**DAFTAR ISI**

[**BAB I PENDAHULUAN** 1](#_Toc33382495)

[**1.1** **Latar Belakang** 1](#_Toc33382496)

[**1.2** **Perumusan Masalah** 3](#_Toc33382497)

[**1.3** **Batasan Masalah** 3](#_Toc33382498)

[**1.4** **Tujuan Penelitian** 3](#_Toc33382499)

[**1.5** **Manfaat Penelitian** 4](#_Toc33382500)

[**1.6** **Metodologi Penelitian dan Pengembangan Sistem** 4](#_Toc33382501)

[**1.6.1** **Metodologi Penelitian** 4](#_Toc33382502)

[**1.6.2** **Metode Pengembangan Sistem** 4](#_Toc33382503)

[**1.7** **Sistematika Penulisan** 7](#_Toc33382504)

[**BAB II TINJAUAN PUSTAKA** 8](#_Toc33382505)

[**2.1** **Citra Digital** 8](#_Toc33382506)

[**2.1.1** **Ruang Warna RGB** 9](#_Toc33382507)

[**2.2** **Steganografi** 11](#_Toc33382508)

[**2.2.1** ***Least Significant Bit*** 11](#_Toc33382509)

[**2.2.2** ***Discrete Cosine Transform*** 12](#_Toc33382510)

[**2.2.3** ***Discrete Haar Wavelet Transform*** 14](#_Toc33382511)

[**2.2.4** **PengujianKualitas MetodeSteganografi** 15](#_Toc33382512)

[**2.3** **Python** 17](#_Toc33382513)

[**2.4** **Tinjauan Literatur** 18](#_Toc33382514)

[**BAB III ANALISIS DAN PERANCANGAN SISTEM** 21](#_Toc33382515)

[**3.1** Komunikasi 21](#_Toc33382516)

[3.1.1 Perumusan Masalah 21](#_Toc33382517)

[3.1.2 Pengumpulan data awal 21](#_Toc33382518)

[3.1.3 Analisis Kebutuhan 21](#_Toc33382519)

[**3.2** Quick Design 21](#_Toc33382520)

[3.2.1 Arsitektur Sistem 21](#_Toc33382521)

[3.2.2 Perancangan Proses 21](#_Toc33382522)

[3.2.3 Perancangan Antarmuka 21](#_Toc33382523)

[**3.3** Pembentukan Prototype 21](#_Toc33382524)

[**3.4** Evaluasi Prototype 21](#_Toc33382525)

[**3.5** Perbaikan Prototype 21](#_Toc33382526)

[**3.6** Produksi Akhir 21](#_Toc33382527)

[**BAB IV ANALISIS DAN PERANCANGAN SISTEM** 22](#_Toc33382528)

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Pentingnya kerahasiaan suatu informasi telah menjadi suatu perhatian tersendiri dari masa ke masa. Manusia berusaha mencari cara bagaimana merahasiakan informasi terhadap pihak yang dianggap tidak berhak untuk mengetahuinya. Berbagai cara telah dilakukan oleh bangsa – bangsa kuno untuk merahasiakan informasi, karena informasi yang jatuh ke orang yang tidak berhak akan menimbulkan kerugian. Sebagai contoh Julius Caesar, kaisar romawi, telah menggunakan suatu metode untuk mengacak pesan sebelum pesan itu dikirimkan kepada penerimanya. Tujuannya adalah agar orang yang tidak berhak menerima pesan tersebut tidak dapat membacanya (Tarigan, 2015).

Zaman sekarang informasi tidak hanya dapat disandikan, tetapi dapat juga disisipkan kedalam media digital. Teknik menyisipkan pesan dikenal dengan nama steganografi. Steganografi sebagai ilmu dan seni untuk menyembunyikan informasi sehingga informasi yang bersifat rahasia tidak dapat diketahui oleh orang lain, kecuali pengirim dan penerima (Atoum, Ibrahim, & M. Ahmad, 2012). Proses steganografi biasanya melibatkan penyandian atau kriptografi. Proses yang dilakukan yaitu dengan enkripsi *plaintext* terlebih dahulu menjadi *Byte cipher* atau pesan rahasia. Kemudian *Byte cipher* disisipkan pada media digital berupa teks, audio, citra atau protocol.

