IMPLEMENTASI TEKNIK *PLAYFAIR CIPHER* UNTUK PENYEMBUNYIAN TEKS TERENKRIPSI PADA CITRA DENGAN METODE *END OF FILE*

SKRIPSI

Oleh

Dwie Putri Donnaro
NIM 141910201001

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2018

IMPLEMENTASI TEKNIK *PLAYFAIR CIPHER* UNTUK PENYEMBUNYIAN TEKS TERENKRIPSI PADA CITRA DENGAN METODE *END OF FILE*

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Strata 1 Teknik
dan mencapai gelar Sarjana Teknik

Oleh

Dwie Putri Donnaro

NIM 141910201001

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2018

PERSEMBAHAN

Puji syukur saya panjatkan kepada Tuhan Yesus Kristus, atas kasih dan sayang-Nya kepada saya. Tidak hanya itu, berkat-Nya dan mujizat-Nya selalu menyertai jalan saya, sehingga saya bisa menyelesaikan penelitian ini.
Akhirnya, saya persembahkan skripsi ini kepada.

1. Kedua orang tua, Mama Sonny Ellen A. Sitompul dan Papa Romaison L. Tobing
2. Guru-guru dan dosen sejak taman kanak-kanak hingga perguruan tinggi;
3. Almamater tercinta, Jurusan Teknik Elektro Universitas Jember;
4. Serta seluruh teman-teman penulis yang saya kenal dan teman-teman yang membaca skripsi ini.

MOTO

I Trust The Lord God to Save Me, and I Will Wit For Him to Answer My Prayer.
(Micah 7:7)

Orang-orang yang menabur dengan mencucurkan air mata, akan menuai dengan bersorak-sorai.
(Mazmur 126:5)

Jangan berkecil hati jika orang lain hanya mengingat kita saat butuh pertolongan, saat susah, tapi cuek bebebk jika tidak butuh, sedang enak, seolah tidak kenal lagi. Karena dengan demikian, sebenarnya malah keren, kita dianggap seorang yang amat penting dalam hidupnya. Kita adalah "solusi", sedangkan dia adalah "masalah-nya".
(Tere Liye)

Bertangggung jawablah akan segala hal yang telah kamu lakukan, kerjakan dengan bersungguh-sungguh dan ikuti permainannya, maka engkau akan menikmati hasilnya.
(Dwie Putri Donnaro)

Jadilah orang yang selalu tersenyum baik dikala duka maupun senang, agar orang disekelilingmu nyaman berada didekatmu, meskipun mereka tidak tau kamu juga memiliki masalah yang bahkan lebih rumit dari yang mereka hadapi.
(Dwie Putri Donnaro)

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Dwie Putri Donnaro

NIM : 141910201001

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Implementasi Teknik *Playfair Cipher* untuk Penyembunyian Teks Terenkripsi pada Citra dengan Metode *End Of File*" adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 6 Juni 2018

Yang menyatakan

Dwie Putri Donnaro
NIM 141910201001

SKRIPSI

IMPLEMENTASI TEKNIK *PLAYFAIR CIPHER* UNTUK PENYEMBUNYIAN TEKS TERENKRIPSI PADA CITRA DENGAN METODE *END OF FILE*

Oleh

Dwie Putri Donnaro
NIM 141910201001

Pembimbing :

Dosen Pembimbing Utama : Catur Suko Sarwono, S.T., M.Si.

Dosen Pembimbing Anggota : Widya Cahyadi, S.T., M.T.

PENGESAHAN

Skripsi berjudul ”Implementasi Teknik *Playfair Cipher* Untuk Penyembunyian Teks Terenkripsi Pada Citra dengan Metode *End Of File*” telah diuji dan disahkan oleh Fakultas Teknik Universitas Jember pada :

Hari : Rabu
Tanggal : 6 Juni 2018
Tempat : Fakultas Teknik Universitas Jember

Tim penguji,

Ketua,	Sekretaris,
Catur Suko Sarwono, S.T., M.Si.	**Widya Cahyadi, S.T., M.T.**
NIP 196801191997021001	NIP 198511102014041001
Anggota I,	Anggota II,
Ike Fibriani, S.T., M.T.	**Alfredo Bayu Satriya, S.T., M.T.**
NIP 198002072015042001	NIP 198905192015041001

Mengesahkan
Dekan Fakultas Teknik

Dr. Ir. Entin Hidayah, M.U.M.
NIP 196612151995032001

RINGKASAN

Implementasi Teknik *Playfair Cipher* Untuk Penyembunyian Teks Terenkripsi Pada Citra dengan Metode *End Of File*; Dwie Putri Donnaro, 141910201001; 2018; 107 halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Pencurian data yang sering terjadi diberbagai bidang, yaitu rumah sakit, perbankan, departemen pemerintahan, perusahaan teknologi, sosial media, dan sebagainya. Oleh karena itu diperlukannya sistem keamanan data. Sistem keamanan data itu sendiri adalah sistem yang digunakan untuk memastikan bahwa data yang disimpan aman atau tidaknya dari pihak luar, dan cara membukanya selaras dengan cara menguncinya, sehingga data yang disimpan tersebut dapat bersifat privasi. Sistem keamanan salah satunya menyisipkan file teks kedalam sebuah gambar berwarna, yaitu menggunakan metode *End Of File* dan algoritma *Playfair Cipher* sebagai pengacakan enkripsinya. Untuk pengacakan enkripsinya yaitu dengan menggunakan matrik 7x5 yang telah dibuat, matrik ini terdiri dari 25 huruf kapital alfabet dan 10 angka.

Dalam pengujian terhadap citranya yaitu membandingkan nilai *pixel* RGB gambar asli dan gambar *embedding*, hasil yang didapat setelah melakukan perbandingan yaitu gambar asli dan gambar *embedding* tidak terdapat perubahan sama sekali. Kemudian melakukan pengujian dengan menambahkan *Brightness* dan *Darkness* sebesar 100, hasil yang disapat yaitu ukuran gambar yang telah ditambahkan *Brightness* dan *Darkness* menjadi berkurang dan file teks yang disisipi hilang karena file teks tersebut tidak tahan terhadap pengujian ini. Untuk pengujian kuesioner pengamat menyatakan gambar asli dan gambar *embedding* tidak ada perbedaan. Dan untuk pengujian MSE dan PSNR ketika mengujia gambar asli dan gambar *embedding* hasil MSE 0 dB dan PSNR 99 dB, sehingga gambar yang dihasilkan Bagus. Ketika menguji antara gambar asli dan gambar yang dicerahkan atau digelapkan nilai PSNR semuanya berada dibawah 20 dB, gambar yang dihasilkan sangat Buruk.

PRAKATA

Puji syukur kepada Tuhan Yesus Kristus atas segalanya, karena dengan kasih, pertolongan dan petunjukNya, penulis dapat menyelesaikan skripsi ini. Selama penyusunan skripsi ini penulis mendapat bantuan berbagai pihak yang turut memberikan bantuan berupa motivasi, inspirasi, bimbingan, doa, fasilitas dan dukungan lainnya yang membantu memperlancar pengerjaan skripsi ini. Untuk itu penulis mengucapkan terimakasih kepada.

1. Ibu Dr. Ir. Entin Hidayah, M.U.M., Selaku Dekan Fakultas Teknik Universitas Jember;
2. Bapak Dr. Bambang Srikaloko, S.T., M.T., Selaku Ketua Jurusan Teknik Elektro Universitas Jember dan dosen pembimbing akademik yang selalu membimbing dan memberi motivasi dari awal semester;
3. Bapak Catur Suko Sarwono, S.T., M.Si. dan Bapak Widya Cahyadi, S.T., M.T. selaku dosen pembimbing yang telah membimbing menyelesaikan tugas akhir ini;
4. Ibu Ike Fibriani, S.T., M.T. dan Bapak Alfredo Bayu Satriya, S.T., M.T. selaku dosen penguji yang sudah memberikan saran untuk memperbaiki tugas akhir ini;
5. Kedua Orang tua saya Mama Sonny Ellen A. Sitompul dan Papa Romaison L. Tobing, yang telah membesarkan, mendidik, mendoakan tiada henti, memberi motivasi semangat, menitikkan air mata dan memberi kasih sayang yang tak pernah habis serta pengorbanannya selama ini;
6. Kakak dan Adik kandung saya Christy Margretha R., Jonathan R., Naomi Safira dan Daud Markhesywan M. yang senantiasa menjadi inspirasi;
7. Muhammad Arief, Hanifatus Sa'diyah dan Dwi Sukma Aji yang sangat membantu dalam memberian semangat hingga saya bisa menyelesaikan skripsi ini;
8. Faiq Aprilian Romzi yang senantiasa membantu dalam pembuatan program skripsi saya ini.

9. Keluarga Laboratorium Jaringan Komputer dan Multimedia khususnya kepada Herlambang dan Rendra yang selama ini memberikan support dan bela-belain nungguin saya di lab untuk mengerjakan skripsi ini;
10. ABDHKN yang telah menjadi tempat curhat saya selama saya di Jember;
11. Terimakasih pula kepada KETEK UJ (Keluarga Teknik Elektro Universitas Jember) angkatan 2014 yang selalu mendampingi dan memberi semangat dimanapun berada;
12. Serta semua pihak yang tidak dapat disebutkan satu per satu, yang telah mendukung dalam penyelesaian skripsi ini.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang mambangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya;

Jember, 6 Juni 2018

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL ..... **i**
HALAMAN PERSEMBAHAN ..... **ii**
HALAMAN MOTO ..... **iii**
HALAMAN PERNYATAAN ..... **iv**
HALAMAN PEMBIMBING ..... **v**
HALAMAN PENGESAHAN ..... **vi**
RINGKASAN ..... **vii**
PRAKATA ..... **viii**
DAFTAR ISI ..... **x**
DAFTAR TABEL ..... **xiii**
DAFTAR GAMBAR ..... **xv**
BAB 1. PENDAHULUAN **1**
1.1 Latar Belakang **1**
1.2 Rumusan Masalah **3**
1.3 Batasan Masalah **3**
1.4 Tujuan Penelitian **4**
1.5 Manfaat Penelitian **4**
BAB 2. TINJAUAN PUSTAKA **5**
2.1 Kriptografi **5**
2.1.1 Sejarah Kriptografi 6
2.1.2 Algoritma Kriptografi 6
2.1.3 Jenis Kriptografi Berdasarkan Perkembangan 7
2.1.4 Jenis Kriptografi Berdasarkan Kunci 8
2.2 Algoritma *Playfair Cipher* **8**
2.3 Steganografi **11**
2.3.1 Sejarah Steganografi 12
2.3.2 Kegunaan Steganografi 13
2.3.3 Media Steganografi 13
2.4 Metode *End Of File* **14**

