ISSN : 2407-1730

VOL 7. NO.1, JANUARI-JUNI 2021

# INFORMANIKA

## JURNAL MANAJAMEN INFORMATIKA


POLITEKNIK ANIKA
Jln.Kol. H. Burlian Km.7 Palembang
Website : journal.politekanika.ac.id
E-Mail : jurnalinformanika@gmail.com

## ARTICLES

# KEAMANAN DOKUMEN DIGITAL MENGGUNAKAN METODE *STEGANOGRAFI LSB* DAN *DES*

**Kurniati**
Teknik Informatika, Fakultas Ilmu Komputer
Universitas Bina Darma
Email: kurniati@binadarma.ac.id

## ABSTRAK

Perkembangan teknologi informasi yang sangat pesat, mengubah pola pikir manusia untuk beranjak dari dokumen yang awalnya dalam bentuk lembaran-lembaran kertas beralih menjadi dokumen digital berbentuk file *pdf. Namun, dengan menjadikan dokumen ke dalam bentuk digital sangat rentan terhadap kejahatan *cybercrime*. Oleh karena itu, sangat memerlukan pengamanan data akan isi dan privasinya. Steganografi merupakan seni untuk menyisipkan pesan gambar *.jpeg tanda tangan rahasia ke dalam data yang kita anggap penting dengan menggunakan metode LSB yang diyakini mudah dalam implementasinya. Data yang disisipkan tadi perlu kita tambah pengamanannya dengan menerapkan Algoritma DES yang mana merupakan bagian dari kriptografi sebuah seni untuk mengacak data dalam bentuk *cipher* saat proses *embedding stego* dan mengembalikannya kembali dengan menjalankan proses *extracting stego*. Hasil dari perbandingan kapasitas dari file *.pdf sebelum dan sesudah disisipi pesan gambar *.jpeg dengan menghasilkan kapasitas yang sama dan tidak mengalami perubahan kapasitas dalam ukuran file.

**Kata Kunci***:* LSB, Kriptografi, DES, Steganografi , Java.

## I. PENDAHULUAN

Dampak dari perkembangan teknologi informasi saat ini mampu mengubah pola pikir manusia terhadap pemanfaatan teknologi tersebut guna mempermudah aktivitas kerja manusia agar lebih efisien dari sisi waktu, tempat dan biaya. Terutama dalam pengolahan data berupa dokumen yang saat ini manusia dengan mudahnya dapat mengakses informasi apapun tanpa batas. Pengiriman data yang begitu mudah dengan menggunakan beberapa media pengiriman dapat mengakibatkan ancaman keamanan ataupun privasi bagi pengguna pada saat melakukan pengiriman data. Dokumen pada suatu instansi pemerintah maupun swasta merupakan salah satu data penting yang harus dijaga aspek digitalnya dari pihak yang tidak bertanggung jawab.

PalComTech merupakan salah satu instansi yang bergerak dalam bidang pendidikan dimana banyak dokumen penting yang harus dijaga. Salah satu dokumen yang harus dijaga adalah dokumen kuesioner layanan dosen terhadap mahasiswa, dimana dokumen kuesioner layanan adalah hasil dari kinerja dosen selama mengajar permata kuliah. Dengan melakukan pengisian kuesioner secara *online* mahasiswa dapat memberikan penilaian berkenaan dengan kinerja dosen yang bersangkutan sehingga, dari hasil kuesioner yang didapat akan

mempermudah pihak instansi untuk melihat kinerja dosen secara keseluruhan. Untuk mencegah terjadinya kebocoran data pencurian data serta pemalsuan data yang dilakukan oleh pihak yang tidak bertanggung jawab. Maka, perlu dilakukan pengamanan menggunakan aplikasi yang dapat melalukan penanaman pesan gambar pada dokumen penting tersebut. Tidak hanya pada dokumen hasil kuesioner saja, namun semua file yang berformat *.pdf dapat juga diamankan dengan aplikasi tersebut. Dengan menggunakan metode penyembunyian pesan atau sering disebut *steganografi*.