Beberapa penelitian telah dilakukan berkaitan dengan pengamanan data dengan menerapkan beberapa algoritma. Salah satunya adalah algoritma *Least Significant Bit*. Dalam penelitian Danuputri & Kom (2018) menggunakan algoritma *Least Significant Bit* yang diperkuat dengan adanya algoritma *Vignere* *Key* untuk melakukan enkripsi terhadap dokumen berekstensi \*.doc, \*.docx, \*.xls, \*.xlsx, \*.txt, dan \*.pdf untuk dihasilkan plaintext yang kemudian menggunakan *Least Significant Bit* untuk menyisipkannya ke dalam gambar. Penelitian serupa terkait algoritma *Least Significant Bit* juga pernah dilakukan oleh Anselmus (2012) untuk melakukan penyisipan gambar lain yang bersifat identitas untuk melindungi hak cipta dari suatu gambar dengan menggabungkan algoritma *Least Significant Bit* dengan *Redundant Pattern Encoding.* Penelitian lain terkait algoritma *Least Significant Bit* juga pernah dilakukan oleh Devi Rahmadita, dkk (2016) yang menggunakan algoritma tersebut untuk menyembunyikan pesan ke dalam media video sebagai *cover object* dengan hasil uji yang memuaskan. Namun hasil tersebut akan terpengaruh jika *stego object* diserang atau dimanipulasi. Selain itu, terdapat juga penelitian oleh Garno (2017) yang mengkombinasikan metode *Discrete Cosines Transform* danInterpolasi *Bilinear* untuk menyembunyikan pesan. Pada penelitian tersebut Garno berhasil menyisipkan pesan ke dalam citra digital, namun terjadi distorsi yang cukup signifikan pada *stego object* sehingga nilai hasil uji kurang memuaskan.

Penelitian-penelitian tersebut telah menggunakan algoritma-algoritma steganografi yang mempunyai kelebihan masing-masing. Dengan banyaknya metode-metode yang dapat digunakan, maka akan sangat membantu pengguna dalam penyembunyian pesan. Namun, melihat dari kelebihan saja tidaklah cukup. Perlu diketahui bahwa stego object tidak selamanya aman dan tidak menimbulkan kecurigaan. Ketika stego object dikirimkan menggunakan perantara pihak ketiga, ada kemungkinan stego object tersebut rusak atau isinya telah dilihat oleh pihak lain tanpa sepengetahuan pembuat. Hal tersebut terjadi karena setiap metode memiliki tingkat *robustness* (ketahanan) dan *fidelity* (ketidaknampakan perbedaan) yang rendah.

Dari permasalahan yang telah dipaparkan, maka diperlukan perbandingan metode-metode tersebut sehingga dapat diketahui metode mana yang memiliki tingkat *robustness* dan *fidelity* yang baik untuk digunakan dalam steganografi maupun watermarking pada suatu objek. Dalam penelitian ini, metode yang akan diteliti adalah *Least* *Significant* *Bit*, *Discrete* *Cosines* *Transform*, *Discrete* *Haar* *Wavelet* *Transform*, serta gabungan dari algoritma tersebut yang menghasilkan sebuah *modified* *algorithm* yang diharapkan dapat meningkatkan tingkat *robustness* dan *fidelity* dari *stego* *object* yang dibuat.

Oleh karena itu disusunlah penelitian dengan judul “Analisis Faktor *Robustness* dan *Fidelity* pada Metode *Least Significant Bit*, *Discrete Cosines Transform*, dan *Discrete Haar Wavelet Transform* dalam Implementasi Steganografi pada Citra Digital”.

## **Perumusan Masalah**

Berdasarkan latar belakang yang telah dipaparkan, maka dapat diambil rumusan masalah dalam penelitian ini adalah bagaimana perbandingan kualitas *robustness* pada *stego object* setelah dilakukan serangan pada *stego* *object* yang menggunakan metode *Least Significant Bit*, *Discrete Cosine Transform*, dan *Discrete Haar Wavelet Transform* serta bagaimana perbandingan tingkat *fidelity* dari *stego object* setelah disisipkan *hidden object.*

## **Batasan Masalah**

Pada penelitian ini, permasalahan dibatasi sebagai berikut:

1. Citra yang digunakan menggunakan ekstensi \*.jpg, \*.png, dan \*.bmp.
2. *Hidden* *object* atau *embedded* *message* berupa teks dan citra lain.