2.5 Citra Warna 16
2.6 Format Gambar..... 17
2.7 Resolusi Citra 18
2.8 Perhitungan PSNR dan MSE..... 18
2.9 Visual Basic 6.0..... 19
BAB 3. METODOLOGI PENELITIAN 21
3.1 Waktu dan Tempat Penelitian 21
3.2 Tahapan Perancangan Aplikasi 21
3.3 Alat dan Bahan 22
3.4 Desain Aplikasi 22
3.4.1 Desain Penyisipan (*Embedding*) 22
3.4.2 Desain Pengekstrakan 23
3.4.3 Desain Pengujian 23
3.5 Rancangan Matrik *Password* 24
3.6 Diagram alir proses penelitian yang dilakukan 27
3.7 Metodologi Pelaksanaan Penelitian 28
3.7.1 *Flowchart* Metode *End Of File*..... 28
3.7.2 *Flowchart* Algoritma *Playfair Chiper*..... 29
3.7.3 *Flowchart* Penyisipan File Teks Ke Dalam Gambar . 30
3.7.4 *Flowchart* Pengekstrakan File Teks..... 31
BAB 4. HASIL DAN PEMBAHASAN..... 32
4.1 Analisis Pengujian Aplikasi untuk File Teks Berformatkan .doc 32
4.1.1 Menyisipkan File Teks Berformatkan .doc 32
4.1.2 Mengekstrak Kembali File Teks .doc..... 37
4.2 Analisis Pengujian Aplikasi untuk File Teks Berformatkan .pdf 39
4.2.1 Menyisipkan File Teks Berformatkan .pdf..... 39
4.2.2 Mengekstrak Kembali File Teks .pdf..... 44
4.3 Analisis Pengujian Aplikasi untuk File Teks Berformatkan .txt 47

4.3.1 Menyisipkan File Teks Berformatkan .txt 47
4.3.2 Mengekstrak Kembali File Teks .txt 51
4.4 Analisis Pengujian Manipulasi Citra 53
4.4.1 Menambahkan *brightness* dan *darkness* sebesar 100. 53
4.4.1.1 Pada gambar berformat .bmp 53
4.4.1.2 Pada gambar berformat .gif 57
4.4.1.3 Pada gambar berformat .jpg 61
4.4.2 Pengujian data dengan flip horizontal dan flip vertikal 64
4.4.2.1 Pada gambar berformat .bmp 64
4.4.2.2 Pada gambar berformat .gif 68
4.4.2.3 Pada gambar berformat .jpg 72
4.5 Analisis Perbandingan Histogram RGB Gambar Asli dan Gambar Embedding 75
4.5.1 Format gambar .bmp dengan resolusi 200x149 76
4.5.2 Format gambar .bmp dengan resolusi 300x224 78
4.5.3 Format gambar .bmp dengan resolusi 400x299 79
4.5.4 Format gambar .gif dengan resolusi 200x107 76
4.5.5 Format gambar .gif dengan resolusi 300x161 84
4.5.6 Format gambar .gif dengan resolusi 400x215 86
4.5.7 Format gambar .jpg dengan resolusi 200x130 88
4.5.8 Format gambar .jpg dengan resolusi 300x196 90
4.5.9 Format gambar .jpg dengan resolusi 400x261 93
4.6 Analisis Pengujian Berdasarkan Visual 95
4.7 Analisis Pengujian Kualitas Citra Dengan PSNR dan MSE 102
BAB 5. KESIMPULAN DAN SARAN 107
5.1 Kesimpulan 107
5.2 Saran 107
DAFTAR PUSTAKA 108
LAMPIRAN 109

DAFTAR TABEL

Halaman

2.1 Jenis citra dilihat dari ukuran bitnya 14
2.2 Kode Warna RGB 16
2.3 Skala PSNR 19
4.1 Hasil Penyisipan (*Embedding*) File Teks .doc ke dalam gambar berwarna (RGB) 33
4.2 Hasil Ekstrak File Teks .doc 37
4.3 Hasil Penyisipan (*Embedding*) File Teks .pdf ke dalam gambar berwarna (RGB) 40
4.4 Hasil Ekstrak File Teks .pdf 44
4.5 Hasil Penyisipan (*Embedding*) File Teks .pdf ke dalam gambar Berwarna (RGB) 47
4.6 Hasil Ekstrak File Teks .txt 51
4.7 Hasil Pengujian dengan Menambah *Brigthness* dan *Darkness* Pada Gambar Berformat .bmp 53
4.8 Hasil Pengujian dengan Menambah *Brigthness* dan *Darkness* Pada Gambar Berformat .gif 57
4.9 Hasil Pengujian dengan Menambah *Brigthness* dan *Darkness* Pada Gambar Berformat .jpg 61
4.10 Hasil Pengujian Flip Horizontal dan Flip Vertikal Pada Gambar Berformat .bmp 64
4.11 Hasil Pengujian Flip Horizontal dan Flip Vertikal Pada Gambar Berformat .gif 68
4.12 Hasil Pengujian Flip Horizontal dan Flip Vertikal Pada Gambar Berformat .jpg 72
4.13 Hasil Pengujian Dengan Penilaian Kuesioner 96
4.14 Nilai Tingkat Kecerahan 97
4.15 Nilai Tingkat Kualitas 97
4.16 Nilai Tingkat Ketajaman Warna 98
4.17 Hasil Perhitungan MSE dan PSNR Menggunakan Program Pada

Citra Asli dan Citra *Embedding* Berformatkan .bmp 102
4.18 Hasil Perhitungan MSE dan PSNR Menggunakan Program Pada
Citra Asli dan Citra yang Dicerahkan Berformatkan .gif 103
4.19 Hasil Perhitungan MSE dan PSNR Menggunakan Program Pada
Citra Asli dan Citra yang Digelapkan Berformatkan .jpg 105

DAFTAR GAMBAR

Halaman

2.1 Proses enkripsi dan dekripsi pada kriptografi 5
2.2 Matrik 5x5 *Playfair Cipher* 9
2.3 Proses steganografi secara umum 13
2.4 Citra sebelum disisipi pesan dan citra setelah disisipi pesan dengan metode EOF 15
2.5 *Icon* Aplikasi Visual Basic 6.0 20
3.1 Desain aplikasi penyisipan file teks kedalam gambar berwarna 22
3.2 Desain untuk pengekstrakan file teks 23
3.3 Desain pengujian data 23
3.4 Matrik *Password* 7x5 24
3.5 Matrik *Password* 25
3.6 Proses enkripsi 1 26
3.7 Proses enkripsi 2 26
3.8 Proses enkripsi 3 26
3.9 Diagram alir penelitian 27
3.10 Flowchart Metode *End Of File* 28
3.11 Flowchart Algoritma *Playfair Cipher* 29
3.12 Flowchart Penyisipan File Teks Kedalam Gambar 30
3.13 Flowchart Pengekstrakan File Teks 31
4.1 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc 76
4.2 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf 76
4.3 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt 77
4.4 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc 78
4.5 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf 78
4.6 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt 79
4.7 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc 80
4.8 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf 80
4.9 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt 81
4.10 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc 82

4.11 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf.... 82
4.12 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt..... 83
4.13 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc ... 84
4.14 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf.... 85
4.15 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt..... 85
4.16 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc ... 86
4.17 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf.... 87
4.18 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt..... 87
4.19 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc ... 88
4.20 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf.... 89
4.21 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt..... 90
4.22 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc ... 91
4.23 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf.... 91
4.24 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt..... 92
4.25 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .doc ... 93
4.26 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .pdf.... 94
4.27 Perbandingan Grafik Gambar Asli dan Gambar disisipi File .txt..... 94
4.28 Hasil penilaian kuesioner *online* untuk tingkat kecerahan pada gambar 1 dan gambar 2 99
4.29 Hasil penilaian kuesioner *online* untuk tingkat kualitas pada gambar 1 dan gambar 2 100
4.30 Hasil penilaian kuesioner *online* untuk tingkat ketajaman warna pada gambar 1 dan gambar 2 101

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Sejak tahun 2000-an hingga sekarang pencurian data sering terjadi. Tidak hanya didalam negeri, bahkan pencurian data tersebut juga terjadi diluar negari. Apalagi pencurian data ini sering terjadi dalam suatu rumah sakit, asuransi, perbankan, departemen pemerintahan, perusahaan teknologi, sosial media, dan masih banyak lagi. Oleh karena itu diperlukannya sistem keamanan data untuk menyimpan data dalam berbagai bidang, salah satunya yaitu penyembunyian data pada sebuah gambar. Hal ini dilakukan karena masih banyak yang menyimpan data tersebut dalam *flashdisk* ataupun *harddisk*. Memang cara ini aman dilakukan, namun suatu saat pasti *flashdisk* atau *harddisk* ini akan hilang karena lupa menaruhnya dimana, atau bahkan bisa di format oleh orang yang berniat jahat. Tujuan dari penelitian ini agar membuat file teks tersebut lebih aman dan tidak dapat dicurigai oleh pihak luar, yaitu dengan menyisipkan file teks kedalam gambar berwarna dan hasil keluarannya berupa gambar berwarna kembali namun sudah disisipi file teks didalamnya atau biasa disebut dengan gambar *embedding*. Agar file teks tersebut lebih aman, nantinya akan di enkripsi terlebih dahulu dengan mengacak *password*-nya hingga menjadi kode-kode yang sulit dimengerti. *Password* yang digunakan bersifat pribadi sehingga pihak luar tidak dapat mengetahui *password* apa yang digunakan untuk mengamankan file teks tersebut.

Teknik penyisipan file teks tersebut dinamakan steganografi, dan salah satu metode dari steganografi adalah *end of file*. Metode *end of file* memiliki algoritma untuk menyisipkan data pada akhir nilai matrik gambar, hasil dari metode ini tidak dapat dilihat oleh kasat mata. Namun saat penyisipan file teks dengan menggunakan metode ini ukuran file teks tersebut tidak terbatas, sehingga kapasitas ukurannya akan bertambah besar dari sebelumnya, jika pihak luar lebih teliti akan ukuran gambar tersebut maka akan menimbulkan kecurigaan. Oleh karena itu saat proses penyisipan data diharapkan agar bentuk atau hasil gambar tidak mengalami perubahan dari gambar aslinya.

Pada penelitian sebelumnya menguji data yang telah tersisipkan dengan 5 buah pegujian yaitu verifikasi, validasi, deteksi error, pengubahan format file induk *.pdf yang sudah terenkripsi menjadi file *.docx dan hasil apabila kapasitas file pesan melebihi kapasitas file induk. Untuk pengujian verifikasi dan validasi menggunakan kuisioner, sedangkan untuk pengujian yang lainnya menggunakan *blackbox* dan *whitebox*. Kunci yang digunakan untuk lebih mengamankan filenya peneliti menggunakan salah satu algoritma berjenis *asymetric key* yaitu, *stream cipher* (Bely Arifpriyanto).

Agar file teks yang akan disisipkan kedalam gambar tersebut lebih aman diperlukannya kunci yang dinamakan teknik kriptografi. Salah satu teknik kriptografi adalah algoritma *playfair cipher*, prinsip kerja dari kunci ini yaitu akan mengacaknya dan menghilangkan huruf J. Hal ini dilakukan karena matrik yang digunakan berukuran 5x5, sehingga huruf yang digunakan berjumlah 25. Pada dasarnya penghilangan satu huruf tersebut tidak harus J, dapat diganti dengan huruf yang lain. Namun peneliti melakukannya berdasarkan ketetapan algoritma yang ditemukan oleh Sir Charles Wheatstone dan Baron Lyon pada tahun 1854, mereka menganggap jika huruf J dan I berada pada satu matrik yang sama maka, akan menimbulkan ambigu. Sehingga mereka harus menghilangkan salah satunya, yaitu huruf J.

Pada penelitian sebelumnya kunci *playfair cipher* dikembangkan lagi menjadi super *playfair*, ini merupakan gabungan antara algoritma *playfair cipher* dengan super enkripsi. Dengan menggabungkan kedua algoritma tersebut semakin banyak kunci untuk *playfair cipher* yang dapat diciptakan (Egi Andriana, 2016).