## II. TINJAUAN PUSTAKA

### *2.1 Steganografi*

Dalam Bahasa Yunani *Steganografi* berasal dari dua kata yaitu *stegos* bermakna penyamaran dan *graphia* yang memiliki arti tulisan. *Steganografi* dapat dimanfaatkan guna melakukan proses penyembunyian informasi rahasia agar tidak diketahui orang pihak lain pada sebuah media. [1] Dengan menyamarkan pesan tersebut guna menghilangkan kecurigaan terhadap sebuah data merupakann tujuan dari dilakukannya *steganografi*. [2]

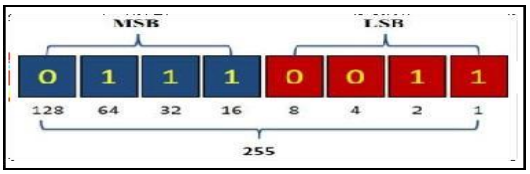
Data rahasia yang disembunyikan ke dalam citra *digital* dapat membuat kualitas citra berubah. [3] Penyembunyian data tharus memperhatikan beberapa kriteria diantaranya:

1. *Fidelity*. Mutu dari sebuah citra-*cover* tidak mengalami perubahan yang signifikan dan *file* masih dalam kondisi baik.
2. *Robustness*. Data harus dapat melalui berbagai operasi manipulasi citra-*stego*, seperti proses mengubah kontras, penajaman, *rotation*, pemampatan, *cutting*, dan sebagainya.
3. *Recovery*. Data yang telah mengalami proses penyembunyian harus mampu diekstrak kembali. Karena tujuan *steganography* merupakan data-*hiding*, maka sewaktu-waktu data rahasia didalam citra- *stego* harus dapat diambil kembali untuk digunakan lebih lanjut.

### *2.2 Least Significant Bit (LSB)*

*Least Significant Bit* (*LSB*) adalah pilihan yang dapat dipilih untuk melakukan proses penyembunyian pesan. Dimana dengan cara melakukan modifikasi *bit- bit* de dalam setiap *byte* warna dalam bentuk *pixel*. [4] *Bit-bit* akan melalui proses modifikasi dengan lakukan pergantian setiap *LSB* dengan *bit-bit* pesan rahasia yang akan melalui proses penyembunyian citra digital.

Syarat yang dibutuhkan dalam metode ini yaitu, ketika akan melakukan kompresi pada *filestego*, dengan format *lossless compression*. Hal itu dikarenakan metode ini menggunakan *bit-bit* pada setiap *pixel* pada *image*. Jika digunakan format *lossy compression*, pesan rahasia yang disembunyikan dapat hilang. Contoh penggunaan *LSB*, sebuah susunan *bit* pada sebuah *byte*:




Gambar 1. Representasi Biner

Berdasarkan gambar di atas menyataakan bahwa:

*(MSB = Most Significant Bit, LSB = Least Significant Bit)*

*Bit* yang dianggap sesuai untuk dilakukannya proses penukaran adalah *bit LSB* yang mana perubahan tersebut akan menjadikan nilai *byte* lebih tinggi 1 angka atau lebih rendah 1 angka dari nilai sebelumnya.

Jika *cover* menggunakan *image* 8*bit color*, hanya akan 1*bit* saja dari setiap *pixel* warna yang dapat dimodifikasi sehingga pemilihan *image* harus dilakukan dengan sangat hati-hati dan *grayscale* pilihan warna yang tepat dikarekan akan membuat mata manusia kesulitan pendeteksi data.

Proses ekstraksi pesan dapat dengan mudah dilakukan dengan mengekstrak *LSB* dari masing- masing *pixel* pada *stegofile* secara berurutan dan menuliskannya ke *output file* yang akan berisi pesan tersebut.