## **Tujuan Penelitian**

Tujuan pada penelitian ini adalah sebagai berikut:

1. Mengetahui hasil identifikasi uji kualitas *robustness stego object* setelah diimplementasikan metode *Least Significant Bit*, *Discrete Cosine Transform*, dan *Discrete Haar Wavelet Transform* dan manipulasi *stego object.*
2. Mengetahui hasil identifikasi tingkat *fidelity stego object* setelah diimplementasikan metode *Least Significant Bit*, *Discrete Cosine Transform*, dan *Discrete Haar Wavelet Transform* dan menentukan metode manakah yang paling baik dalam menyembunyikan *hidden object*.

## **Manfaat Penelitian**

Manfaat yang dapat diperoleh dari penelitian ini adalah:

* + 1. Dapat mengetahui kelebihan serta kelemahan pada metode *Least Significant Bit*, *Discrete Cosine Transform*, dan *Discrete Haar Wavelet Transform* berdasarkan faktor *robustness* dan *fidelity* pada *stego* *object*.
    2. Dapat menjadi referensi dalam pemilihan metode steganografi dengan mempertimbangkan faktor *robustness* dan *fidelity*.

## **Metodologi Penelitian dan Pengembangan Sistem**

Metodologi penelitian berisi langkah-langkah yang digunakan dalam penelitian ini agarterstruktur dengan baik. Pengembangan system merupakan bagian dari metode penelitian yang digunakan pada penelitian ini.

### **Metodologi Penelitian**

Metode yang digunakan dalam penelitian ini terdapat tiga tahap utama, yaitu tahap pendahuluan, tahap perancangan dan pembuatan prototype sistem, dan tahap penyelesaian. Tahap pendahuluan terdiri dari identifikasi masalah dan tinjauan pustaka. Tahap perancangan (*Quick Design*) dan pembuatan *prototype* sistem dijelaskan terpisah pada sub bab 1.6.2 mengenai metode pengembangan sistem. Tahap penyelesaian berisi penyusunan laporan tugas akhir.

### **Metode Pengembangan Sistem**

Metode yang digunakan dalam pengembangan system ini adalah metode *Prototype* (purwarupa). Tahapan dalam metode *prototype* (Pressman, 2010) dapat dilihat pada gambar 1.1.



**Gambar 1.1.** Metode *Prototype*

Metode *prototype* cocok digunakan untuk mengembangkan sebuah perangkat yang akan dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, dalam hal ini pengguna dari perangkat yang dikembangkan adalah peserta didik. Kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara besar.

*Prototype* bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik.

Berdasarkan model *prototype* yang telah digambarkan diatas, maka dapat diuraikan pembahasan masing-masing tahap dalam model tersebut adalah sebagai berikut:

1. *Communication*/Komunikasi

Perancang perangkat lunak melakukan analisis untuk menentukan kebutuhan perangkat lunak yang saat itu diketahui dan untuk menggambarkan area-area dimana definisi lebih jauh untuk iterasi selanjutnya.

1. *Quick Plan*/Perencanaan Secara Cepat

Dalam perencanaan ini iterasi pembuatan prototipe dilakukan secara cepat. Setelah itu dilakukan pemodelan umum dalam bentuk “rancangan cepat/Quick Design”.

1. *Modeling Quick Design*/Pemodelan Rancangan Cepat

Pada tahap ini dilakukan pemodelan perencanaan ditahap sebelumnya denan menggunakan pemodelan terstuktur dalam bentuk flowchart dan UML untuk menggambarkan analisis dan desain system.

1. *Construction of Prototype*/Pembuatan Prototipe

Dalam pembuatan rancangan cepat berdasarkan pada representasi aspek-aspek perangkat lunak yang akan terlihat oleh end user ( tampilan antarmuka ). Rancangan cepat merupakan dasar untuk memulai konstruksi pembuatan prototipe.

1. *Deployment Delivery and Feedback*/Penyerahan dan Pemberian Umpan Balik

Prototipe kemudian dipresentasikan untuk mengevaluasi prototipe yang telah dibuat sebelumnya dan memberikan umpan balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan. Iterasi akan terjadi saat pengembang melakukan perbaikan prototipe tersebut.

Penelitian ini dilakukan hanya sampai pada tahap Construction of Prototype / Pembuatan Prototype dalam merancang cepat berdasarkan pada representasi aspek- aspek Perangkat Lunak pada *end user*.

