BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Benny, Diyah, dan Funny (2017) dalam penelitiannya berhasil mengimplementasikan steganografi pesan teks dengan metode *Least Significant Bit* (LSB). Dalam penelitian tersebut *file* audio wav dan mp3 digunakan sebagai media penampung dari pesan. Aplikasi yang dibuat dengan MATLAB tersebut berhasil melakukan penyisipan dan ekstraksi pesan dengan kesesuaian isi pesan 100%. Ukuran audio tidak mengalami perubahan setelah proses penyisipan dilakukan. *File* audio hasil steganografi mempunyai kualitas sangat baik dengan nilai PSNR diatas 50dB. Akan tetapi steganografi dengan metode LSB ini tidak tahan terhadap serangan kompresi yang akan menyebabkan isi pesan di dalam *file* audio hasil steganografi hilang.[3]

Dalam penelitiannya, Rizky, Bambang, dan Gelar (2017) berhasil mengimplementasikan suatu sistem steganografi audio stereo dengan menggabungkan metode *Compressive Sampling* (CS), *Spread Spectrum* (SS), *Lifting Wavelet Transform* (LWT), dan *Singular Value Decomposition* (SVD) dengan sinkronisasi dan *Quantization Index Modulation* (QIM). Tahap yang dilakukan pertama adalah transformasi sinyal audio dengan LWT kemudian didekomposisikan pada matriks nilai singular dengan metode SVD. Sebelum proses embedding, data rahasia dimodulasi dengan metode SS lalu dikompres menggunakan metode CS, kemudian ditambahkan bit sinkronisasi. Berdasarkan hasil penelitian, aplikasi tersebut dapat menghasilkan sistem steganografi audio yang baik dengan kualitas yang tinggi. Hal itu didapatkan berdasarkan beberapa parameter yaitu parameter *Signal to Noise Rasio* (SNR) > 25, *Objective Different Grade* (ODG) > -1, dan *Bit Error Rate* (BER) < 5%, serta pengujian secara subjektif dengan parameter *Mean Opinion Score* (MOS).[4]

Tri Nusanti dan Rinaldo (2019) dalam penelitiannya berhasil menerapkan steganografi pada *file* audio menggunakan metode F5. Algoritma F5 menyisipkan

bit pesan ke dalam bit koefisienn DCT yang diurutkan. Dari Hasil uji coba, diketahui bahwa dengan metode F5 penyisipan dan ekstraksi pesan dapat dilakukan dengan baik. Penyisipan pesan tidak berpengaruh terhadap ukuran berkas audio, akan tetapi berkas audio stego tidak tahan terhadap kompresi, manipulasi amplitudo dan pemotongan audio.[5]

Achmad Fauzi (2019) dalam penelitiannya berhasil melakukan penyisipan pesan ke dalam *file* audio dengan memanfaatkan metode *Data Encryption Standard* (DES) dan metode *End of File* (EOF). Algoritma DES merupakan algoritma kriptografi simetri yang tergolong jenis blok yang beroperasi pada ukuran blok 64bit. DES mengenkripsikan 64bit teks asli menjadi 64bit teks kode menggunakan 56bit kunci internal. Sedangkan metode EOF merupakan salah satu teknik untuk menyisipkan data pada akhir file. Pesan akan dienkripsi menggunakan metode DES sebelum disisipkan ke dalam *file* audio dengan bantuan metode EOF. Berdasarkan hasil pengujian, proses penyisipan pesan dapat dilakukan dengan baik dan pesan dapat terbaca kembali saat proses ekstraksi pesan. File yang ukurannya lebih kecil dapat menampung teks yang ukurannya lebih besar. Ukuran file audio akan bertambah setelah disisipi pesan. [6]

Dalam tesisnya, Lilik Widyawati (2019) melakukan penelitian terkait penjagaan informasi pada media digital yang bertujuan untuk membuat sebuah aplikasi steganografi dengan menggabungkan algoritma SLT (*Slantlet Transform*) dan DCT (*Discrete Cosine Transform*). Algoritma SLT digunakan karena dapat menghasilkan *file* dengan kualitas citra yang lebih baik dibandingkan dengan yang lainnya. Sedangkan algoritma DCT memiliki keunggulan pada ketahanannya pada serangan, meskipun *file* hasil keluarannya tidak memiliki kualitas sebaik algoritma SLT. Kedua algoritma dikombinasikan untuk menutupi kekurangan dari masing-masing algoritma. Proses pengujian dan perhitungan kualitas citra digital dilakukan berdasarkan perhitungan nilai PSNR yang ditentukan oleh nilai MSE. Semakin tinggi nilai MSE citra maka semakin jelas pula perbedaan antar citra asli dengan hasil steganografi, sehingga pesan akan menjadi lebih mudah diketahui. Hasil penelitian ini menunjukkan penerapan algoritma SLT-DCT pada sebuah aplikasi dapat bekerja dengan baik. Kualitas citra steganografi yang dihasilkan

memiliki nilai PSNR rata-rata lebih dari 40db. Dengan nilai PSNR yang tinggi menunjukkan bahwa citra hasil steganografi sulit dibedakan dengan citra yang asli. Citra hasil penerapan algoritma SLT-DCT memiliki ketahanan terhadap proses *convert* atau perubahan format, penambahan *contrast*, dan *histogram* [7].