### *2.3 Algoritma Data Encryption Standard (DES)*

*DES* termasuk ke dalam sistem *kriptografi* simetri dan tergolong jenis *cipher* blok. *DES* beroperasi pada ukuran blok 64 *bit*. *DES* mengenkripsikan 64 *bit plainteks* menjadi 64 *bit cipherteks* dengan menggunakan 56 *bit* kunci internal (*internalkey*) atau sub-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 *bit*. Algoritma DES (*Data Encryption Standard*) merupakan sebuah sistem kriptografi yang simetri dan termasuk ke dalam golongan jenis blok ataupun kode. [5]

Menurut Munir [3] algoritma enkripsi *DES* terdapat beberapa proses enkripsi yaitu:

1. Permutasi Awal, Blok *plainteks* dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP). Hal ini bertujuan untuk melakukan pengacakan *plainteks* agar urutan *bit-bit* di dalamnya mengalami pwberubahan.
2. embangkitan Kunci Internal, terdapat 16 putaran, sehingga membutuhkan 16 kunci internal, yaitu K1, K2, …, K16. Kunci-kunci internal dapat dibangkitkan sebelum melalui proses enkripsi ataupun bersamaan dengan dilakukannya proses enkripsi. Kunci eksternal yang diberikan *user* memiliki panjang 64 *bit* atau 8 karakter.
3. *Enciphering*, hasil permutasi awal kemudian di- *enciphering* sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.

### *2.4 Netbeans 8.2*

*NetBeans*8.2 adalah versi stabil terbaru dari *IDENetbeans*, yang dirilis oleh *Sun Microsystems*. *IDENetbeans* berlisensikan *Sun Public License*, *Netbeans* bersifat *open-source*. Siapapun yang berminat dapat mengambil kode sumbernya. Proyek *NetBeans* berbasiskan komunitas. Jika berkualifikasi, dapat juga ikut terlibat di dalam pembuatan *NetBeans* atau modul- modulnya.

*NetBeans* sebagai *IDE* ditujukan untuk memudahkan pemrograman *Java*. Berpindah dari pemrograman manual *Java*, yang memakai *editor* teks, dari *command-prompt*. Dalam *Netbeans*, pemrograman dilakukan berbasiskan *visual*dan *event-driven*. Persis seperti *IDE* lain, misal *Borland Delphi* dan *Microsoft Visual Studio*.*NetBeans* mencakup *compiler* atau *builder*, dan *debuggerinternal*. [6] Hal ini sangat memudahkan proses paska perancangan program.

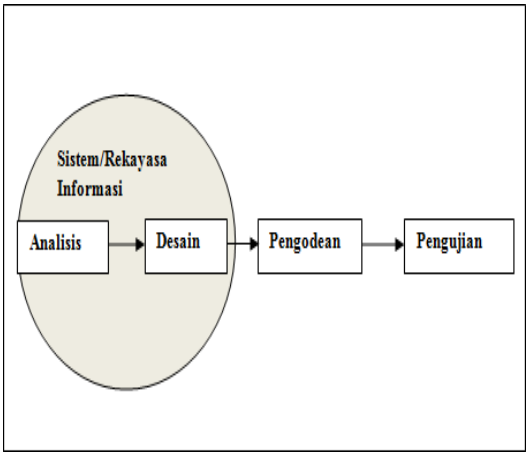
## III. METODOLOGI PENELITIAN

Metode penelitian yang penulis gunakan adalah metode ekperimental. Metode dengan proses manipulasi terhadap objek secara terkontrol untuk

proses perbandingan. [7] Penelitian ekperimental dilakukan dengan menempuh langkah-langkah seperti berikut:

1) Melakukan pengkajian dengan cara induktif terkait dengan masalah yang akan dipecahkan.
2) Mengidentifikasi dan mendefinisikan masalah.
3) Melakukan studi literatur dan beberapa sumber yang relevan, memformulasikan hipotesis penelitian, menentukan variabel, dan merumuskan definisi operasional dan definisi istilah.
4) Membuat rencana penelitian yang didalamnya mencakup kegiatan.