## **Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam menyusun laporan penelitian ini adalah sebagai berikut :

**BAB I Pendahuluan**

Pada bagian ini membahas tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

**BAB II Tinjauan Pustaka**

Tinjauan pustaka memuat tentang dasar teori yang digunakan untuk analisis dan perancangan sistem serta implementasi pada penelitian ini. Selain itu juga sebagai bahan referensi dan pondasi untuk memperkuat argumentasi dalam penelitian ini. Teori-teori yang sesuai dengan penelitian ini antara lain steganografi*,* *Least Significant Bit*, *Discrete Cosines Transform ,*dan *Discrete Haar Wavelet Transform*, dan *Peak Signal to Noise Ratio*.

**BAB III Analisis dan Perancangan Sistem**

Pada bagian ini akan membahas mengenai analisa dan perancangan sistem hingga implementasi system yang akan dibangun.

**BAB IV Hasil, Pengujian dan Pembahasan**

Pada bab ini akan menyajikan hasil penelitian yang berisi hasil implementasi dari perancangan yang telah dibuat pada bab sebelumnya. Selain itu berisi pengujian terhadap hasil penelitian beserta pembahasannya.

**BAB V Kesimpulan dan Saran**

Pada bagian ini berisi kesimpulan dari hasil penelitian dan saran yang diajukan oleh penulis untuk pengembangan penelitian selanjutnya.

# **BAB II TINJAUAN PUSTAKA**

## **Citra Digital**

Secara harafiah, citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam (Munir, 2004).

Agar dapat diolah dengan dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi malar (kontinu) menjadi nilai-nilai diskrit disebut digitalisasi. Citra yang dihasilkan inilah yang disebut citra digital (*digital* *image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (atau lebar x panjang). Citra digital yang berukuran NxM lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom sebagai berikut:

|  |  |  |
| --- | --- | --- |
|  | ………………… | ( 2.1 ) |

Indeks baris (i) dan indeks kolom (j) menyatakan suatu koordinat titik pada citra, sedangkan f(i, j) merupakan intensitas (derajat keabuan) pada titik (i, j). Masing-masing elemen pada citra digital (berarti elemen matriks) disebut *image* *element*, *picture* *element* atau *pixel* atau pel. Jadi, citra yang berukuran NxM mempunyai NM buah *pixel*. Sebagai contoh, misalkan sebuah berukuran 256x256 *pixel* dan direpresentasikan secara numerik dengan matriks yang terdiri dari 256 buah baris (di-indeks dari 0 sampai 255) dan 256 buah kolom (di-indeks dari 0 sampai 255) seperti contoh berikut:



**Gambar 2.1.** Contoh tata letak pixel dalam matriks

*Pixel* pertama pada koordinat (0, 0) mempunyai nilai intensitas 0 yang berarti warna *pixel* tersebut hitam, *pixel* kedua pada koordinat (0, 1) mempunyai intensitas 134 yang berarti warnanya antara hitam dan putih, dan seterusnya.

### **Ruang Warna RGB**

Warna pada dasarnya merupakan hasil persepsi dari cahaya dalam spektrum wilayah yang terlihat oleh retina mata, dan memiliki Panjang gelombang antara 400nm sampai dengan 700nm. Ruang warna atau sering disebut model warna merupakan sebuah cara atau metode untuk mengatur, membuat, dan memvisualisasikan warna (Ford & Roberts, 1998).

Salah satu dari ruang warna adalah RGB. RGB sering digunakan didalam sebagian besar aplikasi komputer karena dengan ruang warna ini, tidak diperlukan transformasi untuk menampilkan informasi di layar monitor. Alasan diatas juga menyebabkan RGB banyak dimanfaatkan sebagai ruang warna dasar bagi sebagian besar aplikasi. Model warna RGB adalah model warna berdasarkan konsep penambahan kuat cahaya primer yaitu *Red*, *Green* dan *Blue*. Dalam suatu ruang yang sama sekali tidak ada cahaya, maka ruangan tersebut adalah gelap total. Tidak ada signal gelombang cahaya yang diserap oleh mata kita atau RGB (0,0,0). Apabila ditambahkan cahaya merah pada ruangan tersebut, maka ruangan akan berubah warna menjadi merah misalnya RGB (255,0.0), semua benda dalam ruangan tersebut hanya dapat terlihat berwarna merah. Demikian juga apabila cahaya diganti dengan hijau atau biru.

Apabila diberikan 2 macam cahaya primer dalam ruangan tersebut seperti (merah dan hijau), atau (merah dan biru) atau (hijau dan biru), maka ruangan akan berubah warna masing-masing menjadi kuning, atau magenta, atau cyan. Warna-warna yang dibentuk oleh kombinasi dua macam cahaya tersebut disebut warna sekunder. Warna tersier adalah warna yang hanya dapat terlihat apabila terdapat tiga cahaya primer, jadi apabila dinonaktifkan salah satu cahaya, maka benda tersebut berubah warna. Contoh warna tersier adalah abu-abu, putih.