Selanjutnya Bayu Angga, Adrian, Bayu Andreas, dan Lovely Septian (2021) melakukan sebuah penelitian dalam jurnalnya yang berjudul "*Steganography Text Message Using LSB and DCT Methods*". Penelitian tersebut bertujuan untuk mengatasi permasalahan seperti pencurian dan pemalsuan data dengan membuat aplikasi yang menerapkan konsep steganografi menggunakan metode LSB dan DCT pada sebuah gambar untuk kemudian diuji tingkat efisiennnya dalam menyembunyikan pesan. Dalam penerapannya, gambar yang akan diisi pesan harus melewati proses konversi terlebih dahulu ketika menggunakan metode DCT untuk kemudian bisa dilakukan tahap *encoding*, sedangkan pada metode LSB gambar tidak memerlukan proses konversi terlebih dahulu. Dari penelitian ini dapat disimpulkan bahwa aplikasi berjalan dengan baik dan metode DCT menghasilkan kualitas citra yang lebih baik dibandingkan LSB. Metode DCT mendapatkan nilai rata-rata PSNR 83,7728 dengan nilai MSE 0,0592 dan metode LSB mendapat nilai PSNR 8,887 dengan nilai 104,85 [8].

Mohammad Hamdani dan Gloria Samosir (2018) melakukan penelitian pada jurnalnya yang berjudul "Implementasi Steganografi untuk Keamanan Pengiriman Citra Digital Menggunakan Metode DCT (*Discrete Cosine Transform*)". Penelitian bertujuan untuk merancang sebuah algoritma implementasi steganografi citra digital dengan metode DCT. Perancangan ini terdapat dua bagian yaitu bagian pengirim yang berisi penyisipan data digital berformat bmp *grayscale* ke dalam *file* bmp RGB dan bagian penerima sebagai tempat ekstraksi data. Proses *encode* dan *decode* dilakukan dengan metode DCT yang melakukan penyisipan lewat domain frekuensi. Metode DCT menghasilkan koefisienn DCT yang akan dikuantisasi dengan matriks kuantisasi standar JPEG. Penelitian ini menghasilkan aplikasi yang dapat menerapkan steganografi dengan metode DCT dengan baik. Setiap hasil steganografi kemudian diuji performanya menggunakan metode *Mean Square Error* (MSE), *Peak Signal to Noise Ratio* (PSNR), dan

Mean Opinion Score (MOS). Nilai PSNR citra steganografi menunjukkan angka lebih dari 40db. Tingkat kemiripan mendapatkan hasil yang dapat diterima dengan baik meskipun terdapat sedikit perubahan resolusi pada citra [9].

Amin Samsudin, Andy Haryoko, dan Adityo Nugroho (2018) dalam penelitiannya berhasil merancang aplikasi steganografi berbasis perangkat *mobile*. Metode *Least Significant Bit* (LSB) diterapkan dalam aplikasi ini untuk menyisipkan teks ke dalam berkas audio mp3. Dari hasil pengujian, aplikasi ini dapat melakukan proses penyisipan dan ekstraksi pesan dengan baik dan dalam waktu yang cepat. Dalam tahap pengujian digunakan juga nilai *Peak Signal to Noise Ratio* (PSNR) sebagai salah satu parameter penilaian objektif untuk mengukur kelayakan hasil steganografi. Rata-rata nilai PSNR yang didapatkan adalah 32,02dB yang dapat disimpulkan bahwa hasil steganografi dari aplikasi ini memiliki nilai yang cukup baik. [10]

Nur Qodariyah Fitriyah dan Yusuf Yudi Prayudi (2017) dalam penelitiannya berhasil menerapkan mengimplementasikan steganografi dengan metode *Discrete Cosine Transform* (DCT). Metode DCT yang digunakan berfokus pada satu dimensi. Data yang digunakan sebagai *carrier* dan pesan merupakan file audio yang memiliki format wav. Pengujian dilakukan dengan menggunakan metode IDCT (*Inverse Discrete Cosine Transform*) untuk mengetahui kondisi kapasitasnya, dimana diperoleh tingkat keberhasilan sampai dengan 90% dimana dengan hasil ini metode DTC cukup baik digunakan. [11]

Tahun 2019 dalam penelitiannya, Rama Edwinda Putra dan Rifki Rijkulloh melakukan sebuah implementasi steganografi teks pada *file* audio menggunakan bahasa pemrograman *Python*. Dalam penelitian tersebut, proses steganografi *file* audio dilakukan dengan metode *Least Significant Bit* (LSB). Teks diubah menjadi bentuk biner dan disisipkan pada bagian akhir nilai biner dari *file* audio. *File* audio yang digunakan dalam penelitian ini memiliki format *file* wav. Hasil penelitian menunjukkan bahwa, program dapat berjalan dengan baik untuk melakukan audio steganografi dengan bahasa *python*. Pada tahap pengujian, perubahan *decibel* pada hasil steganografi dapat menghasilkan hasil ekstraksi kumpulan teks acak. Hal ini disebabkan proses ini mengubah seluruh struktur *file*

audio yang sudah disisipkan teks. Kemudian, pesan tidak akan terekstraksi jika melakukan pemotongan pada bagian awal *file* audio. Pesan akan terurai saat dilakukan pemotongan pada bagian akhir audio. Hal ini disebabkan karena pesan disisipkan pada bagian akhir nilai biner *file* audio dengan metode LSB [12].

Tabel 2.1 Ringkasan Penelitian Terdahulu

No	Judul	Metode	Hasil	Perbedaan dengan Penelitian yang dilakukan
1	Perancangan Dan Pembuatan Aplikasi Steganografi Pesan Teks Pada Audio Digital Dengan Metode *Least Significant Bit* oleh Benny Kuniadi S Y, Diyah Puspitaningrum, Funny Farady Coastera (2017).	• Metode: *Least Significant Bit* (LSB) • Pesan: teks • *Carrier*: *file* audio • Pengujian: *encoding, decoding,* serangan kompresi, dan PSNR • Dibuat dengan MATLAB	• Menghasilkan aplikasi steganografi pesan teks pada file audio wav dan mp3 dengan metode LSB berbasis desktop. • Berhasil melakukan *encoding* dan *decoding*. • File audio stego tidak tahan terhadap serangan kompresi. • Menghasilkan nilai PSNR di atas 50db.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB
2	Steganografi Audio Stereo Tersinkronisasi Berbasis SS dengan Metode Gabungan LWT-SVD oleh Rizky Eka Liyanty, Bambang Hidayat, Gelar Budiman (2017)	• Metode: *Compressive Sampling (CS), Spread Spectrum (SS), Lifting Wavelet Transform (LWT), dan Singular Value Decomposition (SVD) dengan sinkronisasi dan Quantization Index Modulation (QIM)* • Pesan: teks • *Carrier*: *file* audio • Pengujian: SNR, ODG, BER, MOS	• Menghasilkan sistem steganografi audio yang baik dengan kualitas yang tinggi. • *Signal to Noise Rasio* (SNR) > 25 • *Objective Different* • *Grade* (ODG) > -1 • *Bit Error Rate* (BER) < 5%	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB

No	Judul	Metode	Hasil	Perbedaan dengan Penelitian yang dilakukan
3	Penyisipan Pesan Rahasia Ke dalam Audio Menggunakan Algoritma F5 oleh Tri Nusanti Sianturi, Rinaldo Gomgom Hutagaol Prodi (2019)	• Metode: *Algoritma F5* • Pesan: gambar • Pengujian: kompresi ketahanan data, PSNR • *Carrier*: *file* audio	• Berhasil mengimplementasikan steganografi gambar ke dalam file audio dengan algoritma F5. • Berkas audio stego tidak tahan terhadap kompresi, manipulasi amplitudo dan pemotongan audio. • Nilai PSNR di atas 60dB.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB
4	Analisa Kombinasi Pesan Teks Ke Dalam File Audio Memanfaatkan Algoritma Data Encryption Standard Dan Metode *End of File* oleh Achmad Fauzi (2019).	• Metode: DES, EOF • Pesan: teks • Pengujian: kompresi ketahanan data, PSNR • *Carrier*: *file* audio	• Penyisipan pesan dapat dilakukan dengan baik. • Pesan dapat terbaca kembali saat proses ekstraksi pesan. • File yang ukurannya lebih kecil dapat menampung teks yang ukurannya lebih besar. • Ukuran file audio akan bertambah setelah disisipi pesan.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB

No	Judul	Metode	Hasil	Perbedaan dengan Penelitian yang dilakukan
5	Steganografi Pada File Audio Mp3 Untuk Pengamanan Data Menggunakan Metode *Least Significant Bit* oleh Amin Samsudin, Andy Haryoko, dan Adityo Nugroho (2018)	• Metode: *Least Significant Bit* (LSB) • Pesan: teks • *Carrier*: *file* audio mp3 • Pengujian: *encoding, decoding,* kecepatan proses, dan PSNR • Berbasis aplikasi *mobile*	• Menghasilkan aplikasi *mobile* yang dapat melakukan teknik steganografi dengan baik. • Waktu yang dibutuhkan untuk setiap proses steganografi yang dilakukan relatif cukup cepat. • Nilai PSNR yang didapat rata-rata adalah 32.02dB	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB
6	Implementasi Steganografi Audio File Wav Dengan Metode *Discrete Cosine Transform* (DCT) oleh Nur Qodariyah Fitriyah dan Yusuf Yudi Prayudi (2017)	• Metode: *Discrete Cosine Transform* (DCT) satu dimensi • Pesan: *file* audio wav/mp3 • *Carrier*: *file* audio wav/mp3 • Pengujian: *encoding, decoding,* membandingkan kapasitas file asli dan file stego • Dibuat dengan MATLAB	• Menghasilkan aplikasi yang dapat melakukan penyisipan pesan berbentuk audio ke dalam *file* audio dengan baik. • Mendapatkan tingkat keberhasilan sampai dengan 90% • Ukuran berkas mengalami penurunan yang cukup signifikan setelah proses steganografi.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB

No	Judul	Metode	Hasil	Perbedaan dengan Penelitian yang dilakukan
7	Implementasi Metode Steganografi SLT-DCT pada Citra Untuk Meningkatkan Kualitas Citra Steganografi, oleh Lilik Widywati (2019)	• Metode: *Slantlet Transform (SLT)* dan *Discrete Cosine Transform (DCT)* • Pesan: teks format txt • *Container*: Citra digital dengan ukuran 256x256, 512x512 dan 1024x1024 piksels • Dibuat dengan MATLAB R2018a.	• Menghasilkan aplikasi yang mampu melakukan steganografi pada citra dengan baik, dengan hasil *file* output yang memiliki kualitas setara dengan metode SLT dan memiliki ketahanan serupa dengan metode DCT dengan nilai rata-rata PSNR di atas 40db.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB
8	*Steganography Text Message Using LSB and DCT Methods*, oleh Bayu Angga, Adrian, Bayu Andreas, dan Lovely Septian (2019)	• Metode: L*east Significant Bit* (LSB) dan *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: gambar format png	• Aplikasi dapat melakukan steganografi dengan baik untuk setiap metode yang digunakan (LSB dan DCT) dengan keunggulan masing-masing pada hasil output. • Metode LSB lebih efisien dibandingkan DCT, tetapi hasil dari metode DCT lebih baik • Semakin besar pesan yang disembunyikan semakin kecil nilai PSNR dan kuliatas *file* audio stego.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB

No	Judul	Metode	Hasil	Perbedaan dengan Penelitian yang dilakukan
9	Implementasi Steganografi untuk Keamanan Pengiriman Citra Digital Menggunakan Metode DCT (Discrete Cosine Transform) oleh Mohammad Hamdani dan Gloria Samosir (2018)	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: citra digital format .bmp RGB pada bagian pengirim, hasil steganografi DCT berformat JPEG	• Aplikasi dapat menyispkan pesan pada citra digital menggunakan metode DCT. • Nilai PSNR citra steganografi menunjukkan angka di atas 40db. • Proses pengiriman paling lama terjadi pada citra ukuran 48x48 piksel dengan 13.668 detik.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB
10	Implementasi Audio Steganografi Teks ke Audio pada Python oleh Rama Edwinda Putra dan Rifki Rijkulloh (2019)	• Metode: Metode *Least Significant Bit* (LSB), bahasa pemrograman *python* • Pesan: teks • *Container*: audio format wav	• Program berjalan dengan baik untuk audio steganografi. • Perubahan *decibel file* audio stego menghasilkan kumpulan teks acak. • Pemotongan bagian awal tidak akan menampilkan pesan.Pemotongan bagian akhir akan menampilkan pesan.	• Metode: *Discrete Cosine Transform* (DCT) • Pesan: teks • *Container*: audio format wav • Pengujian: metode *black box* dan nilai PSNR. • Dibuat dengan MATLAB

2.2 Dasar Teori

2.2.1 Audio

Audio atau suara merupakan sebuah fenomena fisik yang dihasilkan oleh suatu getaran berupa sinyal analog dengan amplitudo yang berubah secara kontinu terhadap satuan waktu atau frekuensi. Perbedaan tekanan udara terjadi selama proses getaran berlangsung dan dinamakan sebagai gelombang. Pada interval tertentu gelombang memiliki kecenderungan pola yang sama disebut periode [13].

Sebagai proses digitalisasi terhadap format rekaman musik analog, musik atau audio digital mempunyai beraneka ragam format yang bergantung pada teknologi yang digunakan [14]. Terdapat beberapa jenis format *file* audio yang dikenal masyarakat. Setiap jenis *file* audio memiliki karakteristik yang berbeda-beda yang dapat dipilih seseuai kebutuhan pengguna. Berikut merupakan beberapa jenis *file* audio yang sering dijumpai:

1. *Advance Audio Coding* (AAC)

Advance Audio Coding (AAC) merupakan jenis *file* audio yang berbasis MPEG2 dan MPEG4 yang dikembangkan oleh *Motion Picture Expert Group* (Fraunhofer Institute, Dolby, Sony, Nokia dan AT&T). *File* ini bersifat *lossy compression* dan memiliki lisensi yang tidak gratis. AAC merupakan sebuah audio codec yang menyempurnakan dalam hal medium dan efisien bit rates serta dinilai lebih efisienn pada kecepatan 128kbps dengan suara stereo dibandingkan format audio mp3 [15].

2. MPEG Layer 3 (MP3)

MPEG Layer 3 atau mp3 adalah sebuah format kompresi audio yang dikembangkan oleh Moving Picture Experts Group (MPEG). *File* dengan format ini termasuk *file* yang paling dikenal oleh masyarakat karena sering digunakan untuk menyimpan *file* musik, lagu, dan *audiobooks*. Kualitas suara yang dihasilkan jenis *file* ini mendekati kualitas CD stereo dengan 16-bit.

Kualitas suara *file* mp3 tergantung pada sebagian besar bit rate yang digunakan untuk kompresi. Kisaran Bit rate yang sering digunakan yaitu 128, 160, 192, 256 dan 320 kbps. Semakin besar bit rate, semakin bagus kualitasnya, namun ukuran *file* juga akan ikut membesar. Sebagai *file* kompresi, mp3 menggunakan

teknik lossy compression, sehingga ada kemungkinan kualitas sebuah *file* akan berkurang jika dikonversi ke dalam mp3.

Pada tahun 2001, MP3 Pro generasi berikutnya diperkenalkan dan menawarkan kualitas suara dan kompresi yang sudah ditingkatkan, namun karena tidak ada decoder MP3 Pro gratis, format *file* ini belum bisa menggantikan kepopuleran format mp3 yang standar [15].

3. Ogg dan Ogg Vorbis

Ogg adalah format multimedia gratis yang dirancang untuk kebutuhan streaming dan penyimpanan yang efisienn. Format ini dikembangkan oleh Xiph.org Foundation. Begitu pula Vorbis yang merupakan codec audio gratis. Vorbis biasanya dipasang bersama Ogg, sehingga muncullah yang namanya Ogg Vorbis. Peluncuran format dan *codec* ini merupakan respon atas rencana pemilik mp3 pada tahun 1998 yang hendak mengenakan biaya lisensi untuk format mp3. OggVorbis sangat populer dikalangan open source, karena kualitas dan sifatnya yang gratis. Meskipun gratis, penggunaan format ini masih sedikit dijumpai. Winamp menjadi salah satu platform yang mendukung *file* format Ogg Vorbis [15].

4. *Windows Media Audio* (WMA)

Windows Media Audio (WMA) adalah sebuah codec untuk *lossy compression* yang dikembangkan pertama kali juga oleh microsoft untuk menyaingi mp3. Sementara ini Microsoft menempatkan WMA untuk bersaing dengan AAC yang digunakan pada produk Apple seperti iPod dan iTunes Music Store. WMA menggunakan sistem *Digital Rights Management* seperti AAC untuk proteksi penggandaan dan membatasi pemutaran pada PC atau perangkat tertentu tertentu. WMA audio stream hampir selalu dengan *file Advanced Systems Format* (ASF). Jika hanya membawa data audio, biasanya *file* mempunyai ekstensi wma. Terdapat juga versi lossless untuk multichannel surround sound dan untuk voice encoding (WMA Voice) [15].

5. *Free Lossless Audio Codec* (FLAC)

Free Lossless Audio Codec (FLAC) merupakan format audio kategori *lossless* yang paling banyak digunakan, menjadikannya pilihan terbaik bagi

pengguna yang ingin menyimpan audio dengan sedikit mengurangi kualitas suaranya. Tidak seperti format wav yang tidak mengalami kompresi, FLAC tetap mengalami kompresi meskipun itu *lossless* [15].

