi tersebut dapat dilihat pada Gambar IV.1, Gambar IV.2, Gambar IV.3, dan Gambar IV.4.

Setelah audio dipilih dan proses dekomposisi selesai, akan muncul sebuah *pop-up* yang menunjukkan estimasi kualitas pesan yang dihasilkan pada beberapa rentang ukuran pesan yang ingin disisipkan.

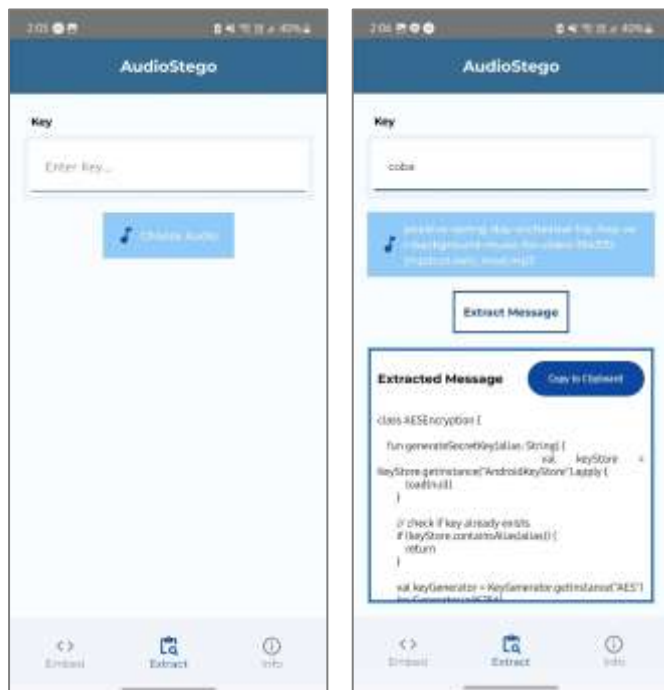


Gambar IV.1 Implementasi antarmuka halaman penyisipan

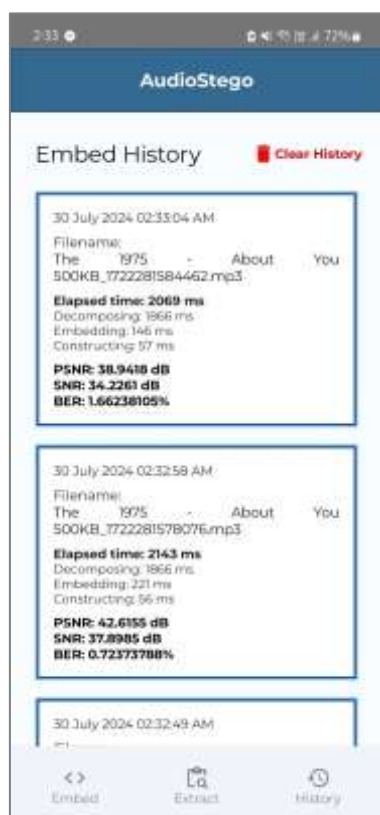
Setelah tombol *Embed Message* diklik, maka proses penyisipan akan berlangsung, dan akan muncul pemberitahuan ketika proses penyisipan telah selesai seperti pada Gambar IV.2.



Gambar IV.2 Pemberitahuan bahwa penyisipan telah berhasil



Gambar IV.3 Implementasi antarmuka halaman ekstraksi



Gambar IV.4 Implementasi antarmuka halaman riwayat penyisipan

Jika pengguna sudah pernah melakukan penyisipan pesan, maka akan tampilan riwayat penyisipan pesan beserta informasi

metrik kualitas audio beserta waktu yang dibutuhkan dari hasil masing-masing penyisipan seperti pada Gambar IV.4.

Untuk pergi ke halaman ekstraksi, pengguna dapat mengklik tombol navigasi yang terdapat pada bagian bawah sehingga akan muncul tampilan awal seperti pada Gambar IV.3. Ketika proses ekstraksi pesan telah berhasil, pengguna dapat melihat pesan yang berhasil diekstrak pada layar dan pengguna juga dapat menyalin pesan tersebut.