Dari kedua penelitian tersebut terdapat kelebihan dari masing-masing penelitiannya. Berdasarkan penelitian diatas, pada tugas akhir ini peneliti akan menggabungkannya antara algoritma dan metode tersebut. Proses penelitian ini dilakukan dengan sistem *offline*, dimana peneliti menyisipkan data tersebut dengan mengubahnya kedalam bentuk desimal kemudian menyisipkannya diakhir nilai desimal citra warna. Kemudian mengacak kunci yang dibuat agar tidak dapat diketahui oleh pihak luar. Kunci ini hanya dapat dibuka jika pihak yang bersangkutan mengetahui cara membukanya (dekripsi). Hal ini dilakukan agar

keamanan data tersebut dapat diterapkan oleh berbagai bidang. Data yang didapatkan diharapkan tidak mengalami perubahan dari segi manapun, jika gambar yang telah disisipi file tersebut berubah dari segi tingkat kecerahannya, bentuk, warna, maka pihak luar akan langsung mencurigai bahwa didalam gambar berwarna tersebut terdapat sesuatu.

1.2 Rumusan Masalah

Dalam penelitian ini dirumuskan masalah, adalah :

a. Bagaimana merancang dan mengimplementasikan kriptografi *Playfair Cipher* dengan metode *End Of File* dalam mengenkripsi dan mendekripsi file teks?
b. Bagaimana ukuran dan kualitas citra setelah disisipi file teks yang telah dienkripsi menggunakan kriptografi *Playfair cipher* dengan metode *End Of File*?
c. Bagaimana pengaruh pengujian terhadap ketahanan file teks pada gambar yang telah dienkripsi menggunakan kriptografi *Playfair cipher* dengan metode *End Of File*?

1.3 Batasan Masalah

Penelitian ini berisikan tentang batasan-batasan yang terdapat pada perancangan dan simulasi yaitu:

1. Menggunakan *software* Visual Basic 6.0 untuk penyisipan, pengekstrakan dan pengujian pada penelitian ini selain MSE dan PSNR.
2. Hanya menggunakan *software* Matlab untuk menguji PSNR dan MSE
3. Tidak menampilkan bilangan ASCII
4. Password yang digunakan berdasarkan matrik 7x5 yang telah dibuat
5. Gambar yang digunakan gambar berwarna dan hanya berformatkan .bmp, .gif dan .jpg
6. Hanya menyisipkan file berformatkan .doc, .pdf dan .txt
7. Berfokuskan pada proses penyisipan file teks untuk keamanan datanya dan pengekstrakan file teks.

1.4 Tujuan Penelitian

Berdasarkan permasalahan yang diuraikan di atas, tujuan dari penelitian ini adalah :

1. Merancang dan mengimplementasikan kriptografi *Playfair Cipher* dengan metode *End Of File* dalam mengenkripsi dan mendekripsi file teks.
2. Menghasilkan gambar *embedding* yang sama persis seperti gambar aslinya agar tidak menimbulkan kecurigaan pihak luar yang melihatnya.
3. Mengetahui pengaruh pengujian terhadap ketahanan file teks yang telah disisipi kedalam gambar.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian yang akan dilakukan, yaitu :

1. Dapat merancang dan mengimplementasi kriptografi *Playfair Cipher* dengan metode *End Of File* dalam mengenkripsi dan mendekripsi file teks.
2. Dapat menghasilkan gambar *embedding* yang sama persis seperti gambar aslinya agar tidak menimbulkan kecurigaan pihak luar yang melihatnya.
3. Dapat mengetahui pengaruh pengujian terhadap ketahanan file teks yang telah disisipi ke dalam gambar.

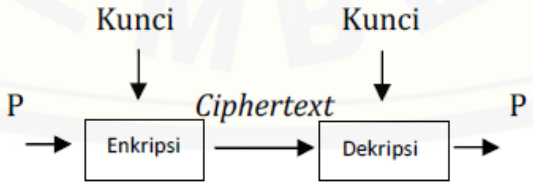
BAB 2. TINJAUAN PUSTAKA

Pada bab ini akan menjelaskan tentang teori dari setiap teknik yang digunakan dan metode-metode yang mendukung proses penyisispan data pada *cover* (*image*). Menganalisa hasilnya setelah disisipkan dan menguji ketahanan dari metode yang digunakan. Kemudian mempelajari keunggulan dan kelemahannya.

2.1 Kriptografi

Kriptografi merupakan teknik keamanan data dengan cara mengacakan suatu pesan. Kriptografi juga merupakan karya seni yang menjaga keamanan data ketika pesan dikirim dari suatu alamt, kealamt yang lain. Kriptografi berasal dari bahasa yunani, menurut bahasa dibagi menjadi dua yaitu kripto dan graphia, kripto berarti *secret* (rahasia) dan graphia berarti *writing* (tulisan).

Kriptografi merupakan salah satu teknik dari beberapa teknk keamanan data yang sering digunakan untuk mengamankan data, seperti halnya menyandikan pesan asli (*plaintext*) ke dalam bentuk pesan rahasia (*ciphertext*). Proses pengamanan ini melibatkan algoritma dan kunci. Kunci yang telah dienkripsi dapat dengan mudah didekripsi kembali, yaitu dari *ciphertext* menjadi *plaintext*. Oleh karena itu diperlukan algoritma kriptografi yang kuat. Namun teknik ini masih menimbulkan kecurigaan pada orang lain yang melihat pesan tersebut. (Simbolon, 2016)



Gambar 2.1 Proses enkripsi dan dekripsi pada kriptografi
(Sumber : Simbolon, 2016)

2.1.1 Sejarah Kriptografi

Pada zaman Romawi kuno dahulu, ada seorang kaisar bernama Julius Caesar. Kaisar ini ingin mengirimkan suatu pesan yang bersifat rahasia kepada seorang jenderal yang berada dimedan peperangan. Pesan yang hendak disampaikan tidak dapat dikirimkan secara langsung, karena masyarakat yang berada didalam maupun diluar istana dapat mengetahui pesan tersebut. Kemudian kaisar terpikir untuk mengirimkan pesan tersebut melalui seorang kurir, tetapi karena pesan tersebut mengandung rahasia, kaisar Julius Caesar tidak ingin pesan tersebut terbuka, hilang atau bahkan dicuri di tengah jalan. Akhirnya kaisar Julius Caesar memikirkan bagaimana cara mengatasinya yaitu denga mengacak pesan tersebut sehingga hanya jenderal yang berada dimedan perang tersebut saja yang dapat memahami pesan tersebut. Sebelum pesan tersebut sampai ke jenderal, Julius Caesar memberi tahu jenderal bagaimana cara membaca pesan yang akan di acak tersebut. Kaisar mengganti semua susunan alfabet pesan dari a, b, c, dan seterusnya, dengan cara menggeser 3 alfabet dari alfabet semula, misalkan alfabet a menjadi d, b menjadi e, c menjadi f dan seterusnya.

Hal ini hampir sama dengan kasus pada perang dunia kedua antara jerman dan sekutu. Jerman menggunakan enigma atau juga disebut dengan mesin rotor yang digunakan Hitler untuk mengirim pesan ke tentaranya. Pada saat itu Jerman sangat percaya bahwa pesan yang dikirim melalui enigma tidak dapat terpecahkan kode-kode enkripsinya. Tapi anggapan itu salah, ketika Jerman mengirimkan pesan ke tentara - tentaranya setelah bertahun-tahun sekutu dapat memecahkan kode-kode tersebut, karena sekutu telah mempelajari metode pemecahan kode enigma terlebih dahulu. (Ariyus, 2006).

2.1.2 Algoritma Kriptografi

Algoritma adalah langkah-langkah untuk penyelesaian masalah yang disusun secara sistematis. Algoritma kriptografi merupakan langkah - langkah mengacak pesan dari orang-orang yang tidak berhak atas pesan tersebut. Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu :

a. Enkripsi

Merupakan hal yang sangat diperlukan dalam kemanan data karena berfungsi untuk menjaga kerahasiaan suatu pesan melalui pengkodean. Pesan asli (*plaintext*) akan diubah menjadi kode-kode yang tidak mudah dimengerti (*ciphertext*).

b. Dekripsi.

Merupakan kebalikan dari enkripsi, yaitu pesan yang telah dienkripsi dikembalikan kebentuk asalnya (*Plaintext*) disebut dengan dekripsi pesan. Dengan kata lain *ciphertext* menjadi *plaintext*

c. Kunci

Dibutuhkan untuk mengamankan pesan saat melakukan enkripsi dan dekripsi, kunci terbagi jadi dua bagian yaitu kunci pribadi (*private key*) dan kunci umum (*public key*).

2.1.3 Jenis Kriptografi Berdasarkan Perkembangan

Algoritma kriptografi dapat diklasifikasikan menjadi dua jenis berdasarkan perkembangannya, yaitu kriptografi klasik dan kriptografi modern.

a. Algoritma Kriptografi Klasik

Algoritma kriptografi klasik digunakan sejak sebelum era komputerisasi ada, kebanyakan manusia dulu menggunakan teknik kunci simetris. Teknik kriptografi simetris yang sering digunakan untuk mengacak pesan adalah teknik substitusi dan transposisi atau bahkan menggabungkan kedua teknik tersebut. Teknik substitusi adalah teknik untuk menggantikan suatu karakter dalam pesan asli menjadi karakter lain yang hasilnya adalah pesan teracak yang telah dienkripsi. Sedangkan teknik transposisi yaitu teknik mengubah pesan asli menjadi pesan teracak yang telah dienkripsi dengan cara permutasi karakter. Penggabungan kedua teknik tersebut dapat mengahasilkan atau membentuk berbagai macam algoritma kriptografi klasik lainnya.

b. Algoritma Kriptografi Modern

Algoritma Kriptografi Modern merupakan perkembangan dari Algoritma Kriptografi Klaisik. Algoritma ini memiliki tingkat kesulitan

yang lebih sulit dibandingkan dengan algoritma kriptografi klasik, dan kekuatan dari pengacakan pesan terdapat pada kuncinya. Algoritma kriptografi modern menggunakan pengolahan simbol biner karena berjalan mengikuti operasi komputer digital. Sehingga membutuhkan dasar berupa pengetahuan terhadap matematika untuk menguasainya.

2.1.4 Jenis Kriptografi Berdasarkan Kunci

Algoritma kriptografi dikelompokkan menjadi dua jenis berdasarkan kuncinya, yaitu algoritma simetris dan algoritma asimetris, yaitu :

a. Algoritma Simetris

Algoritma simetris bersifat sebagai algoritma tunggal, karena kunci atau sandi hanya untuk si pembuat dan orang yang akan menerima pesan tersebut. Proses enkripsi dan dekripsi algoritma ini sangat rahasia sehingga kunci atau sandi tersebut tidak akan dapat diketahui oleh umum.

b. Algoritma Asimetris

Berbeda dengan algoritma simertris, algoritma asimetris bersifat umum, karena daat dikirm ke satu orang bahkan ke publik. Karena proses enkripsi kunci ini diperuntukkan untuk umum, sehingga sehingga pihak ketiga dapat mengetahui isi pesan yang akan disampaikan. Sedangkan untuk proses dekripsi, kuncinya bersifat rahasia, sehingga kunci atau sandi tersebut hanya diketahui oleh orang tertentu. Dengan kata lain hanya orang yang dapat dipercaya oleh si pembuat yang dapat memecahkan kode-kode enkripsinya kunci tersebut.