**Gambar 2.2** Warna RGB

Pada perhitungan dalam program-program komputer model warna direpresentasi dengan nilai komponennya, seperti dalam RGB (rgb) masing-masing nilai antara 0 hingga 255 sesuai dengan urusan masing-masing yaitu pertama *red*, kedua *green* dan ketiga adalah nilai hue dengan demikian masing-masing komponen ada 256 tingkat. Apabila dikombinasikan maka ada 256 x 256 x 256 atau 16777216 kombinasi warna RGB yang dapat dibentuk.

## **Steganografi**

Steganografi berasal dari bahasa Yunani yaitu kata *stegos* yang berarti sembunyi dan *graphia* yang berarti tulisan. Steganografi secara umum memiliki arti ilmu dan seni menyembunyikan suatu fakta untuk berkomunikasi. Dengan menggunakan steganografi, pesan rahasia dapat disisipkan ke dalam sebuah informasi yang tidak mencurigakan dan mengirimkannya tanpa ada yang mengetahui keberadaan dari pesan rahasia tersebut (Krenn, 2004). Dalam era digital ini, steganografi berarti penyisipan pesan dalam bentuk digital ke dalam media digital yang ada. Untuk steganografi terbentuk dari dua macam yaitu pesan digital atau sering disebut dengan *message* yang akan disisipkan dan media tempat penyisipan akan dilakukan. Media tempat penyisipan dapat berupa teks, gambar, suara, dan video. Untuk media gambar, gambar yang dijadikan sebagai citra penampung disebut dengan *cover* *image*. Penyembunyian pesan yang berupa teks maupun gambar ke dalam citra digital akan mempengaruhi kualitas citra tersebut.

### ***Least Significant Bit***

Bit atau *binary digit* adalah unit dasar penyimpanan data di dalam komputer, nilai bit suatu data adalah 0 (nol) atau 1 (satu). Semua data yang ada pada komputer disimpan ke dalam satuan bit ini, termasuk gambar, suara, ataupun video hanya saja penerjemahan representasi bit pada masing-masing media yang tentunya akan berbeda. Seperti penjelasan pada bagian sebelumnya bahwa format pewarnaan pada media gambar juga menggunakan satuan bit dalam penyimpanannya. Sebagai contoh pewarnaan *monochrome* menggunakan 1 bit untuk merepresentasikan warna hitam atau putih, pewarnaan *grayscale* menggunakan 8 bit untuk merepresentasikan tingkat keabuan dan pada pewarnaan RGB menggunakan 24 bit (8 bit untuk *Red*, 8 bit untuk *Green*, 8 bit untuk *Blue*).

*Least Significant Bit* (LSB) adalah bagian dari barisan data biner yang mempunyai nilai paling tidak berarti atau paling kecil. Bit LSB letaknya di paling kanan pada barisan biner, karena nilai 1 bit LSB pada barisan biner hanya merepresentasikan nilai 1 desimal, maka bit ini dianggap tidak berarti. Sehingga jika terjadi perubahan pada nilai bit LSB maka tidak akan terjadi perubahan secara signifikan.



**Gambar 2.3.** Ilustrasi penyisipan bit LSB

*Modified Least Significant Bit* (MLSB) atau modifikasi dari Algoritma LSB digunakan untuk meng-*encode* sebuah identitas ke dalam citra asli. MLSB menggunakan manipulasi beberapa bit-bit penyisip sebelum meng-encode pesan tersebut (Zaher, 2011).

Modifikasi pesan dengan algoritma MLSB dimana bit pesan yang seharusnya 1 karakter memiliki nilai 8 bit ASCII akan dimodifikasi menjadi 5 bit (31 desimal). Pada algoritma ini karakter dan angka direpresentasikan dalam 5 bit sebelum disisipkan ke dalam citra asli dengan teknik LSB.