B. Implementasi Program

Implementasi program dibuat menjadi beberapa modul yang semuanya ditulis dalam bahasa Kotlin. Pada implementasi program ini, berbagi jenis format audio dapat dipilih sebagai masukan untuk menjadi media penyisipan. Akan tetapi, karena keterbatasan kemampuan untuk melakukan dekomposisi audio agar dapat disisipkan pesan, semua jenis audio akan dikonversi menjadi format MP3 terlebih dahulu, sehingga format keluaran audio yang dihasilkan juga berupa audio MP3. Konversi audio menjadi format awalnya tidak memungkinkan dilakukan karena proses tersebut akan mengakibatkan data yang disisipkan menjadi rusak dan tidak dapat diekstrak lagi. Modul tersebut terbagi menjadi tiga *package* utama, yaitu *cryptography*, *steganography*, dan *utils*. *Package cryptography* menangani proses enkripsi dan dekripsi pesan, *package steganography* menangani proses penyisipan dan ekstraksi pesan, sedangkan *package utils* menyediakan keperluan yang berhubungan dengan sistem operasi Android, seperti membaca nama, ukuran, dan *path* dari sebuah berkas dan juga menyimpan pesan hasil ekstraksi ke dalam berkas. Struktur modul dapat dilihat pada Gambar IV.5.



Gambar IV.5 Struktur modul program

Penjelasan singkat untuk setiap modul adalah sebagai berikut:

- 1) Modul *AESEncryption*, digunakan untuk melakukan proses enkripsi dan dekripsi pada pesan yang ingin disisipkan. Fungsi-fungsi yang terdapat pada modul ini adalah fungsi untuk membangkitkan kunci AES-256, fungsi enkripsi, dan fungsi dekripsi.
- 2) Modul *Embedding*, yang menyediakan berbagai fungsi diperlukan untuk melakukan proses penyisipan pesan ke dalam berkas audio. Fungsi-fungsi yang terdapat pada modul ini adalah fungsi untuk menghitung maksimal pesan yang dapat disisipkan, fungsi untuk menghitung metrik kualitas audio baru, serta fungsi ALSB.
- 3) Modul *Extracting*, yang menyediakan berbagai fungsi yang diperlukan untuk melakukan proses ekstraksi pesan dari berkas audio. Fungsi-fungsi yang terdapat pada modul ini adalah fungsi untuk

membaca penanda awal dan fungsi ekstraksi dengan inversi ALSB.

- 4) Modul *Utils*, yang menyediakan berbagai fungsi untuk menangani pembacaan atribut berkas serta penyimpanan berkas ke sistem penyimpanan Android.
- 5) *MainActivity*, yang digunakan untuk membangun tampilan dari aplikasi serta interaksinya dengan modul-modul lainnya.

V. PENGUJIAN

Pengujian bertujuan untuk mengevaluasi mutu audio setelah pesan disisipkan dibandingkan dengan audio orisinalnya serta membandingkan tingkat kesamaan pesan orisinal dan pesan yang didapatkan dari hasil ekstraksi.

A. Metrik Pengujian

Beberapa metrik penting diukur untuk mengevaluasi kualitas suara setelah penyisipan pesan dan keakuratan pesan yang diekstraksi. Metrik-metrik tersebut adalah *Peak Signal to Noise Ratio* (PSNR), *Signal to Noise Ratio* (SNR), dan *Bit Error Rate* (BER). Selain itu, waktu yang dibutuhkan untuk melakukan penyisipan dan ekstraksi pesan juga diukur untuk mengevaluasi performa dari metode yang diimplementasikan.