6. *Pulse Code Modulation* (PCM)

Pulse Code Modulation (PCM) *Raw Data*, PCM (*Pulse Code Modulation*) adalah format audio yang sangat sederhana. Format ini adalah format *file* standar yang belum dikompres seperti halnya *file* wav pada Windows atau *Audio Intercharge File Format* (AIFF) pada Apple. PCM juga dikenal sebagai Linear PCM yang digunakan sebagai standar untuk format CD Audio. Karena sifatnya yang tidak terkompresi, maka format PCM ini mempunyai ukuran *file* yang cukup besar bila dibandingkan dengan format mp3 [14].

7. *Wave-form audio format* (WAV)

Wave-form audio format (WAV) adalah sebuah format audio standar yang dikembangkan Microsoft dan IBM untuk personal komputer. WAV biasanya menggunakan coding PCM (Pulse Code Modulation). WAV berisikan data yang tidak terkompres sehingga seluruh sampel audio disimpan semua pada media penyimpanan dengan bentuk digital. Software yang dapat menciptakan WAV dari analog sound misalnya adalah Windows Sound Recorder [13].

Format *file* wav umumnya digunakan untuk menyimpan audio tanpa kompresi dan *file* suara berkualitas CD. *File* wav juga dapat ditemukan pada *file* suara-suara efek dan musik maupun backsound dari game. *File* wav juga dapat berisi data terkodekan dengan beraneka ragam codec untuk mengurangi ukuran *file*. Akan tetapi untuk keperluan mengoleksi musik, transfer via internet dan memainkan di player portable, format ini kurang populer dibandingkan dengan mp3, Ogg Vorbis dan WMA dikarenakan ukuran *file* yang sangat besar [12].

File wav menggunakan sample rate 44,1KHz yang berarti audio dengan format wav memainkan 44.100 sample setiap detiknya dan setiap sample berukuran 2byte untuk mono, dan 4byte untuk stereo. Sehingga perhitungan untuk memainkan sebuah audio wav stereo berdurasi 4 menit adalah 2 x 2byte x 44.100 kali/detik x 4 x 60 detik = 42.336.000 byte (42MB). Sebagai perbandingan, perhitungan untuk memainkan sebuah *file* audio dengan format mp3 dengan bit

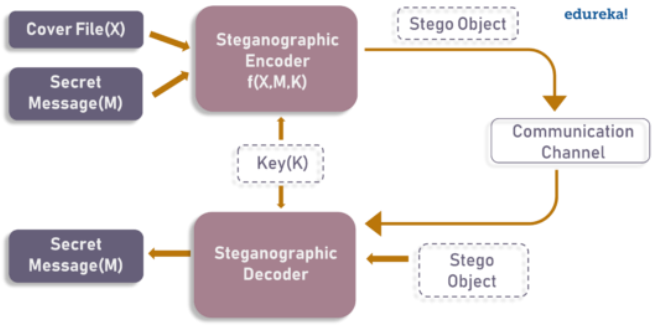
rate 128 kbps berdurasi 4 menit adalah 128 kbps x 4 x 60 detik = 30720 kilobit (4,25MB) [16].

Dari kedua perhitungan di atas, dapat disimpulkan bahwa untuk memainkan *file* audio berdurasi 4 menit dengan format wav membutuhkan kapasitas penyimpanan yang jauh lebih besar dibandingkan *file* audio berdurasi 4 menit dengan format mp3. Hal tersebut disebabkan oleh kerapatan sample rate format wav yang jauh lebih rapat dibandingkan format mp3 [16]. Itulah alasan *file* wav biasanya memiliki ukuran yang relatif lebih besar.

Berdasarkan keterangan tersebut bahwa *file* wav berisi data atau sample audio yang tidak terkompresi maka *file* format wav dinilai sesuai untuk dijadikan media steganografi. Hal ini dikarenakan data dalam *file* audio belum mengalami proses perubahan atau kompresi sehingga *file* audio tidak akan rusak karena terlalu banyak melewati proses pengubahan, baik kompresi maupun transformasi. Sehingga diharapkan tidak akan terjadi perubahan yang mencolok pada audio yang digunakan sebagai pembawa pesan dalam penelitian ini.

2.2.2 Steganografi

Steganografi berasal dari kata Yunani "*steganos*" yang berarti "tersembunyi" dan "*graphein*" yang berarti "menulis" [17]. Steganografi adalah ilmu dan teknik yang menggunakan algoritma untuk menulis dan menyembunyikan pesan, dan keberadaan pesan hanya diketahui oleh pemilik dan penerima [18].



Gambar 2.1 Contoh Skema Steganografi[19]

Tujuan dari kegiatan steganografi adalah untuk merahasiakan atau menyembunyikan keberadaan sebuah informasi atau data penting. Pada umumnya

pesan disembunyikan dengan membuat perubahan seminimal mungkin pada data digital lain. Perubahan ini bergantung pada kunci dan pesan untuk disembunyikan, serupa dengan metode kriptografi. Orang yang menerima gambar kemudian dapat menyimpulkan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan untuk menyembunyikan pesan [20].

Beberapa kriteria harus diperhatikan saat melakukan steganografi. Kriteria tersebut antara lain:

1. *Imperceptibility* yaitu keberadaan pesan dalam media penampung tidak dapat dideteksi.
2. *Fidelity*, kualitas dari media penampung setelah proses penambahan pesan tidak jauh berbeda dengan kualitas sebelum proses penyisipan.
3. *Recovery*, pesan rahasia yang telah disisipkan ke dalam media penampung harus dapat diungkap atau diekstrak kembali.
4. *Robustness*, pesan yang disembunyikan harus memiliki ketahanan terhadap serangan manipulasi. [21]

Terdapat beberapa metode yang dapat digunakan untuk menerapkan steganografi. Metode tersebut memiliki konsep dan keunggulan masing-masing dibandingkan metode lainnya. Adapun metode steganografi yang disebutkan meliputi:

1. *Least Significant Bit* (LSB)

Steganografi dengan metode LSB dilakukan dengan menyisipkan pesan pada bit terakhir atau paling kanan dari *file* asli atau *cover image*. Pada sebuah *file* bitmap 24 bit, setiap pikselnya tersusun dari tiga susunan warna merah, hijau, dan biru. Masing-masing bagian tersebut disusun oleh bilangan 8bit (byte) dari 0 sampai 255 atau dalam format biner 00000000 sampai dengan 11111111. Dengan kata lain, pada setiap piksel *file* bitmap 24bit data disisipi 3bit pesan [22].