2.2 Algoritma *Playfair Cipher*

Playfair Cipher ditemukan oleh Sir Charles Wheatstone dan Baron Lyon *Playfair* pada tahun 1854 dan digunakan pertama kali pada awal abad 20, untuk mengirim pesan antar markas yang ada di inggris pada masa perang dunia pertama. Kunci dari *playfair* menggunakan matriks 5x5 (dengan input terdiri dari 25 karakter dan membuang J yang ada didalam alfabet), dan dengan begitu kunci yang digunakan ada 25 alphabet (Ariyus, 2006).

Playfair cipher juga merupakan kriptografi klasik yang penyandiannya menggunakan subtitusi. Untuk penyandian pada metode ini tidak tunggal melainkan alfabetnya harus ganda. Kelemahan yang lain pada *playfair* adalah terjadinya ambigu pada hasil dekripsi karena pada persiapan enkripsi *playfair cipher* memiliki mekanisme mengganti J dengan I. Perlunya modifikasi tabel *playfair chiper* yang dapat digunakan untuk melakukan enkripsi huruf kapital, huruf kecil, angka dan simbol. Mengacak isi tabel diperlukan agar cipherteks yang dihasilkan menjadi acak (Nurkifli, 2014).

Ada beberapa aturan dalam membuat sandi pada playfair cipher, yaitu:

1. Jika huruf *plaintext* berada pada 1 baris maka untuk membuat sandinya itu merupakan huruf yang berada disebelahnya.
2. Jika huruf tersebut berada saru baris atau dibaris dan kolom berbeda, maka sandi untuk huruf itu berada dibawahnya
3. Jika huruf *plaintext* berada terbalik dengan tabel maka sandi yang akan dihasilkan akan dibaca terbalik, dengan kata lain yang semula bacanya kiri ke kanan menjadi kanan ke kiri.
4. Jika huruf *plaintext* berada pada kolom dan baris yang berbeda atau menyerong maka, untuk membuat sandinya merupakan huruf yang menyerong juga. Misalkan, sandi yang akan di playfair cipher yaitu “HARI MERDEKA NEGARAKU” dengan kunci “SEMUT”.
5. Jika terdapat huruf yang ganda pada *plaintext* harus disisipikan huruf x atau z. Tetapi lebih baik menyisipkan huruf x dari pada z karena huruf z sering terjadi kesamaan pada suatu sandi.

S	E	M	U	T
A	B	C	D	F
G	H	I	K	L
N	O	P	Q	R
V	W	X	Y	Z

Gambar 2.2 Matrik 5x5 *Playfair Cipher*

- Proses pembuatan sandi dengan menggunakan aturan kedua, yaitu mengambil huruf yang berada dibawahnya sebagai kunci

1. HA RI akan menjadi OG ZP

S	E	M	U	T
A	B	C	D	F
G	H	I	K	L
N	O	P	Q	R
V	W	X	Y	Z

2. ME RD akan menjadi CB ZK

S	E	M	U	T
A	B	C	D	F
G	H	I	K	L
N	O	P	Q	R
V	W	X	Y	Z

3. EK AN akan menjadi BQ GV

S	E	M	U	T
A	B	C	D	F
G	H	I	K	L
N	O	P	Q	R
V	W	X	Y	Z

4. EG AR akan menjadi BN GZ

S	E	M	U	T
A	B	C	D	F
G	H	I	K	L
N	O	P	Q	R

V	W	X	Y	Z

5. AK UZ akan menjadi GQ DT

S	E	M	U	T
A	B	C	D	F
G	H	I	K	L
N	O	P	Q	R
V	W	X	Y	Z

Sehingga didapatkan :

Plaintext : HA RI ME RD EK AN EG AR AK UZ

Ciphertext : OG ZP CB ZK BQ GV BN GZ GQ DT

2.3 Steganografi

Steganografi adalah ilmu yang mempelajari tentang penyembunyian pesan berupa dokumen, data, audio, bahkan video sedemikian rupa, sehingga secara kasat mata tidak dapat melihat perubahannya, oleh karena itu teknik ini tidak akan menimbulkan kecurigaan. Pada steganografi memerlukan kunci khusus agar keamanannya terjamin.

Ada beberapa hal yang diperlukan untuk menyembunyikan pesan pada steganografi, yaitu:

1. Algoritma Penyisipan (*Embeding Algorithm*).

   Algoritma tersebut digunakan untuk menyisipkan suatu pesan yang bersifat rahasia dan disisipkan kedalam data yang akan dikirimkan. Proses penyisipan tersebut dilindungi oleh sebuah *key-word* sehingga hanya orang-orang tertentu yang dapat membaca pesan tersebut.

2. Fungsi Detektor (*Detector Function*).

   Fungsi Detektor digunakan untuk mengembalikan pesan-pesan yang telah disembunyikan.

3. *Carrier Document.*

Carrier Document Merupakan dokumen yang berfungsi sebagai wadah penyisipan pesan. Dokumen ini dapat berupa *file-file* seperti *file* audio, video, gambar, teks dan sebagainya.

4. *Key* (Kunci)

Biasanya digunakan untuk proses log in atau membuka suatu aplikasi dengan kode-kode tertentu yang sudah diacak.

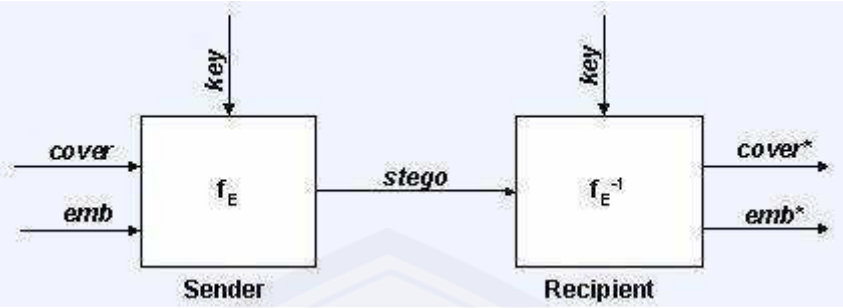
5. *Secret Message/ Plaintext*

Merupakan pesan rahasia yang akan disisipkan kedalam *carrier document*. Pesan inilah yang tidak ingin terlihat dan terbaca oleh orang yang tidak berkepentingan.

2.3.1 Sejarah Steganografi

Pada zaman dahulu kala ada seorang kaisar bernama Caesar yang pertama sekali menemukan teknik steganografi. Namun prinsip kerja dari steganografinya dulu adalah dengan menuliskan sebuah pesan pada papan yang nantinya akan dilapisi oleh lilin sehingga pesan tersebut menjadi sulit dibaca. Proses penyampaian pesan tersebut harus dikirim melalui seorang kurir yang dipercaya. Kurir tersebut tidak akan diberi tahu kode pemecah pesan tersebut, namun penerima pesanlah yang akan diberitahu cara memecahkan pesan tersebut. Sejak saat itu kaisar caesar sering mengirimkan pesannya decara diam-diam agar musuh tidak mengetahuinya. Dan setelah beberapa lama beliau menggunakan metode tersebut, ternyata musuhnya mengetahui trik tersebut, dan kasiar caesar berhenti menggunakan teknik tersebut.

Tujuan dari *steganography* adalah menyembunyikan keberadaan sebuah pesan. Dalam prakteknya kebanyakan diselesaikan dengan membuat perubahan kecil terhadap pesan digital yang isinya tidak akan membuat curiga bagi pihak luar, misalkan sebuah sebuah audio yang telah berisikan file namun hasil suaranya tetap jernih.



Gambar 2.3 Proses steganografi secara umum

(Sumber : Gemita Ria, 2013)

2.3.2 Kegunaan Steganografi

Steganografi dapat digunakan untuk menyimpan kerahasiaan pesan yang penting, untuk menjaga pesan tersebut dari kemungkinan pencuri dari pihak luar yang memiliki niatan tidak baik.

Namun, steganografi juga dapat digunakan untuk tindakan ilegal. Misalkan, seseorang telah mencuri data, mereka dapat menyembunyikan arsip curian tersebut ke dalam arsip lain dan mengirimkannya keluar tanpa menimbulkan kecurigaan siapapun karena tampak seperti email atau arsip normal. Selain itu, banyak tindakan kejahatan lain yang dilakukan seseorang menggunakan steganografi, misalkan saja video pornografi, tindakan teroris dalam artian data perusahaan yang dicuri dan sebagainya.

2.3.3 Media Steganografi

Hampir semua file digital dapat digunakan untuk steganografi, tetapi format yang paling cocok adalah yang mempunyai nilai bits *redundancy* tinggi, salah satunya adalah citra.

File Citra pada komputer merupakan *array* bilangan yang merepresentasikan nilai intensitas cahaya yang bervariasi (*pixel*). Kumpulan *pixel-pixel* inilah yang membentuk suatu citra. Citra yang sering digunakan umum adalah citra 24 bit dan citra 8 bit (256 *colors*).

Tabel 2.1 Jenis citra dilihat dari ukuran bitnya.

Jumlah Bit	Keterangan
1	Nilai Biner Gambar (0 – 1)
8	Level keabuan (0 – 255)
16	Warna Tinggi (216)
24	Warna Asli (224)
32	Warna Asli (232)

Format gambar digital memiliki 2 parameter :

- Resolusi spasial : Piksel x Piksel
- *color encoding* : Bit / Piksel

2.4 Metode *End Of File*

Metode *End Of File* (EOF) merupakan metode penyisipan data pada akhir file. Hasil dari metode ini tidak dapat dilihat oleh kasat mata, seolah-olah tidak ada perubahan pada gambar sebagai *cover* dari media penyisipan data tersebut. Dengan menggunakan metode ini saat penyisipan ukuran file sebelum disisipkan data harus sama dengan sesudah disipkan data, jika setelah disisipkan data file tersebut ukurannya bertambah besar maka akan menimbulkan kecurigaan pada pihak yang melihatnya. Metode EOF ini merupakan metode hasil perkembangan dari metode *Least Signifikan Bit* (LSB). Prinsip kerja dari metode ini yaitu terdapat penanda khusus jika metode ini terserang ancaman dari luar, jika hal itu terjadi maka sistem akan berhenti membaca proses yang sedang berjalan. Adapun proses penyisipan (Enkripsi) pada *End Of File*, yaitu :

1. Masukkan berupa gambar sebagai *cover* atau sebagai media penampungan penyisipan data.
2. Masukkan berupa data yang akan disisipkan pada gambar.
3. Ubah masing-masing masukkan menjadi kode desimal berdasarkan tabel ASCII.
4. Dapatkan nilai derajat keabuan dari setiap piksel.