Nilai PSNR membandingkan sinyal suara dengan menghitung rasio nilai sinyal maksimum dengan distorsi yang disebabkan oleh proses penyisipan pada tiap byte audio. Berikut ini adalah formula untuk menghitung PSNR:

$$PSNR = 10 \times \log \left[\frac{I_{max}^2}{\frac{1}{N} \sum_{i=1}^N (c_i - s_i)^2} \right] \quad (2)$$

dengan I adalah nilai maksimum dari sinyal audio, N adalah total sampel, serta c_i dan s_i adalah nilai sinyal ke- i dari masing-masing audio orisinal dan audio setelah penyisipan. Nilai sinyal tersebut merupakan representasi nilai integer pada byte ke i .

Nilai SNR hampir mirip dengan PSNR tetapi menggunakan nilai awal byte audio sebelum disisipkan daripada menggunakan nilai maksimum. Berikut ini adalah formula untuk menghitung SNR:

$$SNR = 10 \times \log \left[\sum_{i=1}^N \frac{c_i^2}{(c_i - s_i)^2} \right] \quad (3)$$

dengan N adalah total sampel, serta c_i dan s_i adalah nilai sinyal ke- i dari masing-masing audio orisinal dan audio setelah penyisipan. Nilai PSNR dan SNR yang semakin tinggi mengindikasikan kualitas audio yang dihasilkan lebih baik.

Nilai BER menunjukkan persentase jumlah bit yang berubah pada audio hasil penyisipan pesan dengan total semua bit yang ada pada audio serta tingkat kesamaan pesan yang diekstraksi.

$$BER = 100\% \times \left[\sum_{i=1}^n \frac{b_i \oplus b'_i}{n} \right] \quad (4)$$

dengan n adalah total bit, serta b_i dan b'_i adalah nilai bit ke- i dari masing-masing audio orisinal dan audio setelah penyisipan. Nilai BER Pesan = 0 menunjukkan pesan yang diekstrak 100% sama dengan pesan orisinalnya, sedangkan BER Pesan > 0 menunjukkan terdapat perubahan pada pesan yang dapat terjadi pada proses penyisipan ataupun ekstraksi.

B. Kasus Uji

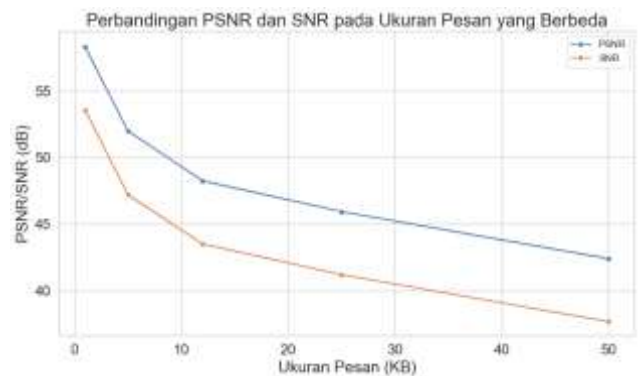
Pengujian terhadap implementasi dilakukan dengan variabel bebas yaitu ukuran pesan dan jenis pesan. Terdapat 5 jenis variasi *genre* audio yaitu *Classic*, *Pop*, *Rock*, *Jazz*, dan *R&B* yang terdiri dari lima lagu sebagai berikut:

- 1) Canon in D - Johann Pachelbel (*Classical*)
- 2) About You - The 1975 (*Pop*)
- 3) There Is a Light That Never Goes Out - The Smiths (*Rock*)
- 4) Waktuku Hampa - Ardhito Pramono (*Jazz*)
- 5) Somebody's Pleasure - Aziz Hedra (*R&B*)

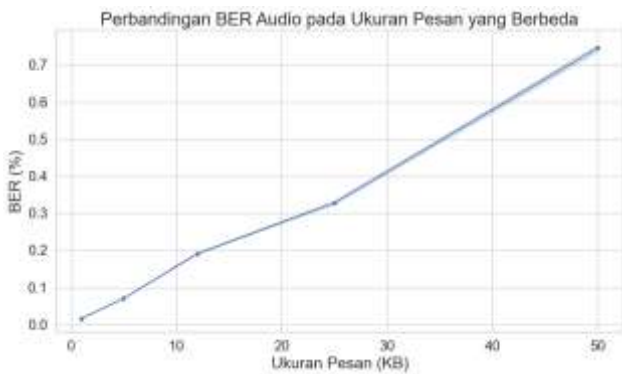
Terdapat tiga variasi jenis pesan yaitu teks, gambar, dan dokumen dengan format PDF. Untuk variasi ukuran pesan, ukuran audio yang digunakan adalah 3 MB dengan ukuran pesan 1 KB dan 5 KB berupa teks (kecil), 12 KB berupa gambar (sedang), serta 25 KB dan 50 KB berupa dokumen dengan format PDF (besar).