Dalam teori rekayasa perangkat lunak terdapat beberapa macam model pengembangan perangkat lunak. Model pengembangan yang digunakan dalam penelitian ini adalah model SDLC air terjun (*waterfall)* sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model *waterfall* menerapkan pendekatan alur hidup perangkat lunak secara sekuensial terurut dimulai dari *analisys*, desain, *coding*, pengujian, dan *support*.[8] Berikut adalah gambar model air terjun (*waterfall*):



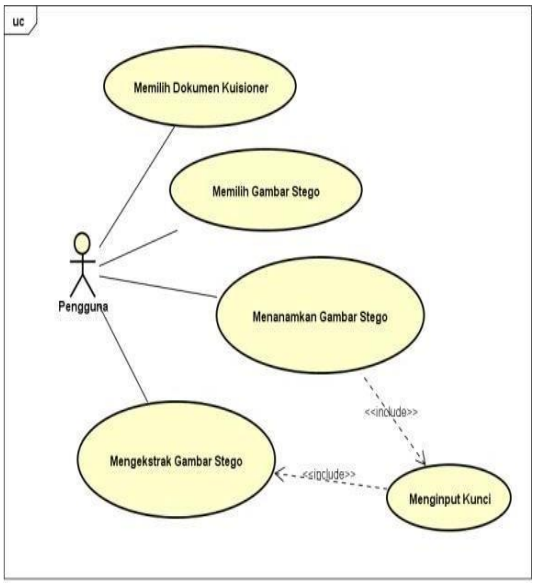

Gambar 2. *Waterfall*

## IV. HASIL DAN PEMBAHASAN

### 4.1 Impelementasi

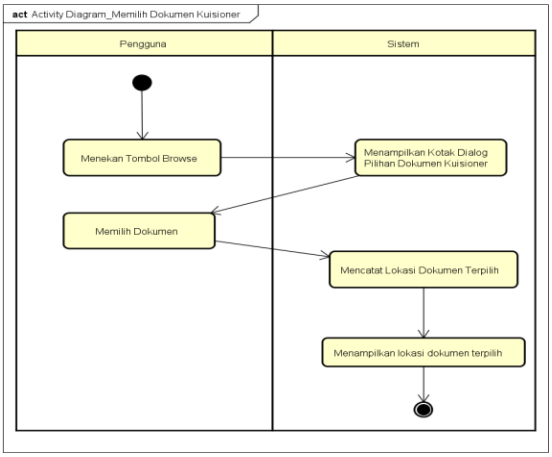
Tahapan implementasi di dalam metode pengembangan *waterfall* merupakan tahap di mana dilakukan realisasi terhadap desain yang telah diciptakan pada tahapan sebelumnya. Mulai dari menentukan lingkungan pengembangan, mempelajari cara kerja algoritma, menentukan aktor dan merancang *usecase*, memodelkan *activity* diagram dan *class* diagram serta merancang antar muka pengguna.

*Use Case* merupakan sebuah model yang akan dilakukan sistem guna mendeskripsikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. [8] *Use case* dari aplikasi ini dapat dilihat pada gambar 4.



