ISBN: 978-602-74635-1-6

REKAYASA PERANGKAT LUNAK PENYANDIAN FILE ALGORITMA VIGENERE CIPHER DAN STEGANOGRAFI DENGAN MENGGUNAKAN METODE INCREMENTAL

Benedictus Effendi[1], Yonathan Salim[2]
[1]Teknik Informatika STMIK PalComTech, Palembang
[2]Sistem Informasi STMIK PalComTech, Palembang
[1,2]Jl. Basuki Rahmat No. 05, Palembang 30129, Indonesia

Abstrak **-** Menjaga keamanan file diperlukan untuk melindungi data penting dan rahasia agar tidak dengan mudah dapat diakses oleh orang-orang yang tidak bertanggung jawab. Penelitian ini bertujuan untuk menerapkan algoritma *vigenere chiper* dan Steganografi untuk membuat rekayasa perangkat lunak sebuah aplikasi penyandian data, dimana manfaat yang dapat diambil yaitu untuk menghasilkan sebuah aplikasi yang dapat digunakan untuk menyandikan *file* dengan tingkat keamanan data yang lebih tinggi. Metode yang digunakan untuk membangun perangkat lunak ini adalah metode pengembangan incremental. Hasil yang diperoleh dalam penelitian ini adalah aplikasi perangkat lunak yang menerapkan algoritma *vigenere chiper* dan steganografi yang menghasilkan penyandian *file* yang disembunyikan.
Kata kunci— kriptografi, *vigenere chiper*, steganografi, enkripsi, *dekripsi, Rekayasa Perangkat Lunak*

1. PENDAHULUAN

Kriptografi merupakan ilmu dan seni untuk menjaga pesan agar aman[1]. Kriptografi terdapat 2 proses yaitu enkripsi (disandikam) dan dekripsi (membuka). Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yag tidak berhak[1]., sedangkan dekripsi adalah proses yang digunakan untuk membaca pesan. Untuk melakukan proses enkripsi dan dekripsi diperlukan sebuah kunci, kunci yang digunakan memberi kekuatan untuk penyandian pesan. Teknik yang dipakai dalam kriptografi bermacam-macam, salah satunya adalah dengan menggunakan *Vigeneree Chiper.*

Penelitian tentang kriptografi pernah dilakukan oleh fairuzabadi dengan mengimplementsikan algoritma kriptografi klasik menggunakan Borland Delphi yang merupakan salah satu bahasa berbasis visual yang penggunaaannya paling luas di dunia akademis[2]. Nurnawati meneliti tentang *vigenere chiper* dimana pada penelitiannya Algoritma *Vigenere Cipher* asli hanya menampung 26 huruf alfabeth dalam bentuk huruf kecil sedangkan tanda baca lain tidak dapat terbaca. Sehingga perlu dilakukan suatu pengevaluasian yaitu dengan memperluas jangkauan 26 huruf alfabeth tersebut menjadi 256 karakter ASCII. Dari pengevaluasian tersebut maka algoritma *Vigenere Cipher* asli tersebut disebut dengan algoritma *Vigenere Cipher* +.[3]

Steganografi adalah metode untuk menyembunyikan informasi pada sebuah media, bisa berupa media gambar, suara ataupun video. Aspek terpenting pada steganografi adalah tingkat keamanan penyembunyian informasinya, yang mengacu pada seberapa besar ketidakmampuan pihak ketiga dalam mendeteksi keberadaan informasi yang tersembunyi. Tujuannya untuk menghindari kecurigaan. Steganografi yang umum digunakan adalah penyembunyian informasi text pada media gambar. Namun metode yang sering digunakan masih cukup sederhana sehingga pihak ketiga masih bisa mendapatkan informasi yang disembunyikan. [4]

Penelitian yang dilakukan oleh Sitorus yaitu mengimplementasikan steganografi text pada media gambar menjadi lebih kuat dan aman. Implementasi yang digunakan adalah mengenkripsi pesan text terlebih dahulu dengan sebuah kata kunci menggunakan algoritma kriptografi. Metode yang dipakai adalah *Least Significant bit insertion* (LSB). Dari hasil uji coba, diketahui bahwa dengan metode Least *Significant Bit Insertion* (LSB) penyisipan dan ekstraksi pesan dapat dilakukan dengan baik. Jenis pesan yang dapat disisipkan adalah pesan *text*. [4]

Algoritma Kriptografi dan Steganografi dapat dikembangkan untuk membuat sebuah perangkat lunak aplikasi, dimana proses penyandian dilakukan dengan perhitungan *Vigenere* yang digabungkan dengan penyembunyian pesan menggunakan *steganografi*. Bahasa pemograman yang dipakai dalam pengembangan aplikasi adalah bahasa pemograman Java dengan menggunakan metode *increamental*.