### ***Discrete Cosine Transform***

*Discrete Cosine Transform* (DCT) Transformasi DCT merupakan salah satu *transform* *coding* yang akan merubah *byte* dari domain spasial menjadi domain frekuensi dan memisahkan *byte* data tersebut menjadi dua bagian, yaitu frekuensi tinggi (koefisien DC) dan frekuensi rendah (koefisien AC). Pada DCT, koefisien DC digunakan sebagai tempat penyisipan pesan. Hal ini dikarenakan koefisien DC memiliki kapasitas persepsi yang lebih tinggi dari pada koefisien AC sehingga proses penyisipan tidak akan mengubah kualitas gambar secara visual. Selain itu, sinyal proses dan distorsi gambar memiliki pengaruh yang lebih rendah terhadap koefisien DC daripada koefisien AC (Reva, Susilo, & Purwandari, 2016).

Transformasi citra dilakukan dengan menggunakan DCT (*Discrete Cosine Transform*), sehingga dapat dikatakan bahwa penyisipan dilakukan pada ranah DCT. Penyisipan dilakukan terhadap citra bitmap dengan kedalaman warna 24 bit.DCT digunakan untuk metransformasikan nilai intensitas blok 8x8 pikselnya yangberurutan dari image menjadi 64 koefisien DCT kedalam frekunesi dasarnya, diubahkoefisien- koefisiennya dan kemudian ditransformasikan kembali dengan IDCT (*Inverse Discrete Cosine Transform* ). Setiap basis matriks dikarakteristikan olehfrekuensi spatial horizontal dan vertical. Dalam konteks citra, hal ini menunjukkan tingkat signifikasi secara perseptual, artinya basis fungsi dengan frekuensi rendah memiliki sumbangan yang lebih besar bagi perubahan penampakan citra dibandingkan basis fungsi yang memiliki frekuensi tinggi. Nilai konstanta basis fungsi yang terletak di bagian kiri atas sering disebut basis fungsi DC, dan DCT koefisien yang bersesuaian dengannya disebut koefisien DC (DC coeficient). Masukan proses DCT berupa matrik NxN. Persamaan DCT untuk matrik berukuran NxN dapat dituliskan sebagai berikut :

|  |  |  |
| --- | --- | --- |
|  | …….. | ( 2.2 ) |

Dimana ;

S(u,v) = Data pada domain frekuensi

S(x,y) = Data pada domain ruang

Output dari fungsi DCT adalah nilai komponen frekuensi tertentu dan output darifungsi ini ditentukan oleh dua parameter, yaitu u dan v. Cara menentukan mana yang frekuensi rendah dan mana yang frekuensi tinggi adalah dengan menjumlahkan nilai u dan v. Jadi jika u+v makin tinggi berarti S(u,v) menyatakan komponen frekuensi yang makin tinggi. Input dan Output dari fungsi DCT juga merupakan suatu matriks dengan ukuran NxN. P(x,y) adalah nilai pixel pada koordinat (x,y), index dimulai dari 0.

### ***Discrete Haar Wavelet Transform***

Secara umum Transformasi Wavelet Diskrit merupakan dekomposisi citra pada frekuensi subband citra tersebut. Komponen subband transformasi wavelet dihasilkan dengan cara penurunan level dekomposisi. Implementasi Transformasi Wavelet Diskrit dapat dilakukan dengan melewatkan sinyal melalui sebuah *lowpass* *filter* dan *highpass* filter dan melakukan *downsampling* pada keluaran masing-masing filter. *Highpass* filter digunakan untuk menganalisis frekuensi tinggi dan *lowpass* *filter* digunakan untuk menganalisis frekuensi rendah. Proses dekomposisi dapat melalui satu atau lebih tingkatan. Dekomposisi satu tingkat dapat ditulis dengan persamaan sebagai berikut:

|  |  |
| --- | --- |
| …...……………………………………………… | ( 2.3 ) |
| ..……………………………………………..... | ( 2.4 ) |

Variabel ytinggi[k] dan yrendah[k] adalah hasil dari *highpass* *filter* dan *lowpass* filter, merupakan sinyal asal, adalah *highpass* *filter*, dan adalah *lowpass* *filter*.



**Gambar 2.4.** Dekomposisi Wavelet tiga tingkat

Dengan menggunakan koefisien Transformasi Wavelet Diskrit ini maka dapat dilakukan proses *Inverse* *Discrete* *Wavelet* *Transform* (IDWT) untuk merekonstruksi menjadi sinyal asal.

|  |  |
| --- | --- |
| ……….. | ( 2.5 ) |

Proses rekonstruksi merupakan kebalikan dari proses dekomposisi sesuai dengan tingkatan pada proses dekomposisi. Transformasi wavelet diskrit dimensi dua dapat digambarkan seperti gambar 2.3.