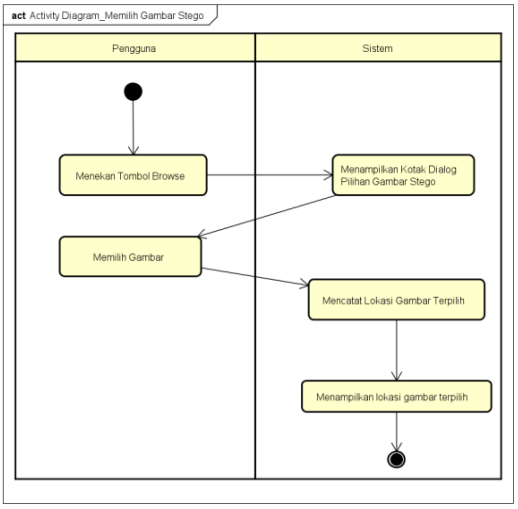

Gambar 5. Rancangan Diagram *Usecase*

Dalam pemilihan dokumen kuesioner, pengguna cukup menekan tombol *browse* yang kemudian akan diikuti dengan tampilnya kotak dialog pemilihan *file.* Sehingga, sistem akan mengingat lokasi *file* yang dipilih seperti yang dijelaskan pada gambar 6.



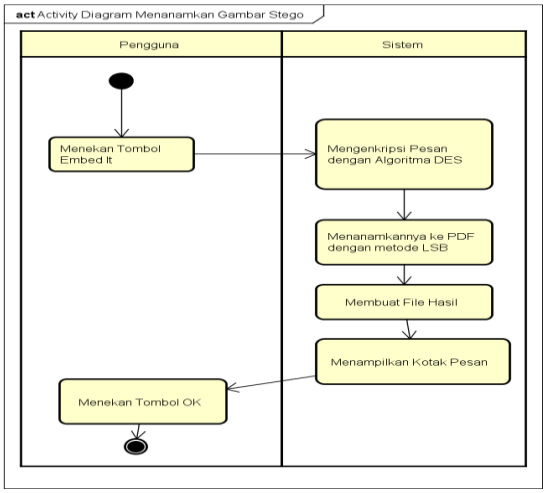

Gambar 6. Rancangan *Activity Diagram* Memilih Dokumen Kuesioner

Dalam pemilihan dokumen gambar *stego*, pengguna cukup menekan tombol *browse* yang kemudian akan diikuti dengan tampilnya kotak dialog pemilihan *file*. Sistem akan mengingat lokasi *filestego* yang dipilih dan kemudian lokasi tersebut ditampilkan ke layar. Aktivitas-aktivitas itu digambarkan pada gambar 7.



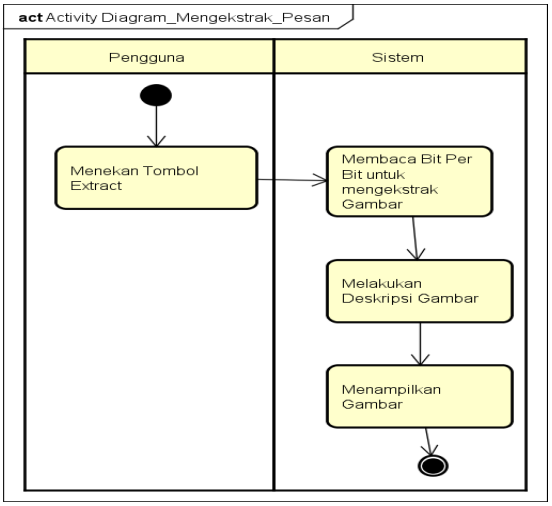

Gambar 7. Rancangan *Activity Diagram* Memilih Gambar *Stego*

Penanaman gambar *stego* cukup dengan menekan tombol yang kemudian akan diikuti dengan serangkaian proses oleh sistem. Sistem akan menerapkan algoritma *DES* dan metode *LSB* dan diakhiri dengan membuat *file* hasil. Aktivitas-aktivitas itu digambarkan pada gambar 8.