Penelitian yang dilakukan oleh Fanani dkk tentang metode pengembangan perangkat lunak untuk membuat aplikasi *Use Case Point* adalah metode *incremental* model. Hasil dari penelitihan ini berupa Aplikasi *Use Case Point* melalui 3 kali *increment*. *Increment* pertama ditambahkan fitur estimasi usaha. *Increment* kedua ditambahkan fitur biaya, *increment* ketiga ditambahkan fitur-fitur untuk melakukan kalibrasi perhitungan estimasi. Pengujian terhadap aplikasi dilakukan disetiap *increment* yang ada dengan menggunakan metode pengujian *blackbox testing*/*correctness testing*, *useablity testing*, dan *portability testing*, dan *security testing*. Keluaran dari penelitian ini yaitu berupa Aplikasi *Use Case Point*, dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL),Deskripsi Perancangan Perangkat Lunak (DPPL), dokumen pengujian, dan dokumen panduan.[5]

2. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah deskriptif kuantitatif, data yang digunakan adalah daata sekunder yang diambil dari referensi buku, artikel dan jurnal penelitian terdahulu. Metode pengembangan perangkat lunak yang digunakan adalah metode Incremental yaitu, sedangkan metode penyandian pesan dengan menggunakan algoritma *vigenere chipher* dan penyamaran pesan dengan menggunakan teknik steganografi.

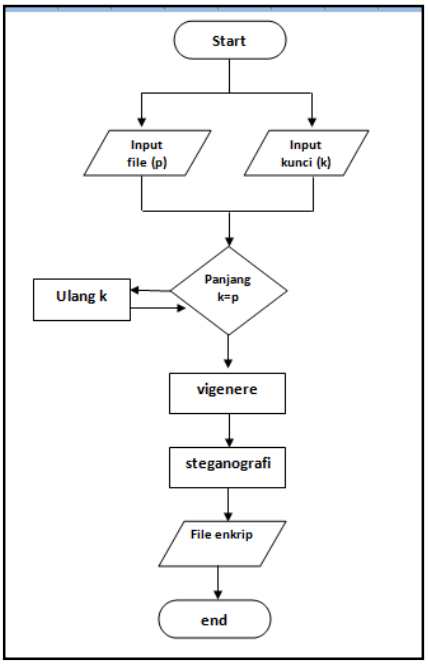
3. HASIL DAN PEMBAHASAN

Rekayasa prangkat lunak aplikasi penyandian file algoritma *vigeneree* dan *steganografi* dengan menggunakan

metode incremental mempunyai tahapan yaitu: *requirment* (analisis kebutuhan), *specification* (proses yang lebih spesifik dengan menggunakan analis kebutuhan), *architecture disign* (perancangaan *software*), *code* (pengkodean), dan *testing* (pengujian).

Requirment (analisis kebutuhan) yang diperlukan pada pembuatan perangkat lunak ini terdiri dari, laptop, aplikasi netbeen, dan *progammer*.

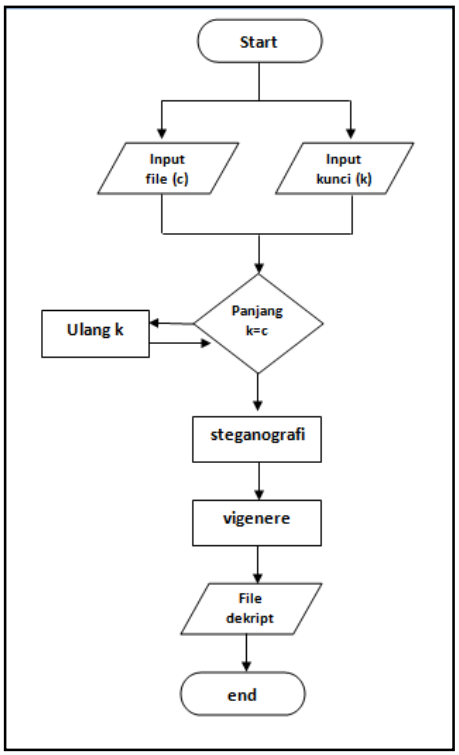
Specification terdiri dari skema proses yang akan dilakukan oleh perangkat lunak yang dibuat. Skema proses yang digunakan untuk proses enkripsi adalah dengan menggunakan 2 inputan awal yaitu input *file* yang akan disandikan dan penginputan kunci.