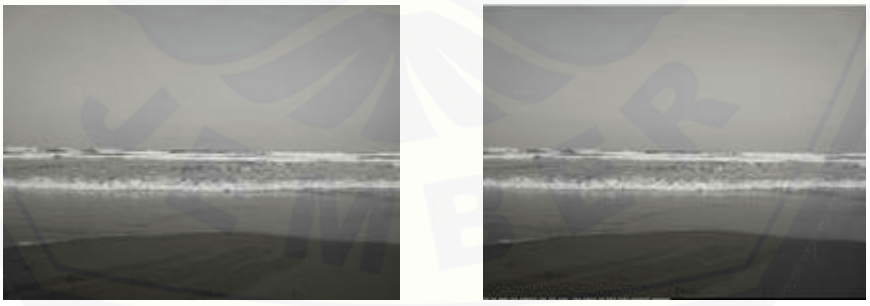
5. Mencari nilai akhir dari matrik gambar.
6. Menyisipkan kode desimal data pada akhir nilai matrik gambar.
7. Merubah kembali kode desimal setelah proses penyisipan menjadi gambar.
8. Menyimpan hasil citra yang telah tersisipi data (*stego image*).

Adapun untuk proses dekripsi pada *End Of File*, yaitu :

1. Masukkan gambar yang telah tersisipi data.
2. Mengubahnya kedalam kode desimal berdasarkan tabel ASCII.
3. Mendapatkan nilai derajat keabuan pada gambar.
4. Mencari nilai akhir matrik pada gambar tersebut.
5. Mengubah nilai akhir tersebut menjadi data dan menampilkan data

Matrik tingkat derajat keabuan citra 6x6 piksel dengan kode biner pesan "**#aku**" disisipkan diakhir citra sehingga citra menjadi (Krisnawati, 2008) :

196	10	97	182	100	40
67	200	100	50	90	50
25	150	45	200	75	28
176	56	77	100	25	200
101	34	250	40	100	60
44	66	99	125	190	200
97	**107**	**117**	**35**		



(a). Citra sebelum disisipi pesan (b). Citra setelah disisipi pesan dengan metode EOF

Gambar 2.4 (a). Citra sebelum disisipi pesan dan (b). Citra setelah disisipi pesan dengan metode EOF (Krisnawati, 2008).

2.5 Citra Warna

Citra warna adalahkumpulan dari berbagai warna yang memiliki tingkat disetiap warnanya. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*).

RGB merupakan penyusun citra warna atau dapat dikatakan sebagai induk yang akan menciptakan warna-warna lainnya. Setiap warna dasar, yaitu merah, hijau dan biru dapat diberi rentang nilai. Pada komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada 8 digit bilangan biner yang digunakan oleh komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak 256 x 256 x 256 = 1677726 jenis warna. Jenis warna, dapat diibaratkan sebagai sebuah vektor di ruang dimensi 3 yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-*x*, komponen-*y* dan komponen-z. Misalkan sebuah vektor dituliskan sebagai r = (*x* ,*y*, *z*). Komponen-komponen tersebut digantikan oleh komponen R(ed), G(reen), B(lue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut:

- warna = RGB (30, 75, 255).
- Putih = RGB (255, 255, 255)
- Hitam = RGB (0, 0, 0).

Warna primer dapat digunakan untuk menghasilkan warna sekunder, yaitu :

- Magenta = merah + biru
- Cyan = hijau + biru
- Kuning = merah + hijau

Tabel 2.2 Kode Warna RGB

Colour	*Red*	*Green*	*Blue*
Black	0	0	0
Blue	0	0	255
Green	0	255	0
Cyan (Blue + Green)	0	255	255
Red	255	0	0

Magenta (Red + Blue)	255	0	255
Yellow (Red + Green)	255	255	0
White (Red + Green + Blue)	255	255	255
Gray	128	128	128

2.6 Format Gambar

Gambar merupakan media yang sering dilihat oleh indra manusia, karena warna yang terkandung pada gambar tersebut sangat variasi. Untuk format penyimpanan gambar memiliki fungsi, kelebihan, dan kekurangan masing-masing, yaitu :

1. BMP (Bitmap)

Bitmap merupakan susunan citra grafis yang terdiri dari titik-titik yang terdapat pada memori penyimpanan komputer. Pada bitmap nilai setiap titik diawali dengan 1 bit untuk warna hitam-putih atau lebih untuk warna warni.

- Kelebihan dari bitmap yaitu dapat menyimpan informasi dimulai dari 1 bit hingga 24 bit.
- Kekurangannya adalah gambar bitmap sangat bergantung pada resolusi citra, jika resolusi gambar tersebut diperbesar maka file yang akan dihasilkan akan semakin besar pula dan hasil gambar tersebut akan pecah-pecah.

2. GIF (*Graphics Interchange Format*)

Graphics Interchange Format atau yang lebih dikenal dengan GIF merupakan format gambar yang terdiri dari 8 bit warna dengan kata lain dibatasi oleh 256 jenis warna, format gambari ini merupakan warna standard yang warnanya berdasarkan palet warna. GIF juga dapat menyimpan gambar berupa animasi namun dengan kapasitas yang kecil, selain itu GIF juga dapat digunakan untuk desain web yang dapat menyimpan latar belakang transparan dalam bentuk animasi yang sederhana.

- Kelebihannya adalah GIF mendukung animasi gambar karena sifat gambar yang dihasilkan tidak pecah-pecah, namun memiliki batasan warna yaitu 256.
- Kekurangannya adalah kapasitas penyimpanan pada GIF sangat kecil dibandingkan dengan JPG, oleh karena itu tidak cocok untuk penyimpanan

gambar dengan variasi warna yang banyak. Hanya cocok untuk gambar yang bersifat mempertegas warna saja atau *grayscale*.

3. JPG (*Joint Photographic Expert*)

Joint Photographic Expert merupakan format gambar yang dapat mengurangi kualitas suatu gambar tersebut. Karena pada format gambar ini biasanya gambar akan mengalami pengompresan atau pengurangan ukuran, selain itu jumlah *pixel* pada gambar dengan format ini juga akan berkurang. Format .jpg memiliki ukuran gambar yang kecil, oleh karena itu sering digunakan untuk standar gambar diinternet.

- Kelebihannya adalah format .jpg ini sangat cocok untuk digunakan pada gambar dengan jumlah warna yang banyak atau bervariasi
- Kekurangannya adalah kualitasnya buruk dibandingkan gambar berformat .bmp atau .png. selain itu format gambar ini tidak cocok untuk fotografi karena hasilkan akan menjadi blur

2.7 Resolusi Citra

Resolusi citra merupakan ukuran yang terdapat pada citra. Pada resolusi citra terbagi menjadi dua yaitu ada resolusi spasial dan tingkat bit. Resolusi spasial biasanya untuk menentukan informasi pada suatu gambar. Semakin tinggi resolusi suatu gambar maka semakin jelas informasi yang terkandung pada gambar tersebut. Misalkan saja terdapat sebuah gambar berukuran 625 x 320 yang berarti 625 merepresentasikan jumlah kolom pada citra dan 320 merepresentasikan jumlah baris pada citra tersebut. Kemudian untuk tingkat bit, yaitu merupakan nilai bit pada gambar, biasanya dalam bentuk desimal atau biner yang diubah berdasarkan tabel ASCII.

2.8 Perhitungan PSNR dan MSE

Peak Signal to Noise Ratio (PSNR) merupakan pengukuran untuk membandingkan kualitas gambar yang asli dan gambar terrekonsktruksi (telah disisipi pesan). Selain itu PSNR merupakan pegukuran dari nilai maksimum sinyal yang diukur berdasarkan noise yang terdapat pada gambar. Satuan dari

PSNR adalah desibel (dB). Sebelum mendapatkan nilai PSNR, terlebih dahulu harus mencari nilai MSE (*Mean Square Error*) yaitu merupakan nilai error kuadrat rata-rata antara gambar asli dan gambar yang telah disisipi pesan.

Rumus :

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

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

Keterangan :

- M dan N = Merupakan resolusi gambar
- I dan j = Merupakan koordinat titik gambar
- I = Merupakan gambar yang telah disisipi pesan
- K = Merupakan gambar asli
- MAX_I = Merupakan nilai *pixel* terbesar pada gambar

Tabel 2.3 Skala PSNR

(Sumber : Tyas, 2011)

PSNR (dB)	Kualitas Citra
50 >	Bagus
40-49	Layak
30-39	Cukup
20-29	Tidak Dapat Dipakai

2.9 Visual Basic 6.0

Aplikasi Visual Basic 6.0 atau yang dikenal dengan sebutan VB 6 merupakan aplikasi yang biasa digunakan untuk pengolahan data pada database. Selain itu pada VB 6 ini biasanya digunakan untuk menampilkan grafik atau

diagam batang yang tersambung dengan excel sebagai sumber datanya. Pada VB 6 telah dilengkapi fitur yang terbaik yang dapat mempermudah dalam programming, pembuatan grafik, pengolahan data, dan sebagainya. Biasanya penerapan penggunaan aplikasi ini adalah pada perhitungan barang disebuah gudang, perhitungan pengeluaran perusahaan, perhitungan gaji, dan masih banyak lagi. Untuk penginstallan software VB 6 sendiri, tidak memerlukan memori yang besar pada laptop anda, software ini sangat ringan begitu juga dengan file programnya. Kelemahannya adalah software ini sangat rentan terkena virus, oleh karena itu disarankan untuk file hasil projectnya harus di kompres ke dalam ZIP biar lebih aman.

Gambar 2.5 *Icon* Aplikasi Visual Basic 6.0

BAB 3. METODOLOGI PENELITIAN

Pada bab ini akan menjelaskan tentang waktu dan tempat pelaksaan penelitian, tahapan penelitian, sumber data, serta metode dan algoritma pengumpulan data. Untuk metode pengumpulan data menggunakan Metode *End Of File* sebagai penyisipan file teks kedalam gambar berwarna, serta algoritmanya berupa *Playfair Cipher* sebagai sandi yang membuat file teks tersebut sulit untuk dibuka.

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Jaringan Komputer dan Multimedia, Fakultas Teknik, Universitas Jember yang beralamat di Jalan Slamet Riyadi no. 62 Patrang, Jember. Waktu untuk penelitian lebih kurang 5 bulan, dimulai pada bulan November 2017.

3.2 Tahapan Perancangan Aplikasi

Dalam pembuatan aplikasi penyisipan dan pengekstrakan file teks dibutuhkan langkah - langkah perancangannya, yaitu:

1. Studi Literatur
2. Perancangan desain aplikasi
3. Pembuatan perancangan desain terbagi menjadi 2, yaitu desain penyisipan (*embedding*) dan desain pengekstrakan. Untuk desain penyisipan terdiri dari 4 buah *textbox* yang berfungsi sebagai tempat pengalamatan gambar, file teks dan masukan untuk sandi, serta 4 buah *push button* sebagai tombol untuk memilih gambar dan file teks, menghitung dan mengacak sandi. Kemudian desain pengekstrakan terdiri dari 3 buah *textbox* dan 2 buah *push button* yang memiliki fungsi seperti pada desain penyisipan.
4. Mengisi setiap *push button* dengan program visual basic 6.0 sesuai dengan perintah. Kemudian mengenkripsi antara gambar yang telah tersisipkan file teks menggunakan metode *End Of File* dengan sandi *playfair cipher*.

5. Menganalisa setiap hasil penyisipan pada gambar berformat .bmp, .jpg dan .gif dengan 3 format file teks (.pdf, .doc, dan .txt) yang akan disisipkan ke masing – masing format dan resolusi gambar.