Gambar 8. Rancangan *Activity Diagram* Menanamkan Gambar *Stego*

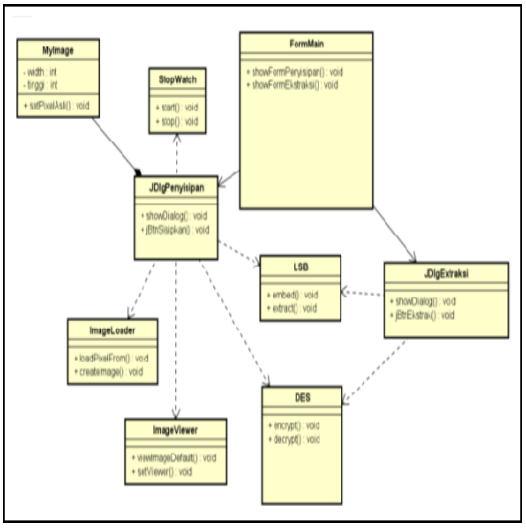
Pengguna dapat mengekstrak gambar *stego* hanya dengan menekan tombol yang mana akan diikuti oleh sub proses yang dilakukan oleh sistem seperti membaca *bit-bit* pada bagian yang kurang *significant* untuk disusun kembali datanya. Aktivitas-aktivitas itu digambarkan pada gambar 9.




Gambar 9. Rancangan *Activity Diagram* Mengekstrak Gambar *Stego*

Sedangkan pada diagram kelas terdapat tiga kelas *interface* yaitu *Form Main, JDlg Extraksi* dan *JDlg* Penyisipan. Selebihnya terdapat kelas

lain yang memiliki peran tertentu. Hal ini digambarkan pada gambar 10.




Gambar 10. Rancangan *Class Diagram*

## 4.2 Hasil

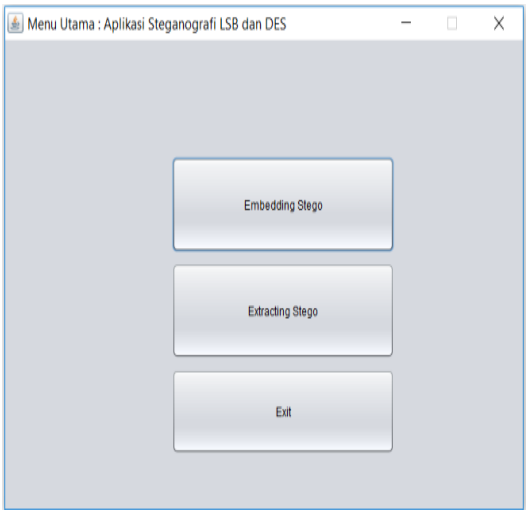
Dari hasil analisis penelitian yang ada di objek, bukan hanya dokumen kuesioner saja yang dapat diamankan, namun sumua dokumen yang berformat *.pdf dapat juga diamankan menggunakan aplikasi ini melaui penyimpanan lewat *email* dengan dokumen yang telah terenkripsi agar terhindar dari heacking dan kebocoran dokumen bisa di cegah tersebut. Cara kerja dari aplikasi ini adalah dimulai dengan menjalankan menu utama. Setelah itu dapat dilanjutkan dengan menggunakan fungsi penyisipan pesan gambar dan ekstraksi pesan gambar. Penyisipan pesan gambar dimulai dengan proses memilih input berupa dokumen kuesioner maupun dokumen lainnya diikuti dengan *stegoimage* dan kemudian penyisipan *stegoimage* dimulai. Penyisipan pesan berjalan dengan baik dengan metode *LSB*, namun sebelum disisipkan, *stegoimage* terlebih dahulu dilakukan enkripsi dengan algoritma *DES* dengan memberi pengamanan terhadap pesan.

Pesan yang sudah disisipkan, dapat dikeluarkan kembali atau dapat diektraksi dengan cara membaca kembali *bit-bit* yang posisinya pada bagian *least significant* dan disusun menjadi *matrix pixel* yang kemudian dilanjutkan dengan proses dekripsi dengan algoritma *DES*. Selanjutnya akan ditampilkan ke layar kembali *stegoimage* yang berhasil diekstraksi tadi.

## 4.3 Pembahasan

Aplikasi berjalan dilingkungan *desktop* dengan bahasa pemrograman Java di *platform Windows*. Berikut adalah tampilan bagaimana aplikasi berhasil di jalankan.