C. Hasil Pengujian

Keluaran aplikasi pada tahap pengujian penyisipan adalah audio baru yang telah disisipkan pesan serta metrik kualitas berupa PSNR, SNR, dan BER yang diukur pada proses penyisipan pesan dan disimpan pada halaman riwayat penyisipan. Keluaran aplikasi pada tahap pengujian ekstraksi adalah pesan yang telah disisipkan sebelumnya. Hasil pengujian untuk proses penyisipan dapat dilihat pada Gambar IV.6, Gambar IV.7, Gambar IV.8, Gambar IV.9, Gambar IV.10, dan Gambar IV.11. Hasil pengujian untuk proses ekstraksi pesan dapat dilihat pada Gambar IV.12, Gambar IV.13, dan Gambar IV.14.

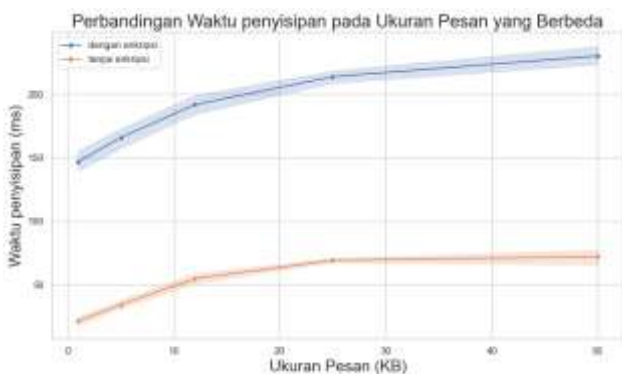


Gambar IV.6 Perbandingan PSNR dan SNR pada ukuran pesan yang berbeda

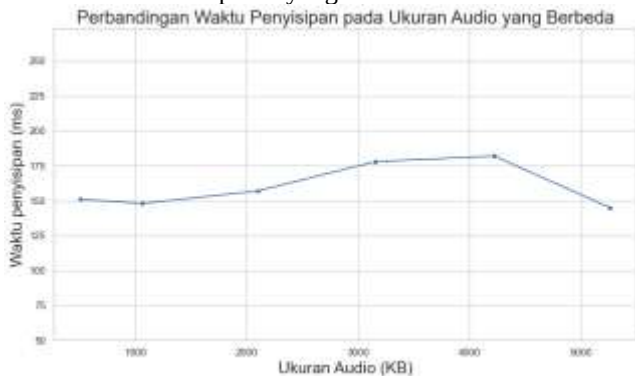


Gambar IV.7 Perbandingan BER Audio pada ukuran pesan yang berbeda

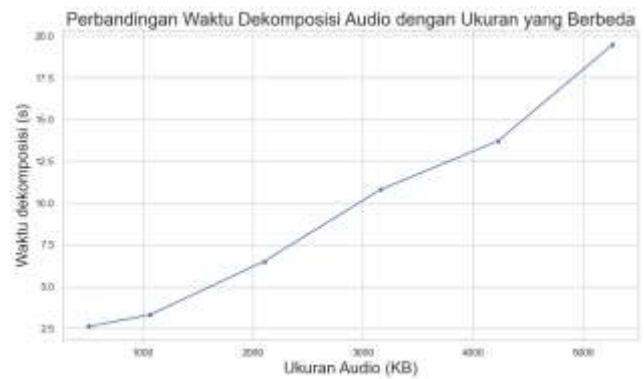
Berdasarkan grafik perbandingan antara nilai PSNR, SNR, dan BER Audio terhadap ukuran pesan yang disisipkan yang terdapat pada Gambar IV.5 dan Gambar IV.6, nilai PSNR dan SNR akan menurun secara logaritmik ketika ukuran pesan yang disisipkan semakin besar dan nilai BER Audio akan meningkat secara linear. Nilai PSNR dan SNR yang menurun menunjukkan penurunan kualitas audio yang dihasilkan dari hasil penyisipan pesan. Begitu juga dengan nilai BER Audio yang semakin meningkat menunjukkan tingkat perbedaan antara audio hasil penyisipan dengan audio orisinalnya semakin meningkat akibat adanya perubahan nilai bit pada penyisipan data.