2. Algorithms and Transformation

Metode steganografi *Algorithms and Transformation* merupakan sebuah metode untuk menyembunyikan data melalui fungsi matematika yang disebut algoritma *compression*. Terdapat dua fungsi yang termasuk ke dalam algoritma *compression*, yaitu *Discrete Cosine Transformation* (DCT) dan *Wavelet*

Transformation. Fungsi DCT mentransformasi data pada domain spasial ke domain frekuensi, sedangkan *Wavelet Transformation* mentransformasikan data dari satu domain ke domain yang lainnya [22].

3. Redundant Pattern Encoding

Redundant Pattern Encoding yaitu suatu cara untuk menggambar pesan kecil pada kebanyakan gambar. Keuntungan dari metode ini adalah dapat bertahan dari cropping, kekurangannya metode ini tidak dapat menggambar pesan dengan ukuran yang lebih besar [22].

4. Spread Spectrum Method

Spread Spectrum adalah sebuah teknik transmisi dengan penggunaan kode *pseudo noise* sebagai gelombang modulasi untuk menyebarkan energi sinyal melalui sebuah bandwith yang jauh lebih besar daripada *bandwidth* sinyal informasi. Ide dasar metode ini adalah untuk menyebarkan sinyal informasi melalui bandwidth yang lebih luas untuk mencegah dilakukannya pencegatan informasi dan gangguan - gangguan lainnya. Proses penyebaran *bandwidth* sinyal informasi disebut spreading. Penyebaran ini berguna untuk menambah tingkat redundansi. Besaran redundansi ditentukan oleh faktor pengali x yang bernilai skalar. Panjang bit-bit hasil penyebaran ini menjadi x kali panjang bit-bit awal [23].

2.2.3 MATLAB

MATLAB merupakan akronim dari kata *Matrix Laboratory* dan salah satu platform yang pertama kali dirilis oleh MathWorks pada tahun 1970. MATLAB adalah salah satu jenis platform pemrograman yang menggunakan bahasa dengan kinerja tinggi berbasis matriks. Platform ini biasanya digunakan untuk menganalisis data, membuat algoritma, serta melakukan pemodelan aplikasi. MATLAB mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar. Penggunaan MATLAB meliputi bidang matematika dan komputasi, engineering, mengolah permasalahan aljabar linear, analisis numerik dan data riset, serta

melakukan simulasi dan pemodelan [1].

Fitur-fitur MATLAB sudah banyak dikembangkan, dan lebih kita kenal dengan nama *toolbox*. *Toolbox* ini merupakan kumpulan dari fungsi-fungsi seperti sum, sin, cos, dan complex arithmetic, sampai dengan fungsi-fungsi yang lebih kompleks seperti *matrix inverse, matrix eigen values, bessel functions, dan fast fourier transforms*.

Terdapat enam bagian yang merupakan sistem utama MATLAB, yaitu:

1. MATLAB *language*

   MATLAB menggunakan *high-level matrix array language* yang mampu mengolah program atau fungsi yang kompleks.

2. Working *environment*

   Bagian ini berisi kumpulan tools dan fasilitas yang diberikan oleh MATLAB. Dengan fitur ini pengguna dapat mengelola variabel, melakukan *import, eksport*, dan lain sebagainya.

3. Sistem grafis

   Sistem grafis MATLAB adalah bagian yang digunakan untuk memproses gambar, visualisasi data, membuat animasi, dan mempresentasikan grafis.

4. Mathematical function library

   Berisi kumpulan fungsi matematis yang dapat diproses dengan kecepatan tinggi.

5. *Application Program Interface* (API)

   MATLAB memberikan akses untuk menulis program C dan Fortran.

6. MATLAB *Guide*

   MATLAB *Guide* merupakan sebuah fitur yang dapat digunakan *user* untuk membuat *Graphical User Interface* (GUI) atau tampilan antarmuka dengan lebih mudah pada saat melakukan pengembangan sebuah aplikasi. Dengan fitur ini, *user* hanya perlu memilih dan menarik komponen yang tersedia ke area lembar kerja. Komponen – komponen tersebut juga dapat disesuaikan kembali dengan memanfaatkan *toolbar* dan menu lainnya yang tersedia.

2.2.4 *Discrete Cosine Transform* (DCT)

Discrete Cosine Transform (DCT) termasuk ke dalam metode steganografi *algorithms and transformation* yang merupakan suatu metode yang biasa digunakan untuk mengubah sebuah sinyal menjadi komponen frekuensi dasarnya. DCT mengubah sebuah kawasan spasial menjadi kawasan frekuensi dan sebaliknya kawasan frekuensi dapat dikembalikan ke kawasan spasial dengan menggunakan invers DCT [1] [18]. DCT merupakan transformasi matematis yang mengambil sinyal dan mentransformasikannya dari domain spasial ke domain frekuensi. DCT pertama kali diperkenalkan oleh Ahmed, Natarajan dan Rao pada tahun 1974 dalam makalahnya yang berjudul "*On Image Processing and a Discrete Cosine Transform*" (Watson, 1994).