Setelah proses analisa dilakukan maka lanjut ke tahap implementasi sistem dengan melakukan perancangan *interface* sesui kebutuhan. Penjelasan terhadap hasil implementasi sistem ini, berupa *interface* proses ketika terjadinya *embedding* dan *extracting* data pada dokumen digital.




Gambar 11. Tampilan Menu Utama

Dari gambar 11 pada menu utama aplikasi terdapat tiga proses yang dapat dilakukan oleh pengguna yaitu *embedding stego, extracting stego dan exit.* Sedangkan tampilan hasil dari implementasi steganografi difambarkan pada gambar 12.

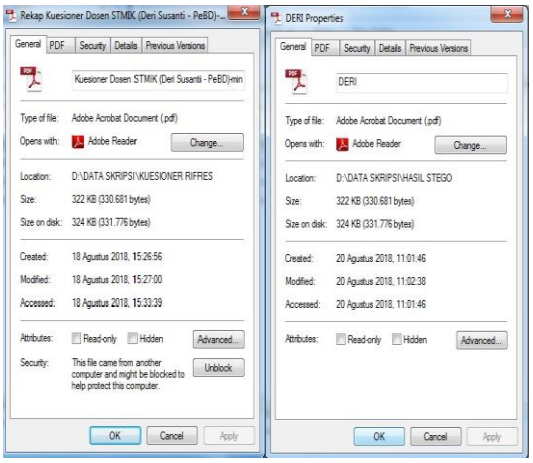

Gambar 12. Hasil Implementasi Penyisipan Gambar

Jika proses penyisipan berhasil disimpan dan ekstrak data berhasil dilakukan maka pada layar akan muncul tampilan seperti gambar 13 dan 14.


Gambar 13. Hasil Implementasi Dokumen Berhasil Disimpan


Gambar 14. Hasil Implementasi *extracting*

Tahap terakhir yang dilakukan pada aplikasi ini berupa pengujian terhadap sistem pada aspek kapasitas *steganografi* yang berguna untuk mengetahui dari perubahan kapasitas dan ukuran *file* sebelum proses terjadinya *embedding* dengan *file* setelah proses *stegofile*. Dengan melakukan perbadingan terhadap perubahan yang terjadi pada ukuran atau kapasistas dari kedua *file*. Kondisi dinyatakan baik dalam proses steganografi ketika ukuran atau kapasitas *file* baik sebelum ataupun setelah dilakukan proses penyisipan sama yaitu tidak mengalami perubahan. Ilustrasi pengujian ini terlihat pada gambar 15 berikut ini.




Gambar 15. Ilustrasi pengujian kapasitas, *file* sebelum disisip pesan (kiri) dan *file* setelah disisipi pesan (kanan)

Berdasarkan gambar 15 di atas dalam melakukan pengujian aspek kapasitas *steganografi* pada sistem, terlihat pada gambar sebelah kiri menunjukkan *file* asli sebelum disisipi pesan dengan *filename* "Kuesioner Dosen STMIK (DERI SUSANTI).pdf berkapasitas 322 KB memiliki ukuran sama dengan *file* setelah disisipi pesan berupa *file* gambar berformat **.jpeg* (kanan) dengan *filename* "DERI*.pdf" dimana kapasitas ukuran pesan yang disisipkan tetap sama 322 KB. Hal ini mengartikan bahwa aplikasi ini dinilai baik dalam melakukan proses *steganografi*. Dengan tidak terjadinya perubahan signifikan terhadap kapasitas pada kedua *file*.

Hasil pengujian terhadap *stegofile* lainnya secara detil dapat dilihat pada tabel 1.