Gambar IV.8 Perbandingan waktu penyisipan pada ukuran pesan yang berbeda

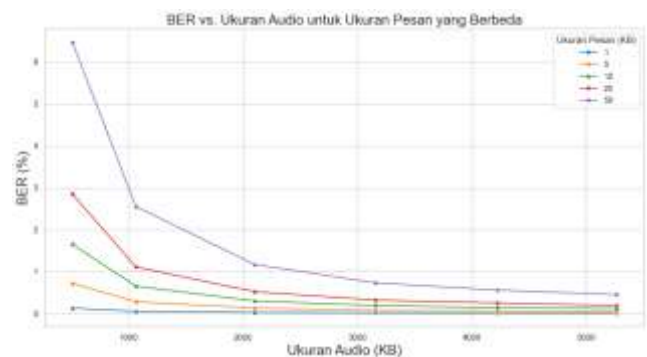


Gambar IV.9 Perbandingan waktu penyisipan pada ukuran audio yang berbeda



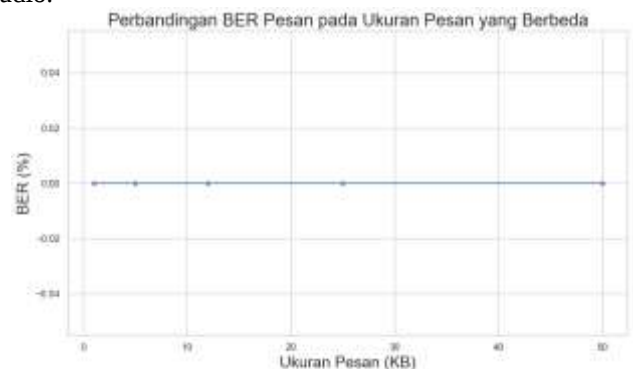
Gambar IV.10 Perbandingan waktu dekomposisi audio pada ukuran yang berbeda

Penambahan ukuran dari pesan yang disisipkan menyebabkan peningkatan waktu yang dibutuhkan untuk proses penyisipan yang dapat dilihat pada Gambar IV.9. Pada Gambar IV.10 penambahan ukuran audio relatif tidak mengubah waktu penyisipan dan ekstraksi pesan dengan ukuran yang sama, tetapi berpengaruh terhadap waktu yang dibutuhkan untuk melakukan dekomposisi audio.



Gambar IV.11 Perbandingan BER audio dengan ukuran audio pada ukuran pesan yang berbeda

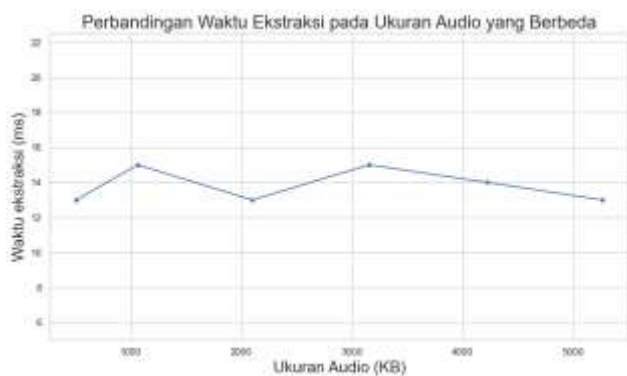
Pada Gambar IV.11 dapat dilihat bahwa nilai BER Audio yang cukup tinggi akibat ukuran pesan yang semakin besar dapat diakomodasi melalui penambahan ukuran dari audio.