3.3 Alat dan Bahan

Secara umum, alat dan bahan yang dibutuhkan untuk penelitian pada dasarnya terdiri dari perangkat keras dan perangkat lunak. Adapun perangkat keras dan perangkat lunak yang dibutuhkan untuk menjalankan aplikasi penyisipan dan pengekstrakan file teks, yaitu :

1. Laptop.
2. Software Visual Basic 6.0.
3. Rancangan aplikasi untuk penyisipan dan pengekstrakan file teks.
4. Format gambar yang digunakan, yaitu .bmp, .jpg dan .gif.
5. Format file teks yang digunakan, yaitu .doc, .pdf dan .txt.

3.4 Desain Aplikasi

3.4.1 Desain Penyisipan (*Embedding*)

Gambar 3.1 Desain aplikasi penyisipan file teks kedalam gambar berwarna

Dari gambar 3.1 menjelaskan desain aplikasi pada visual basic 6.0. *push button* (*text*) dan *push button* (*image*) memiliki fungsi yang sama yaitu untuk mencari format apa yang akan digunakan, dan nantinya akan menampilkan alamat dari gambar atau file yang telah dipilih. Kemudian push button sandi berfungsi untuk mengacak sandi yang dimasukkan berdasarkan algoritma *playfair cipher*. Dan *push button* embed berfungsi untuk menghitung jumlah *password* yang di

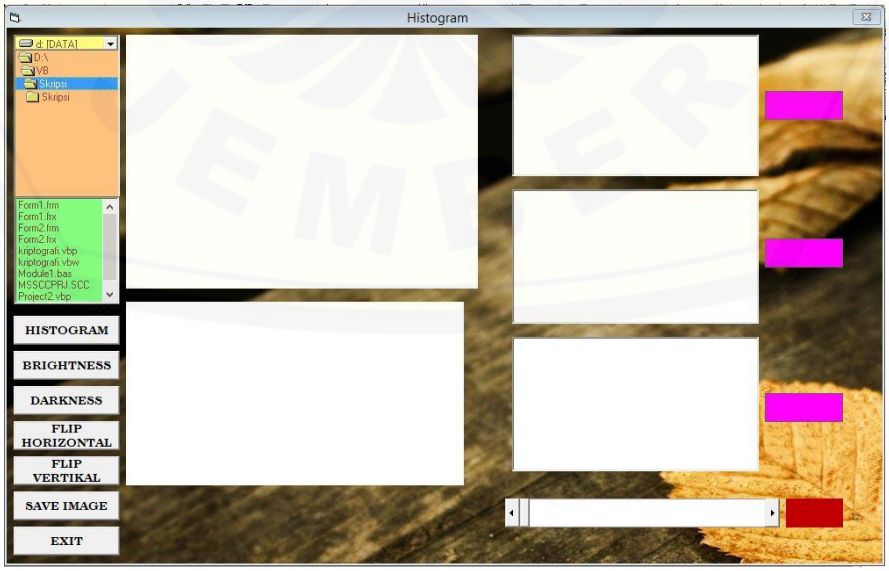
masukkan dan menyisipkan sebuah file kedalam sebuah gambar berwarna berdasarkan metode *End Of File*.

3.4.2 Desain Pengekstrakan

Gambar 3.2 Desain untuk pengekstrakan file teks

Dari gambar 3.2 untuk pengekstrakan datanya, aplikasi ini didesain dengan 3 textbox sebagai tempat untuk menampilkan alamat file atau gambar dan *password*, kemudian terdapat 2 *push button* untuk mencari gambar yang yang telah tersisipkan file dan tombol satunya lagi untuk mengekstrak gambar tersebut, dalam artian memisahkan antara gambar dan file kembali.

3.4.3 Desain Pengujian



Gambar 3.3 Desain pengujian data

Pada gambar 3.3 dapat diketahui bahwa desain pengujian data akan memerlukan memerlukan 5 *push button*, nantinya akan diperlukan untuk Histogram, *Brightness*, *Darkness*, *Save Image* dan *Exit*, kemudian terdapat 5 picture box yaitu *picture* 1 digunakan untuk gambar asli, *picture* 2 digunakan untuk menampilkan grafik *red*, *picture* 3 untuk grafik *green*, *picture* 4 untuk grafik *blue* dan *picture* 5 hanya digunakan saat pengujian *brightness* dan *darkness*. Pada pengujian membandingkan nilai RGB *scale mode* picture 1 menggunakan 1-Twip, jika tidak maka hasil grafik tidak akan muncul, kemudian jika menguji *Brightness* dan *Darkness* picture 1 dan picture 2 *scale mode* nya diubah menjadi 3-Pixel.

Disebelah picture 2, 3, dan 4 terdapat *text box* untuk menampilkan nilai *pixel* dari grafiknya. Kemudian dibawah *picture* 5 terdapat *Horizontal scroll* yang berfungsi untuk menambah dan mengurangi nilai *brightness* dan *darkness*. Selain itu dipojok atas terdapat Dirlist, DriverList, dan Filelist yang memiliki fungsi untuk mencari gambar mana yang akan diuji. Bedanya dengan open yang menggunakan *push button* adalah bahwa *push button* harus mencari datanya satu persatu, sedangkan ketiga komponen tadi langsung menampilkan alamatnya.

3.5 Rancangan Matriks *Password*

N	E	G	A	R
B	C	D	F	H
I	K	L	M	O
P	Q	S	T	U
V	W	X	Y	Z
0	1	2	3	4
5	6	7	8	9

Gambar 3.4 Matrik *password* 7x5

Pada gambar 3.4 merupakan matriks yang akan digunakan untuk *password* dalam menyisipkan file teks kedalam gambar yang terdiri dari huruf kapital dan angka. *Password* tersebut akan diacak dengan menambah 10 berdasarkan huruf

dan angka yang terdapat pada matriks tersebut, ketika angka paling akhir harus dienkripsi maka matriks akan membaca kembali ke awal. Pembuatan *password* dengan algoritma ini harus genap dalam artian setiap membaca *password* akan mengambil 2 huruf dan mengacaknya untuk menghasilkan 2 huruf baru lagi. Jika *password* bernilai ganjil maka akan ditambahkan huruf "Z" diakhirnya, jika terdapat spasi maka akan diganti dengan huruf "Z" dan jika terdapat huruf kembar yang berdempetan maka akan disisipi huruf "Z" diantara huruf kembar tersebut, Misalkan :

Plainkey : POWER SUPPLY

N	E	G	A	R
B	C	D	F	H
I	K	L	M	O
P	Q	S	T	U
V	W	X	Y	Z
0	1	2	3	4
5	6	7	8	9

Gambar 3.5 Matriks *Password*

- Untuk pengambilan *password*-nya adalah

PO WE RS UP ZP LY ⟶ Penambahan huruf Z karena terdapat huruf kembar berdempetan

- Untuk mencari *Cipherkey*

1. PO WE enkripsi menjadi 0Z 6K

N	E	G	A	R
B	C	D	F	H
I	K	L	M	O
P	Q	S	T	U
V	W	X	Y	Z

0	1	2	3	4
5	6	7	8	9

Gambar 3.6 Proses Enkripsi 1

2. RS UP enkripsi menjadi O2 40

N	E	G	A	R
B	C	D	F	H
I	K	L	M	O
P	Q	S	T	U
V	W	X	Y	Z
0	1	2	3	4
5	6	7	8	9

Gambar 3.7 Proses Enkripsi 2

3. ZP LY enkripsi menjadi 90 X8

N	E	G	A	R
B	C	D	F	H
I	K	L	M	O
P	Q	S	T	U
V	W	X	Y	Z
0	1	2	3	4
5	6	7	8	9

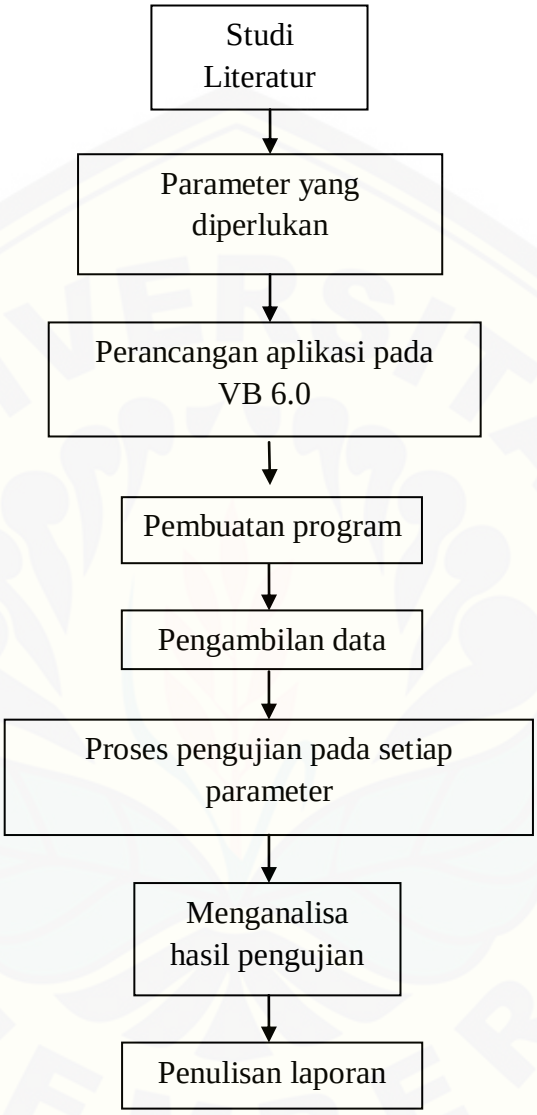
Gambar 3.8 Proses Enkripsi 3

Sehingga :

Plaintext : POWER SUPPLY

Ciphertext : VUECHXZVRVSA

3.6 Diagram alir proses penelitian yang akan dilakukan

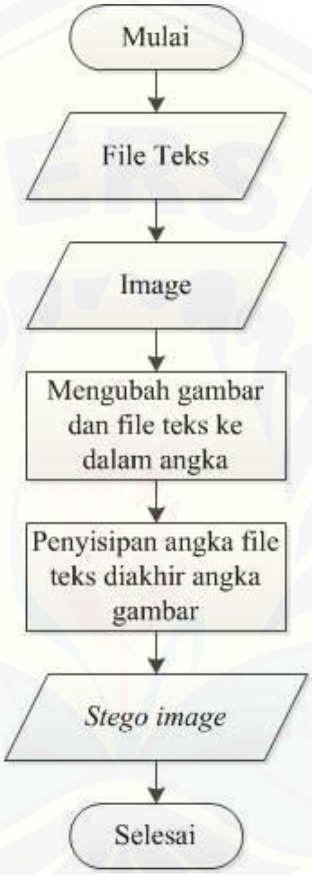


Gambar 3.9 Diagram Alir Penelitian

3.7 Metodologi Pelaksanaan Penelitian

3.7.1 *Flowchart* Metode *End Of File*

Berikut adalah proses Metode *End Of File* yang akan ditunjukkan pada *flowchart* di bawah ini.