Data ditanamkan ke citra *cover* dengan mengganti koefisienn transformasi dari sebuah citra seperti koefisienn transformasi kosinus diskrit. DCT dua dimensi diterapkan pada seluruh piksel citra cover dan dilakukan proses *embedding*. Rumus yang digunakan dalam metode *Discrete Cosine Transform* (DCT) pada data berbentuk matriks MxN pada indeks ke (x, y) yang menghasilkan D (i, j) sebagai nilai dari hasil transformasi DCT dapat dituliskan sebagai berikut:

$$D(i,j) = \sqrt{\frac{2}{M}}\sqrt{\frac{2}{N}}C(i)C(j)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}Z(x,y)\cos\left(\frac{(2x+1)i\pi}{2M}\right)\cos\left(\frac{(2y+1)j\pi}{2N}\right) \quad (2.1)$$

dengan

D (i, j) = nilai transformasi pada indeks ke (x, y)

M = ukuran baris matriks

N = ukuran kolom matriks

Z (x, y) = nilai bit pada indeks ke (x, y)

C(i), C(j) = 1 jika i > 0, j > 0

C(i), C(j) = $\frac{1}{\sqrt{2}}$ jika i = 0 0, j = 0[25]

Sedangkan untuk persamaan invers DCT yang digunakan untuk mengubah nilai transformasi D (i, j) ke bentuk awal Z (x, y) dalam matriks MxN dapat dinyatakan sebagai berikut:

$$Z(x,y) = \sqrt{\frac{2}{M}}\sqrt{\frac{2}{N}}C(i)C(j)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}D(i,j)\cos(\frac{(2x+1)i\pi}{2M})\cos(\frac{(2y+1)j\pi}{2N}) \quad (2.2)$$

dengan

D (i, j) = nilai transformasi pada indeks ke (x, y)

M = ukuran baris matriks

N = ukuran kolom matriks

Z (x, y) = nilai bit pada indeks ke (x, y)

Penerapan metode DCT tidak hanya terpaku pada dimensi 2 atau dengan matriks MxN. DCT memungkinkan untuk dilakukan pada 1 dimensi atau dengan matriks 1xN, seperti yang diterpakan pada penelitian ini. Dalam penerapannya, tidak ada perubahan yang signifikan dari segi persamaannya. Dikarenakan jumlah baris matriks 1, perhitungan hanya akan dilakukan disepanjang kolom matriks. Sehngga persamaan yang digunakan untuk mentransformasikan data pada koordinat (1, y) menjadi betuk D (j) dapat dinyatakan sebagai berikut:

$$D(j) = \sqrt{\frac{2}{N}} C(j) \sum_{x=0}^{N-1} Z(1,y) \cos\left(\frac{(2y+1)j\pi}{2M}\right) \quad (2.3)$$

dengan

D (j) = nilai transformasi pada indeks ke (1, y)

N = ukuran kolom matriks

Z (1, y) = nilai bit pada indeks ke (1, y)

C(j) = 1 jika i > 0, j > 0

C(j) = $\frac{1}{\sqrt{2}}$ jika i = 0 0, j = 0

Sebagai perumpamaan, terdapat sebuah *file* A yang akan digunakan sebagai media steganografi dengan bentuk data bit sebagai berikut:

A = [1 2 3 4 5 6 7 8 9 10]

Kemudian *file* tersebut dikonversi dengan metode DCT sesuai dengan persamaan di atas.

A’ = [6 0.1 0.2 0.3 -0.4 0.5 -0.7 0.9 0.6 0.8]

Terdapat sebuah pesan yang berisi huruf F atau dalam biner 11100001 yang akan disisipkan pada bagian tengah *file* A. Sehingga pesan F yang memiliki 8bit data akan dimasukkan pada bit ke-2 sampai ke-9 dari total 10bit data.

Pesan dimasukkan dengan memodifikasi bit data pada *file* asli. Logika yang digunakan adalah jika nilai bit pada pesan bernilai 1, maka nilai bit pada *file* asli di ubah ke bentuk negatif dan sebaliknya jika nilai bit pesan bernilai 0, maka nilai bit pada *file* asli diubah diubah ke bentuk positif. Sehingga hasilnya sebagai berikut:

A' = [6 0.1 0.2 0.3 -0.4 0.5 -0.7 0.9 0.6 0.8]

F = [1 1 1 0 0 0 0 1]

A" = [6 -0.1 -0.2 0.3 0.4 0.5 0.7 0.9 -0.6 0.8]

Selanjutnya, hasil dari modifikasi bit atau penyisipan pesan diubah kembali ke bentuk semula dengan metode DCT lagi atau disebut invers DCT. Sehingga bentuk *file* akan serupa dengan *file* asli, namun di dalamnya sudah tersimpan sebuah pesan rahasia.

Discrete Cosine Transform merepresentasikan sebuah citra dari penjumlahan sinusoida dari magnitude dan frekuensi yang berubah-ubah. Sifat dari DCT adalah mengubah informasi citra yang signifikan dikonsentrasikan hanya pada beberapa koefisienn DCT. Oleh karena itu DCT sering digunakan untuk kompresi citra seperti pada JPEG. Selain itu, penggunaan metode DCT untuk kebutuhan steganografi dinilai relevan, karena metode DCT dapat menjamin keutuhan dari media penampung dan mudah diterapkan pada media digital. Sehingga media tidak akan rusak atau terjadi perubahan yang signifikan setelah proses penyisipan pesan [9].

2.2.5 *Fast Fourier Transform* (FFT)

Fast Fourier Transform (FFT) adalah teknik komputasi matematis yang digunakan untuk mengubah sinyal analog menjadi sinyal digital berbasis frekuensi. FFT membagi sinyal menjadi frekuensi yang berbeda dengan fungsi eksponensial yang kompleks. FFT dapat digunakan sebagai algoritma untuk menghitung transformasi fourier diskrit dengan cepat dan efisienn [24].