Gambar IV.12 Perbandingan BER Pesan pada ukuran pesan yang berbeda



Gambar IV.13 Perbandingan waktu ekstraksi pada ukuran pesan yang berbeda



Gambar IV.14 Perbandingan waktu ekstraksi pada ukuran audio yang berbeda

Pada hasil pengujian untuk proses ekstraksi pesan, didapatkan nilai BER=0 secara konsisten pada masukan kunci steganografi yang tepat yang dapat dilihat pada Gambar IV.12. Nilai ini menunjukkan bahwa selalu berhasil diekstrak dan memiliki tingkat kesamaan 100% dengan pesan orisinalnya yang menandakan integritas pesan terjaga. Gambar IV.13 menunjukkan waktu yang dibutuhkan untuk mengekstrak pesan juga meningkat dengan penambahan ukuran pesan yang terdapat di dalamnya. Pada Gambar IV.14 dapat dilihat bahwa waktu yang dibutuhkan untuk mengekstrak pesan juga meningkat dengan penambahan ukuran pesan yang terdapat di dalamnya.

Proses enkripsi juga memengaruhi waktu dari penyisipan dan ekstraksi pesan yang cukup signifikan, yaitu sekitar tiga kali lebih lama dari waktu penyisipan dan ekstraksi pesan tanpa proses enkripsi terlebih dahulu. Sehingga penyisipan tanpa enkripsi pesan dapat dimanfaatkan pada kasus yang membutuhkan kecepatan pembacaan pesan.

Dari audio hasil penyisipan yang telah dihasilkan, akan dihasilkan efek *silent* pada bagian yang disisipkan bagian dari pesan. Kehadiran efek *silent* tersebut merupakan hal yang wajar karena terjadi modifikasi pada bagian data audio. Nilai PSNR dan SNR masing-masing lebih besar dari 48dB dan 43dB serta

nilai BER Audio yang lebih kecil dari 0.2% menghasilkan audio dengan kemunculan efek *silent* yang tidak terlalu sering dan memiliki kualitas yang cukup baik.

Dari beberapa lagu yang dijadikan kasus uji, karakteristik dari lagu juga dapat mempengaruhi kualitas dari audio yang dihasilkan. Musik dengan ritme yang konsisten sepanjang lagu dengan jeda instrumen atau vokal yang minimum seperti lagu About You – The 1975 menghasilkan *noise* yang lebih mudah untuk dideteksi jika dibandingkan dengan lagu yang memiliki banyak jeda pada instrumen atau vokalnya seperti Canon in D - Johann Pachelbel walaupun kedua lagu tersebut menghasilkan nilai PSNR dan SNR yang sama.

VI. KESIMPULAN

Dalam penelitian ini, telah berhasil dibangun sebuah aplikasi Android yang mengimplementasikan metode Advanced LSB dan AES untuk penyisipan serta ekstraksi pesan. Aplikasi ini mampu menyisipkan dan mengekstraksi pesan dengan ukuran yang bervariasi, mulai dari kecil hingga besar, serta jenis pesan yang berbeda, termasuk teks, gambar, dan dokumen PDF.

Hasil penelitian menunjukkan bahwa kualitas audio steganografi cenderung menurun seiring dengan meningkatnya ukuran pesan yang disisipkan. Namun, meskipun ukuran pesan bertambah, hasil steganografi tetap menunjukkan kualitas audio yang baik dengan nilai metrik PSNR dan SNR masing-masing melebihi 48 dB dan 43 dB. Selain itu, nilai metrik BER Audio yang lebih kecil dari 0,2% menunjukkan bahwa efek *silent* pada audio hasil steganografi sangat jarang terjadi dan tidak berdekatan, yang menandakan kualitas audio yang relatif baik.