Tabel 1: Hasil Pengujian pada Aspek Kapasitas *Steganografi*

| NAMA FILE (file) | UKURAN File | | UKURAN PESAN YANG DISISIPKAN Jenis Pesan Berupa File (kB) |
|---|---|---|---|
| | Sebelum | Sesudah | jpeg |
| Kuesioner Dosen STMIK (DERI SUSANTI).pdf | 322 KB | 322 KB | 7,78 |

Kesimpulan dari hasil pengujian terhadap aspek kapasitas *steganografi* sistem pada dokumen digital dalam bentuk *file* *.pdf diperoleh kesimpulan seperti berikut:

1. Dokumen digital berbentuk *file* *.pdf tidak mengalami perubahan kapasitas ataupun ukuran baik sebelum dan sesudah pesan disisipkan.
2. Secara kualitas pengujian dari sisi aspek kapasitas disimpulkan bahwa dokumen digital dengan format *file* *.pdf sangat direkomendasikan sebagai media penampung pesan yang mampu membawa hasil pesan baik berupa gambar *.*jpeg* yang disisipkan ke *.pdf.
3. *File* yang tersimpan yang telah mengalami proses steganografi, masih mampu dikembalikan kebentuk file awal tanpa merusak sedikitpun pesan tersebut.

## V. KESIMPULAN DAN SARAN

Setelah melakukan serangkaian tahapan implementasi hingga pengujian aspek terhadap kapasitas *steganografi LSB* dan *DES* pada sistem, maka dapat diambil beberapa kesimpulan disntaranya sebagai berikut:

1. Secara umum, implementasi *steganografi LSB* dan *DES* yang dilakukan berhasil dalam proses penyisipan (*embedding stego*) dan pengembalian pesan (*extracting stego*), yang berupa gambar tanda tangan hitam putih pada laporan penelitian ini.
2. Hasil pengujian yang ditunjukkan pada gambar 15 membuktikan bahwa tidak adanya perubahan pada kapasitas *file* *.pdf sebelum dan setelah disisipkan *file* gambar dengan kapasitas 322 KB. *Steganografi* yang baik adalah *steganografi* yang dapat menghasilkan *stegofile* yang harus sama besar dengan *file* *.pdf sebelum disisipi pesan.
3. Dengan adanya aplikasi ini data hasil kuesioner dapat diamankan dari pihak yang tidak bertanggung jawab sehingga data tersebut hanya dapat dibuka yang berkepentingan memiliki hak untuk membuka dokumen hasil kuesioner tersebut.

## DAFTAR PUSTAKA

[1] P. N. Andono, T. and M. , Pengolahan Citra Digital, Yogyakarta: ANDI OFFSET, 2017.

[2] A. Fauzi, "Analisa Kombinasi Pesan Teks Ke Dalam File Audio Memanfaatkan Algoritma Data Encryption Standard Dan Metode End Of File," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 3, no. 1, pp. 1-8, 2019.

[3] R. Munir, Pengolahan Citra Digital dengan Pendekatan Algoritmik, Bandung: Informatika, 2006.

[4] E. . Y. Hidayat and . K. Hastuti, "Analisis Steganografi Metode Least Significant Bit

(Lsb) Dengan Penyisipan Sekuensial Dan Acak Secara Kuantitatif Dan Visual," *Techno.COM,* vol. 12, no. 3, pp. 157-167, 2013.

[5] A. Siswanto, A. Syukur and I. Husna, "Perbandingan Metode Data Encryption Standard (Des) Dan Advanced Encryption Standard (Aes) Pada Steganografi File Citra," in *Seminar Nasional Teknologi Informasi Dan Komunikasi (SEMNASTIK) X*, Palembang, 2018.

[6] A. Andianto, Pemprograman Dasar Menggunkan JAVA dan NetBeans IDE, Pamekasan: Duta Media, 2017.

[7] T. Metode Penelitian Sistem 3x Baca, Yogyakarta: Deepublish, 2019.

[8] R. A. and M. S. , Rekayasa Perangkat Lunak : Terstruktur dan berorientasi objek, Bandung : Informatika, 2016.