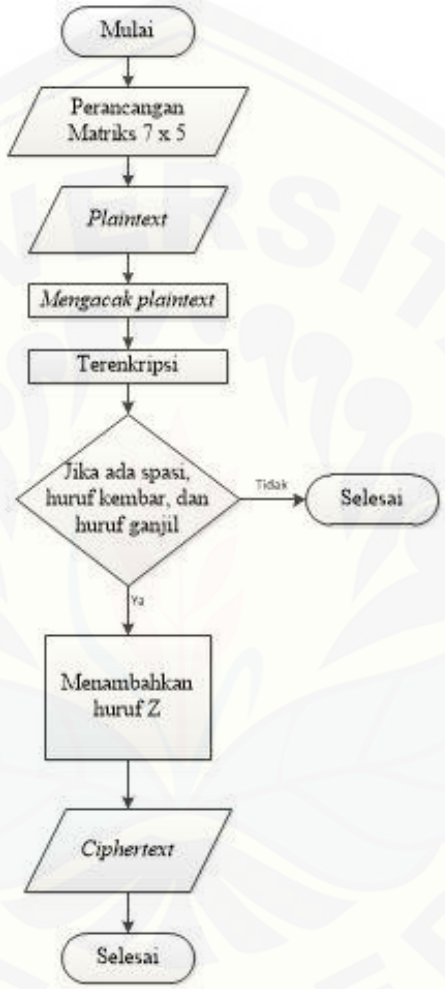


Gambar 3.10 *Flowchart* Metode *End Of File*

Berdasarkan *Flowchart* diatas, dimulai dengan menentukan format file teks yang akan disisipkan kedalam sebuah gambar berwarna, kemudian menentukan format gambar yang digunakan sebagai media penyisipan. Setelah itu melakukan proses untuk mengubah file teks terlebih dahulu kedalam desimal, dilanjutkan dengan mengubah angka kedalam bentuk desimal. Setelah itu melakukan proses penyisipan nilai desimal file teks diakhir nilai akhir desimal gambar. Setelah berhasil maka akan didapatkan *stego image* (gambar yang telah tersisipi file teks).

3.7.2 *Flowchart* Algoritma *Playfair Cipher*

Berikut adalah proses Algoritma *Playfair Cipher* yang akan ditunjukkan pada *flowchart* di bawah ini.

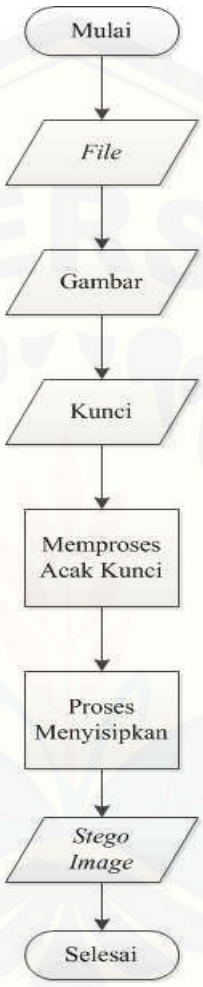


Gambar 3.11 *Flowchart* Algoritma *Playfair Cipher*

Berdasarkan *Flowchart* diatas, dimulai dengan merancang matrik 7x5, kemudian membuat *plaintext*. Selanjutnya melakukan pengacakan pada *plaintext*, maka *plaintext* akan ter-enkripsi, dan proses selanjutnya yaitu, mengetahui plaintext tersebut terdapat spasi dan berjumlah ganjil dan alfabet kembar, jika iya maka ditambahkan huruf "Z"dan mendapatkan *chipertext*, namun jika tidak maka akan langsung selesai.

3.7.3 *Flowchart* Penyisipan File Teks kedalam Gambar

Berikut adalah proses penyisipan file teks ke dalam gambar yang akan ditunjukkan pada *flowchart* di bawah ini.



Gambar 3.12 *Flowchart* Penyisipan File Teks kedalam Gambar

Berdasarkan *Flowchart* diatas, dimulai dengan menentukan format file teks yang akan disisipkan kedalam sebuah gambar berwarna, kemudian menentukan format gambar yang digunakan sebagai media penyisipan. Setelah itu memasukkan *password* yang telah dirancang sebelumnya. Selanjutnya mengacak *password* tersebut, jika berhasil maka file akan tersisipkan dengan *password* yang telah diberikan dan mendapatkan gambar yang yang telah disisipi file teks (*stego image*).

3.7.4 *Flowchart* Pengekstrakan File Teks

Berikut adalah proses pengekstrakan file teks ke dalam gambar yang akan ditunjukkan pada *flowchart* di bawah ini.

Gambar 3.13 *Flowchart* Penyisipan File Teks

Berdasarkan *Flowchart* diatas, dimulai dengan mengambil gambar yang telah disisipi file teks, setelah itu memasukkan *password* yang sesuai dengan *password* saat penyisipan tadi. Jika password sesuai dengan yang sebelumnya maka gambar yang telah disisipi file teks tersebut dapat diekstrak kembali, namun jika tdak sesuai langsung error dan selesai.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian yang telah dilakukan dari penelitian dengan judul "*Steganography* File Teks Pada Citra Menggunakan Metode *End Of File* Dengan Algoritma *Playfair Cipher*" didapatkan beberapa kesimpulan :

1. Dari percobaan yang telah dilakukan bahwa aplikasi *embedding* dan ekstrak tersebut dapat menyisipkan serta mengacak file teks .doc, .pdf dan .txt kedalam gambar .bmp, .gif dan .jpg dengan aman, tanpa menimbulkan kecurigaan pada pihak luar. Selain itu aplikasi ini dapat memisahkan antara gambar dan file teks secara utuh kembali.
2. Ukuran gambar setelah disisipi file teks menjadi bertambah besar, namun kualitas gambar yang telah disisipi file teks tetap sama seperti gambar aslinya. Hal ini dapat dibuktikan pada nilai PSNR yang dihasilkan rata-rata sebesar 99 dB. Kualitas gambar dapat dikatakan layak jika nilai PSNR diatas 40 dB.
3. Setelah melakukan pengujian terhadap gambar yang telah disisipi file teks hasil enkripsi kriptografi *playfair cipher* dengan metode *end of file* bahwa file teks yang disisipi dalam gambar tersebut hilang, karena kriptografi playfair cipher dengan metode end of file ini tidak kebal terhadap pengujian yang dilakukan, sehingga file teks tidak dapat bertahan didalam gambar tersebut.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, maka penulis memberikan saran untuk pengembangan penelitian ini berikutnya, antara lain sebagai berikut :

1. Aplikasi ini dapat dikembangkan dengan menambahkan teknik steganografi dan kriptografi yang lebih baik lagi.
2. Melakukan pengujian yang berbeda.
3. Menggunakan parameter yang berbeda.

DAFTAR PUSTAKA

Andriana, Egi. 2016. *Algoritma Enkripsi Playfair Cipher* dalam ResearchGate. Bandung: UIN Sunan Gunung Djati.

Arifpriyanto, Bely. *Penyembunyian Pesan Text Terenkripsi Menggunakan Metode Kriptografi Stream Cipher dan Steganografi End Of File (EOF) Dengan Induk PDF*. Semarang: Universitas Dian Nuswantoro Fakultas Ilmu Komputer.

Ariyus, Dony. 2006. *Kriptografi*: *Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.

Cogier. 2012. Teknik Menyembunyikan Pesan Dengan Steganografi. https://cogierb201.wordpress.com/2012/05/08/teknik-menyembuyikan-pesan-dengan-steganografi/ [Diakses 10 Juni 2017].

Krisnawati. 2008. *Metode Least Significant Bit (LSB) dan End Of File (EOF) Untuk Menyisipkan Teks Ke Dalam Citra Grayscale* dalam Seminar Nasional Informatika 2008 (semnasIF 2008) UPN "Veteran", ISSN : 1979-2328 Yogyakarta, 24 Mei 2008.

Nurkifli, E. Haodudin. 2014. *Modifikasi Algoritma Playfair dan Menggabungkan Dengan Linear Feedback Shift Register (LFSR)*. Yogyakarta: Seminar Nasional Teknologi Informasi dan Komunikasi 2014 (SENTIKA 2014) ISSN: 2089-9813 Yogyakarta, 15 Maret 2014 .

Simbolon, Ratna Wati. 2016. *Pengamanan Transkrip Nilai Mahasiswa Menggunakan Kriptografi Playfair Cipher dan Steganografi Dengan Teknik Least Significant Bit (LSB)*. Medan: Jurnal Teknologi Informasi dan Komunikasi Vol. 5 No. 1, Juni 2016 : 59 – 70.

The, Gemita Ria. *Studi Perbandingan Steganografi pada Audio, Video, dan Gambar*. Bandung.

Tyas, Lia Ayuning. 2011. *Watermarking Citra DigitalBerbasis DWT-SVD dengan Detektor Non-Blind*. Skripsi. Fakultas Ilmu Pengetahuan Alam. Diponegoro. Semarang.