**Gambar 2.5.** DWT dua dimensi

### **PengujianKualitas MetodeSteganografi**

Terdapat kriteria-kriteria yang harus diperhatikan dalam penyembunyian data yaitu:

1. *Impercebility* yaitu keberadaan pesan rahasia tidak dapat dipersepsi secara indrawi.

2. *Fidelity* yaitu mutu citra penampung tidak jauh berubah

3. *Robustness* yaitu data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penampung (seperti pengubahan kontras, penajaman, pemampatan, rotasi, perbesaran gambar, pemotongan, enkripsi, dan sebagainya)

4. *Recovery* yaitu pesan dapat diekstrak kembali.

#### **Uji Tingkat Robustness**

*Robustness* merupakan salah satu kriteria dalam steganografi yang kurang penting dalam steganografi dikarenakan selama objek stego tidak terdapat perbedaan mencolok dan tidak menimbulkan kecurigaan. Namun kriteria ini tetap dapat dibilang penting apabila menyangkut objek tersembunyi di dalamnya. *Robustness* tidak kalah penting dikarenakan jika terdapat serangan pada objek stego, *hidden object* harus tetap tidak rusak dan dapat memenuhi kriteria *Recovery*.

Dalam pengujiannya, kriteria *robustness* dapat dilakukan dengan beragam cara yang juga disebut sebagai *steganalysis*. Salah satu metode yang dapat diterapkan adalah metode serangan StirMark. StirMark merupakan metode steganalysis yang menggabungkan berbagai teknik dasar dalam manipulasi citra seperti rotasi, pemotongan, *resampling*, *resizing*, dan kompresi pada citra watermark. Pada umumnya skema watermarking berhasil lolos dari manipulasi-manipulasi tersebut dan pada umumnya pula, skema watermark tidak dapat bertahan dari gabungan atau kombinasi dari manipulasi-manipulasi dasar tersebut. Hal ini menjadi dasar pembangunan dari metode StirMark. (Ferdian, 2006)

Dapat dikatakan dalam versi yang paling sederhana, StirMark membangkitkan sebuah proses *resampling*. Proses *resampling* ini dilakukan secara digital. Proses ini dapat diumpamakan dengan cara mencetak sebuah citra ke kertas dan melakukan scanning terhadap citra tersebut, dalam hal ini scanner yang digunakan diasumsikan berkualitas sangat tinggi, sehingga hampir tidak ada perubahan jika hasil resampling dibandingkan dengan sample aslinya.

#### **Uji Tingkat Fidelity**

*Fidelity* merupakan kriteria paling umum yang sangat penting dalam steganografi. Kriteria *fidelity* menunjukan besaran mutu stego object tidak jauh berubah dari sebelum dimasukkan *hidden* *object* maupun sesudahnya. Hal ini berkaitan tentang kecurigaan pihak lain jika melihat *stego* *object* dan demi keamanan objek yang disembunyikan maka kriteria *fidelity* perlu diperhatikan. Jika objek stego mempunyai nilai *fidelity* yang sangat rendah, kriteria lain seperti *impercebility* juga akan terpengaruh dikarenakan distorsi yang ada tentunya akan disadari oleh inderawi.

Pengujian fidelity umumnya menggunakan penghitungan PSNR dan MSE. *Peak Signal to Noise Ratio* (PSNR) merupakan sebuah parameter yang penting untuk mengukur kualitas proses pengolahan citra. PSNR adalah rasio antara intensitas maksimum citra dengan *Mean Square Error* (MSE) dari citra. Persamaan untuk menghitung nilai PSNR adalah sebgai berikut:

|  |  |
| --- | --- |
| …………………………………………………….... | ( 2.6 ) |

Dengan persamaan MSE adalah sebagai berikut:

|  |  |
| --- | --- |
| ………………………………………... | ( 2.7 ) |

Dalam penghitungan kualitas dua buah citra makin besar nilai PSNR maka makin kecil nilai MSE. Nilai MSE yang kecil mengindikasikan dua buah citra mempunyai sedikit perbedaan. Dari pengalaman empiris, citra dengan PSNR≥30 masih dapat dianggap berkualitas bagus, tetpi jika PSNR < 30 dikatakan kualitas citra terdegradasi secara signifikan (Munir, 2019).

## **Python**

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar.