Pada penelitian ini, proses dekomposisi audio MP3 memakan waktu yang cukup lama, sehingga alternatif lain misalnya dengan memecah audio menjadi bagian kecil dan melakukan dekomposisi secara paralel dapat dipertimbangkan untuk mempercepat penyisipan dan ekstraksi pesan. Penyisipan dan ekstraksi juga dapat dilakukan langsung pada format lain, seperti M4A, sehingga tidak perlu konversi ke format MP3 dan hasil audio tetap dalam format yang sama dengan format awal.

REFERENSI

- [1] D. Tan, Y. Lu, X. Yan, and X. Wang, "A simple review of audio steganography," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Mar. 2019, pp. 1409-1413.
- [2] F. Hemeida, W. Alexan, and S. Mamdouh, "Blowfish-secured audio steganography," in *2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, Oct. 2019, vol. 1, pp. 17-20.
- [3] A. P. Patil, "AES Hybridization with DWT Audio Steganography," Ph.D. dissertation, National College of Ireland, Dublin, 2019.
- [4] M. S. Atoum, "A Comparative Study of Combination with Different LSB Techniques in MP3 Steganography," in *Information Science and Applications*, Springer Berlin Heidelberg, 2015, pp. 551-560.
- [5] M. S. Atoum, "New mp3 steganography data set," in *2015 5th International Conference on IT Convergence and Security (ICITCS)*, Aug. 2015, pp. 1-7.
- [6] A. Sarkar, A. Goyal, D. Hicks, D. Sarkar, and S. Hazra, "Android application development: A brief overview of android platforms and evolution of security systems," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, Dec. 2019, pp. 73-79.

- [7] R. Siburian, "Steganography implementation on android smartphone using the LSB (least significant bit) to MP3 and WAV audio," in *2017 3rd International Conference on Wireless and Telematics (ICWT)*, Jul. 2017, pp. 170-174.
- [8] S. P. Rajput, K. P. Adhiya, and G. K. Patnaik, "An efficient audio steganography technique to hide text in audio," in *2017 International conference on computing, communication, control and automation (ICCUBE)*, Aug. 2017, pp. 1-6.
- [9] R. Indrayani, H. A. Nugroho, R. Hidayat, and I. Pratama, "Increasing the security of mp3 steganography using AES Encryption and MD5 hash function," in *2016 2nd International Conference on Science and Technology-Computer (ICST)*, Oct. 2016, pp. 129-132.
- [10] O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm for information security," *Symmetry*, vol. 11, no. 12, pp. 1484, 2019.
- [11] M. S. Atoum, M. M. Alnabhan, and A. Habboush, "Advanced LSB technique for audio stenography," in *CoSIT, SIGL, AIAPP, CYBI, CRIS, SEC, DMA*, 2017, pp. 79-86.
- [12] S. Chandra, S. Bhattacharyya, S. Paira, and S. S. Alam, "A study and analysis on symmetric cryptography," in *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, Nov. 2014, pp. 1-8.
- [13] F. Abbasi and P. Singh, "Cryptography: Security and integrity of data management," *Journal of Management and Service Science (JMSS)*, vol. 1, no. 2, pp. 1-9, 2021.
- [14] M. S. Atoum, S. Ibrahimn, G. Sulong, A. Zeki, and A. Abubakar, "Exploring the challenges of MP3 audio steganography," in *2013 International Conference on Advanced Computer Science Applications and Technologies*, Dec. 2013, pp. 156-161.
- [15] R. Krenn, "Steganography and steganalysis," 2004.
- [16] I. F. Ashari, "The Evaluation of Image Messages in MP3 Audio Steganography Using Modified Low-Bit Encoding," *Evaluation*, vol. 14, no. 2, 2021.
- [17] M. Zaturenskiy, "MP3 files as a steganography medium," in *Proceedings of the 2nd Annual Conference on Research in Information Technology, SIGITE/RIIT'13*, Orlando, FL, USA, Oct. 2013. [Online]. Available: doi:10.1145/2512209.2512215.