Algoritma FFT melakukan $(N/2) \log_2 N$ perkalian kompleks dalam konsep kerjanya. Sebagai contoh, sinyal dengan 16 titik akan memerlukan 4 tahapan dengan metode FFT. Pada tahap awal sinyal akan dibagi dua dengan masing-

masing memiliki 8 titik. Kemudian setiap bagian dengan 8 titik tersebut akan dibagi kembali menjadi sinyal dengan masing-masing memiliki 4 titik. Pola ini terus berlanjut sampai dengan setiap bagian sinyal hanya terdiri dari 1 titik [25]. Algoritma FFT memecah sampel menjadi dua bagian yaitu bagian genap dan bagian ganjil. Persamaan 2.3 menunjukkan persamaan yang digunakan saat proses pemecahan tersebut berlangsung.

$$X(m) = \sum_{n=0}^{\left(\frac{N}{2}\right)-1} x[2n]e^{-\frac{j2\pi(2nm)}{N}} + e^{-j2\pi m)/N} \sum_{n=0}^{\left(\frac{N}{2}\right)-1} x[2n+1].e^{-\frac{j2\pi(2nm)}{N}} \quad (2.4)$$

Dengan,

N = jumlah sampel input

X(m) = urutan ke-m komponen output FFT (X (0), X (1), …, X (N-1))

m = indeks output FFT dalam domain frekuensi (0,1, …, N-1)

x (2n) = urutan ke-n sampel input genap (x (0), x (2), …, x (N-2))

x (2+1) = urutan ke-n sampel input ganjil (x (1), x (3), …, x (N-1))

n = indeks sampel input dalam domain waktu (0,1, …, N/2-1)

j = konstanta bilangan imajiner (−1)

π = derajat (180°)

e = basis logaritma natural (2.7182818284…)

Setelah melakukan transformasi FFT, suara asli yang diperoleh dari rekaman diubah menjadi sampel suara yang merupakan hasil ekstraksi ciri suara. Data audio yang dimodifikasi dalam format satu dimensi, dan output ekstraksi fitur adalah satu dimensi, yang merupakan dimensi yang sama dengan input audio asli [25].

2.2.6 Pengujian *Black Box*

Pengujian *black box* atau *behavioral testing* merupakan sebuah metode pengujian yang dilakukan untuk mengamati *input* dan *output* sebuah perangkat lunak tanpa harus mengetahui atau memperhatikan struktur kode dari perangkat lunak tersebut. Metode ini bertujuan untuk mengetahui kelayakan fungsi dari sebuah sistem yang dirancang dalam pembuatan perangkat lunak [26].

Terdapat beberapa teknik yang dapat dilakukan saat menerapkan metode *blackbox testing* pada sebuah perangkat lunak. Berikut adalah teknik-teknik yang dapat dilakukan saat melakukan pengujian dengan metode *black box*:

1. *Equivalence Partitioning*

   Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.

2. *Boundary Value Analysis / Limit Testing*

   Mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai *input* dan merupakan komplemen dari *Equivalence Partitioning*.

3. *Comparison Testing*

   Menguji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.

4. *Sample Testing*

   Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.

5. Robustness Testing

   Data input dipilih diluar spesifikasi yang telah didefinisikan. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid

6. *Behavior Testing*

   Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data *stack*.

7. *Performance Testing*

   Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya aliran data, ukuran pemakaian memori, kecepatan eksekusi.

8. *Requirement Testing*

   Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.

9. *Endurance Testing*

   Melibatkan kasus uji yang diulang dengan jumlah tertentu.

10. *Cause-Effect Relationship Testing*

Membagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja. [27]

Pada penelitian ini teknik yang digunakan pada saat proses pengujian adalah teknik *comparison testing* untuk memastikan bahwa hasil output merupakan *file* audio yang disisipi pesan atau audio stego, teknik *performance testing* untuk mengetahui kecepatan eksekusi pada saat menyisipkan pesan atau mengekstrak pesan, dan teknik *endurance testing* untuk memastikan sistem bekerja dengan baik saat digunakan secara berulang-ulang.

2.2.7 *Peak Signal to Noise Ratio* (PSNR)

Peak Signal to Noise Ratio (PSNR) merupakan perbandingan antara nilai maksimum dari sebuah sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. Nilai PSNR dinyatakan dalam satuan desibel (dB). Semakin besar nilai PSNR maka semakin mirip pula dengan *file* asli [28].

Nilai PSNR yang berada di bawah 30dB mengindikasikan kualitas yang relatif rendah. Sedangkan kualitas file stego dapat dikatakan baik atau tinggi apabila memiliki nilai PSNR 40db dan di atasnya. File dengan nilai PSNR yang berada dalam kategori di bawah 30db memiliki kemungkinan lebih tinggi untuk dicurigai memiliki pesan rahasia di dalamnya karena pada file tersebut akan terlihat distorsi atau perbedaan yang cukup signifikan. [29]

Perhitungan nilai PSNR dilakukan berdasarkan nilai *Mean Square Error* (MSE) yang merupakan nilai kesalahan kuadrat rata-rata antara *file* asli dengan *file* hasil steganografi. Persamaan nilai MSE dapat dinyatakan sebagai berikut:

$$MSE = \frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} [l(x,y) - l'(x,y)]^2 \qquad (2.5)$$

dengan

- x dan y merupakan koordinat
- M dan N merupakan dimensi
- $l(x,y)$ nilai piksel *file* asli dan $l'(x,y)$ nilai piksel *file* stego

Kemudian, nilai MSE yang diperoleh, disubstitusikan ke dalam persamaan

sebagai pembagi nilai piksel atau bit maksimal. Persamaan perhitungan nilai PSNR dapat dinyatakan sebagai berikut:

$$PSNR = 10 \log_{10} \left(\frac{b^2}{MSE} \right) \quad (2.6)$$

Keterangan:

- b sebagai nilai piksel maksimal
- MSE adalah nilai *Mean Square Error* (MSE)