LAMPIRAN

1. *Listing Program*

   a. Penyisipan dan Pengekstrakan

```
Private Sub Command1_Click()
On Error Resume Next
With CommonDialog1
        .Filter = "Semua File Type (*.*)|*.*"
        .ShowOpen
Text1.Text = .FileName
End With
End Sub

Private Sub Command2_Click()
On Error Resume Next
With CommonDialog1
        .Filter = "Semua File Type (*.*)|*.*"
        .ShowOpen
Text2.Text = .FileName
End With

End Sub

Private Sub Command3_Click()
Text5.Text = Len(Text7.Text)
Dim data As String
Dim X As Long
Dim Y As Long
Dim z As Long
Dim pjg As Long
```

```
Dim ext As String
Dim encpjg As String
encpjg = FileLen(Text2.Text)
ext = Mid(StrReverse(Text2.Text), 1, 4)
ext = StrReverse(ext)
pjg = FileLen(Text1.Text)
FileCopy Text2.Text, Text2.Text & "_EMBED" & ext
X = FileLen(Text1.Text) Mod 10000
Y = FileLen(Text1.Text) - X
Open Text1.Text For Binary Access Read As #1
Open Text2.Text & "_EMBED" & ext For Binary Access
Write As #2
Put #2, FileLen(Text2.Text) + 1, ""
If pjg >= 10000 Then
For z = 1 To Y Step 10000
data = Space$(10000)
Get #1, z, data
Put #2, , encrypt(data, Text7.Text)
Next
Y = X
data = Space$(Y)
Get #1, , data
Put #2, , encrypt(data, Text7.Text)
Put #2, , "|" & encrypt(encpjg, Text7.Text)
Else
data = Space$(pjg)
Get #1, 1, data
Put #2, , encrypt(data, Text7.Text)
Put #2, , "|" & encrypt(encpjg, Text7.Text)
End If
Close #2
```

```
Close #1
MsgBox "Berhasil!!" & vbCrLf & "file ter-embed di " &
Text2.Text & "_EMBED" & ext, vbOKOnly, "embed"
End Sub

Function encrypt(data As String, kunci As String) As
String 'fungsi enkripsi data
Dim i As Double
Dim X As Double
Dim gimmehash As Long
gimmehash = hash(kunci)
Dim enkripsi As String
For i = 1 To Len(data)
X = i Mod Len(kunci)
If X = 0 Then
X = 1
End If
enkripsi = enkripsi & Chr(((Asc(Mid(data, i, 1)) Xor
Asc(Mid(kunci, X, 1)) Xor i) Xor gimmehash) Mod 256)
Next
encrypt = enkripsi
End Function

Private Sub Command4_Click()
On Error Resume Next
Dim ekstrak As String
Dim uncek As String
Dim pmbts As String
Dim pjg2 As Long
Dim z As Long
Dim a As Long
```

```
Dim b As Long
Dim c As Long
Dim d As Long
Dim e As Long
z = 0
uncek = Space$(1)
Open Text4.Text For Binary Access Read As #1
Open Text4.Text & "_EKSTRAK." & Text3.Text For Binary
Access Write As #2
Put #2, 1, ""
While uncek <> "|"
Get #1, FileLen(Text4.Text) - z, uncek
pmbts = uncek & pmbts
z = z + 1
Wend
b = Len(pmbts)
pmbts = Mid(pmbts, 2, Len(pmbts) - 1)
pmbts = encrypt(pmbts, Text6.Text)
z = pmbts
a = FileLen(Text4.Text) - (b + z)
If a >= 10000 Then
c = a Mod 10000
d = (a - c) + z
For e = (z + 1) To d Step 10000
ekstrak = Space$(10000)
Get #1, e, ekstrak
Put #2, , encrypt(ekstrak, Text6.Text)
Next
ekstrak = Space$(c)
Get #1, , ekstrak
Put #2, , encrypt(ekstrak, Text6.Text)
```

```
Else
ekstrak = Space$(a)
Get #1, (z + 1), ekstrak
Put #2, , encrypt(ekstrak, Text6.Text)
End If
Close #2
Close #1
MsgBox "Berhasil!!" & vbCrLf & "file ter-ekstrak",
vbOKOnly, "ekstrak"
End Sub

Private Sub Command5_Click()
Dim concar, SIMPAN, ENDC, LAMA As String
ENDC = ""
KeyAscii = Asc(UCase(Chr(KeyAscii)))
concar = "NEGARBCDFHIKLMOPQSTUVWXYZ0123456789"
For z = 1 To Len(Text5.Text)
For X = 1 To (Len(concar) - 10)

If Mid(Text5.Text, z, 1) = Mid(concar, X, 1) Then
SIMPAN = Mid(concar, X + 10, 1)
End If

Next X

If SIMPAN = LAMA Then
ENDC = ENDC & "Z"
End If

LAMA = SIMPAN
ENDC = ENDC & SIMPAN
```

```
Next z
Text5.Text = ENDC

If Len(Text5.Text) Mod 2 = 1 Then
Text5.Text = Text5.Text & "Z"
End If

Text7.Text = Mid(Text5.Text, 1, Len(Text5.Text))

End Sub
Private Sub Command6_Click()
On Error Resume Next
With CommonDialog1
        .Filter = "Semua File Type (*.*)|*.*"
        .ShowOpen
Text4.Text = .FileName
End With

End Sub

Private Sub Form_Load()
Timer1.Interval = 300
Label1.Caption = "   Embedding And Extract File Teks"
Timer2.Enabled = True
Timer2.Interval = 300
End Sub

Private Sub Form_Terminate()
End
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
End
End Sub


Private Sub keluar_Click()
End
End Sub

Function hash(keys As String) As Long
If keys = "" Then
MsgBox "Silahkan isi key nya dulu", vbCritical,
"error"
End
Else
Dim r As Long
Dim nilai As Long
For r = 1 To Len(keys)
nilai = nilai + Asc(Mid(keys, r, 1))
nilai = nilai Mod Len(keys)
hash = nilai
Next
End If
End Function

Private Sub mnuAboutUsSkripsi_Click()
MsgBox "STEGANOGRAPHY FILE TEKS PADA CITRA MENGGUNAKAN
METODE END OF FILE DENGAN AlGORITMA PLAYFAIR CHIPER" &
vbCrLf & "Skripsi" & vbCrLf & "Oleh" & vbCrLf & "Dwie
Putri Donnaro" & vbCrLf & "141910201001" & vbCrLf &
```

```
"PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO JURUSAN TEKNIK
ELEKTRO FAKULTAS TEKNIK UNIVERSITAS JEMBER 2018"
End Sub

Private Sub mnuEkstrak_Click()
Frame2.Visible = True
Timer4.Enabled = True
End Sub

Private Sub mnuEmbedding_Click()
Timer3.Enabled = True
End Sub

Private Sub mnukeluar_Click()
End
End Sub

Private Sub Timer1_Timer()
Dim kata As String
kata = Label1.Caption
Label1.Caption = Mid(kata, 2) & Left(kata, 1)
End Sub

Private Sub Timer2_Timer()
If Label4.Visible = True Then
Label4.Visible = False
ElseIf Label4.Visible = False Then
Label4.Visible = True
End If
End Sub
```

```
Private Sub Timer3_Timer()
If Frame1.Left > 360 Then
    Frame1.Left = Frame1.Left - 1000
    Frame2.Left = Frame2.Left - 1000
Else
Timer3.Enabled = False
End If

End Sub

Private Sub Timer4_Timer()
If Frame1.Left < 15360 Then
    Frame1.Left = Frame1.Left + 1000
    Frame2.Left = Frame2.Left + 1000
Else
Timer4.Enabled = False
End If
End Sub
```

b. Pengujian

```
Private Sub Command1_Click()
Dim HR(256) As Integer, HG(256) As Integer, HB(256) As
Integer
Dim HT2 As Long
Dim XP As Integer, i As Integer, j As Integer
Dim rr As Integer, gg As Integer, bb As Integer
Dim warna As Long, x As Long, a As Long
Picture2.Cls
Picture3.Cls
Picture4.Cls
Me.MousePointer = vbHourglass
For i = 1 To 256
HR(i) = 0
HG(i) = 0
HB(i) = 0
Next
For i = 1 To Picture1.Width Step 17
For j = 1 To Picture1.Height Step 17
warna = Picture1.Point(i, j)
rr = warna And RGB(255, 0, 0)
gg = Int((warna And RGB(0, 255, 0)) / 256)
bb = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)

If rr > 255 Then rr = 255
If gg > 255 Then gg = 255
If bb > 255 Then bb = 255

HR(rr) = HR(rr) + 1
Text1.Text = HR(rr)
```

```
HG(gg) = HG(gg) + 1
Text2.Text = HG(g)
HB(bb) = HB(bb) + 1
Text3.Text = HB(bb)

Next j
Next i

HT2 = Picture2.Height

For i = 1 To 256
XP = 15 * (i - 1) + 1

Picture2.Line (XP, HT2 - HR(i))-(XP, HT2), RGB(255, 0,
0)
Picture3.Line (XP, HT2 - HG(i))-(XP, HT2), RGB(0, 255,
0)
Picture4.Line (XP, HT2 - HB(i))-(XP, HT2), RGB(0, 0,
255)
Next i

Me.MousePointer = vbNormal
End Sub

Private Sub Command2_Click()
Unload Me
End Sub

Private Sub GetRGB(ByVal Col As String)
On Error Resume Next
Bblue = Col \ (256 ^ 2)
```

```
Ggreen = (Col - Bblue * 256 ^ 2) \ 256
Rred = (Col - Bblue * 256 ^ 2 - Ggreen * 256) '\ 256
End Sub

Private Sub Command3_Click()
Dim i As Long
Dim j As Long

For i = 1 To Picture5.ScaleWidth
For j = 1 To Picture5.ScaleHeight

    warna = Picture1.Point(i, j)

    rr = warna And 255
    gg = Fix(warna / 256) And 255
    bb = Fix(warna / 65536) And 255

    rr = Fix(rr + HScroll1.Value)
    gg = Fix(gg + HScroll1.Value)
    bb = Fix(bb + HScroll1.Value)

    If rr > 255 Then rr = 255
    If rr < 0 Then rr = 0
    If gg > 255 Then gg = 255
    If gg < 0 Then gg = 0
    If bb > 255 Then bb = 255
    If bb < 0 Then bb = 0


    Picture5.PSet (i, j), RGB(rr, gg, bb)
```

```
    Next j
    If x Mod 15 = 0 Then Picture1.Refresh

Next i
Picture1.Refresh
End Sub

Private Sub Command4_Click()
CommonDialog1.CancelError = True
On Error GoTo ja
CommonDialog1.Filter                                    =
"Bitmap|*.bmp|JPG|*.jpg|GIF|*.gif"
CommonDialog1.ShowSave
SavePicture Picture5.Image, CommonDialog1.FileName
Exit Sub
ja:
Exit Sub
End Sub

Private Sub Command6_Click()
Dim i As Long
Dim j As Long

For i = 1 To Picture5.ScaleWidth
For j = 1 To Picture5.ScaleHeight

    warna = Picture1.Point(i, j)

    rr = warna And 255
    gg = Fix(warna / 256) And 255
    bb = Fix(warna / 65536) And 255
```

```
    rr = Fix(r - HScroll1.Value)
    gg = Fix(g - HScroll1.Value)
    bb = Fix(b - HScroll1.Value)

    If rr > 255 Then rr = 255
    If rr < 0 Then rr = 0
    If gg > 255 Then gg = 255
    If gg < 0 Then gg = 0
    If bb > 255 Then bb = 255
    If bb < 0 Then bb = 0

Picture5.PSet (i, j), RGB(rr, gg, bb)

    Next j
    If x Mod 15 = 0 Then Picture1.Refresh

Next i
Picture1.Refresh
End Sub

Private Sub Dir1_Change()
File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
Picture1.Picture = LoadPicture(Dir1.Path & "\" &
File1.FileName)
```

```
End Sub

Private Sub Form_Load()
Drive1.Drive = "D:\"
End Sub

Private Sub HScroll1_Change()
Label1.Caption = HScroll1.Value
End Sub
```

c. MSE dan PSNR

1) MSE

```
%Program for Mean Square Error Calculation

%Author : Athi Narayanan S
%M.E, Embedded Systems,
%K.S.R College of Engineering
%Erode, Tamil Nadu, India.
%http://sites.google.com/site/athisnarayanan/
%s_athi1983@yahoo.co.in

function MSE = MeanSquareError(origImg, distImg)

origImg = double(origImg);
distImg = double(distImg);

[M N] = size(origImg);
error = origImg - distImg;
MSE = sum(sum(error .* error)) / (M * N);
```

2) PSNR

```
%Program for Peak Signal to Noise Ratio Calculation

%Author : Athi Narayanan S
%M.E, Embedded Systems,
%K.S.R College of Engineering
%Erode, Tamil Nadu, India.
%http://sites.google.com/site/athisnarayanan/
%s_athi1983@yahoo.co.in
```

```
function PSNR = PeakSignaltoNoiseRatio(origImg,
distImg)

origImg = double(origImg);
distImg = double(distImg);

[M N] = size(origImg);
error = origImg - distImg;
MSE = sum(sum(error .* error)) / (M * N);

if(MSE > 0)
    PSNR = 10*log(255*255/MSE) / log(10);
else
    PSNR = 99;
end
```

3) Main

```
clc;
clear all;
close all;

%Read Original & Distorted Images
origImg = imread('alpukat.jpg');
distImg = imread('alpukatgelapdoc.jpg');

%If the input image is rgb, convert it to gray image
noOfDim = ndims(origImg);
if(noOfDim == 3)
    origImg = rgb2gray(origImg);
end
```

```
noOfDim = ndims(distImg);
if(noOfDim == 3)
    distImg = rgb2gray(distImg);
end

%Size Validation
origSiz = size(origImg);
distSiz = size(distImg);
sizErr = isequal(origSiz, distSiz);
if(sizErr == 0)
    disp('Error: Original Image & Distorted Image
should be of same dimensions');
    return;
end

%Mean Square Error
MSE = MeanSquareError(origImg, distImg);
disp('Mean Square Error = ');
disp(MSE);

%Peak Signal to Noise Ratio
PSNR = PeakSignaltoNoiseRatio(origImg, distImg);
disp('Peak Signal to Noise Ratio = ');
disp(PSNR);
```