Python mendukung multi paradigma pemrograman, utamanya pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

Python dikembangkan oleh Guido van Rossum pada tahun 1990 di *Stichting Mathematisch Centrum* (CWI), Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC. Versi terakhir yang dikeluarkan CWI adalah 1.2. (Venners, 2003)

Tahun 1995, Guido pindah ke CNRI di Virginia Amerika sambil terus melanjutkan pengembangan Python. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, Guido dan para pengembang inti Python pindah ke BeOpen.com yang merupakan sebuah perusahaan komersial dan membentuk BeOpen PythonLabs. Python 2.0 dikeluarkan oleh BeOpen. Setelah mengeluarkan Python 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke *Digital Creations*.

Saat ini pengembangan Python terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan *Python Software Foundation*. *Python Software Foundation* adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual Python sejak versi 2.1 dan dengan demikian mencegah Python dimiliki oleh perusahaan komersial. Saat ini distribusi Python sudah mencapai versi 2.7.14 dan versi 3.6.3

## **Tinjauan Literatur**

Adapun beberapa penelitian yang telah dilakukan dan dijadikan referensi dalam penelitian berikut ini:

**Tabel 2.1.** Tinjauan Literatur

| **No** | **Penulis** | **Judul** | **Metode** | **Hasil** |
| --- | --- | --- | --- | --- |
| 1 | Anselmus Krisma Adi Kurniawan. 2012 | *Digital Watermarking* pada Gambar Digital dengan Metode *Redundant Pattern Encoding* | *Standart Least Significant Bit* dan *Redundant Pattern Encoding* | Citra *watermark* berhasil diimplementasikan dan tahan terhadap serangan transformasi *cropping*. |
| 2 | Devi Rahmadita, Dr. Ir. Bambang Hidayat, DEA dan I Nyoman Apraz Ramatryana, S.T., M.T. 2016 | Steganografi pada *Frame* *Video* *Stationer* menggunakan Metode *Least Signification Bit* | *Standart Least Significant Bit* | Penghitungan kualitas dengan metode PSNR mendapatkan hasil yang maksimal dan tetap memenuhi syarat dianggap bagus sebagai *stego* *object*. Namun nilai PSNR akan menurun ketika mendapat serangan dengan metode *Salt and Pepper Noise.* |

**Tabel 2.2.** Lanjutan Tinjauan Literatur

| 3 | Thomas Edyson Tarigan.  2015 | Algoritma MEoF (Modifikasi *End of File*) untuk Steganografi pada Citra Bitmap 24 Bit | *Modified End of File* | Algoritma Modifikasi *End of File* berhasil memperbaiki kelemahan algoritma *End of File*, yaitu dapat mengatasi peningkatan ukuran *file* dan perubahan kualitas citra *stego* dari citra asli |
| --- | --- | --- | --- | --- |
| 4 | Garno dan Arip Solehudin.  2017 | Teknik Steganografi dengan metode *Discrete Cosines Transform* (DCT) pada Citra Interpolasi *Bilinear* untuk Pengamanan Pesan | *Discrete Cosines Transform* danInterpolasi *Bilinear* | Dari penelitian yang telah dilakukan menghasilkan *stego* *object* dengan rata-rata belum mencapai nilai 40 db sehingga terbilang kurang bagus namun memiliki nilai MSE yang relatif kecil. |
| 5 | Chyquita Danuputri, M. Kom. 2018 | Pengamanan Data melalui *Cloud Computing* dengan Integrasi Steganografi *Least Significant Bit* dan Kriptografi *Vigenere Key* Berbasis *Android* | *Standart Least Significant Bit* dan enkripsi *Vigenere Key* | Pesan berhasil di sembunyikan pada citra digital dan diimplementasikan pada *cloud drive*. |

Penelitian ini akan menganalisis metode LSB, DCT, dan DWT dalam steganografi ditinjau dari faktor *robustness* dan *fidelity*. Hal yang membedakan dari penelitian sebelumnya adalah objek yang diteliti yang kebanyakan lebih fokus pada performa algoritma berdasarkan waktu eksekusi dan perbandingan visual hasil akhir.

# **BAB III ANALISIS DAN PERANCANGAN SISTEM**

## Komunikasi

### Perumusan Masalah

### Pengumpulan data awal

### Analisis Kebutuhan

## Quick Design

### Arsitektur Sistem

### Perancangan Proses

### Perancangan Antarmuka

## Pembentukan Prototype

## Evaluasi Prototype

## Perbaikan Prototype

## Produksi Akhir

# **BAB IV ANALISIS DAN PERANCANGAN SISTEM**