Gambar 1. Skema proses enkripsi

Proses enkripsi dimulai dengan penginputan *file* yang akan di enkripsikan dan penginputan kunci, panjang kunci yang digunakan pada algoritma *vigenere* harus sama panjang dengan *file* yang di inputkan, jika panjang kunci tidak sama maka aplikasi akan memproses pengulaangan kunci sampai panjang kunci sama dengaan *file plaintcxt*, jika panjang kunci sudah sama maka akan di proses penyandian *file* dengan menggunaakan algoritma *vigenere*, hasil dari *chipertext vigenere* akan di proses penyandian dengan menggunakan steganografi dan menghasilkan *chipertext file.*

Skema proses yang digunakan untuk proses dekripsi adalah dengan penginputan kunci yang sama dengan kunci pada proses enkripsi dan menginputkan *file* chipertext-nya.



Gambar 2. Skema proses dekripsi

Proses dekripsi dimulai dengan penginputan *file* yang akan di dekripsikan dan penginputan kunci, panjang kunci yang digunakan harus sama panjang dengan *file* yang diinputkan, jika panjang kunci tidak sama maka aplikasi akan memproses pengulaangan kunci sampai panjang kunci sama dengan *file chipertext*, jika panjang kunci sudah sama maka akan di proses dengan menggunakan teknik steganografi hasil dari *teknik steganografi* akan di proses pembacaan dengan menggunakan algoritma *vigenere* dan menghasilkan *plaintext file.*

Tahapan selanjutnya adalah peng-code-an dimana pada penelitian ini code yang digunakan adalah bahasa pemograman *java* dengan menggunakan *netbeen.*

Perhitungan enkripsi *Vigenere* menggunakan rumus:

$Ci = (Pi + Ki) \bmod 26 \quad (1)$

Sedangkan untuk rumus dekripsi *Vigenere* Cipher:

$Pi = (Ci – Ki) \bmod 26 \quad (2)$

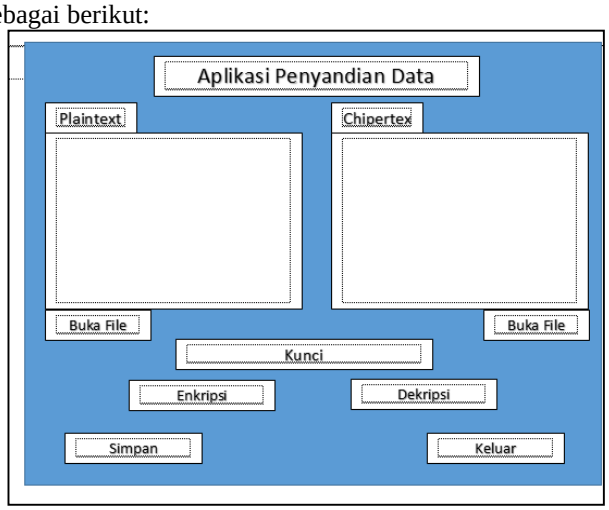
Dimana :
Ci= cipher teks
Pi = plainteks
Ki = kunci

Perhitungan *vigenere* juga dapat menggunakan bujursangkar *Vigenere* untuk melakukan enkripsi. Setiap baris di dalam bujursangkar menyatakan huruf-huruf *ciphertext* yang diperoleh dengan Caesar cipher. Untuk lebih jelasnya perhatikan gambar 3 di bawah ini. Deretan huruf mendatar menunjukkan *plaintext*, sedangkan huruf menurun menunjukkan kunci.

Gambar 3. Tabel *Vigenere*

Proses Steganografi digunakan saat plaintext sudah terenkripsi oleh algoritma *vigeneree.* Untuk menghasilkan steganografi yang baik ada 3 kriteria yang harus diperhatikan [6], yaitu : 1. *Imperceptibility*. Keberadaaan pesan rahasia tidak bisa dikenali oleh indra manusia. Misalnya, jika *convertext* berupa citra maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra covertextnya. 2. *Fidelity*. Mutu stegomedium tidak berubah banyak akibat penyisipan. Misalnya, jika *convertext* berupa citra maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra covertextnya. 3. *Recovery*. Pesan yang disembunyikan harus dapat dikenali kembali. Karena tujuan steganografi adalah data *hiding*, maka sewaktu-waktu pesan rahasia di dalam *stegotext* harus dapat diambil kembali untuk digunakan.

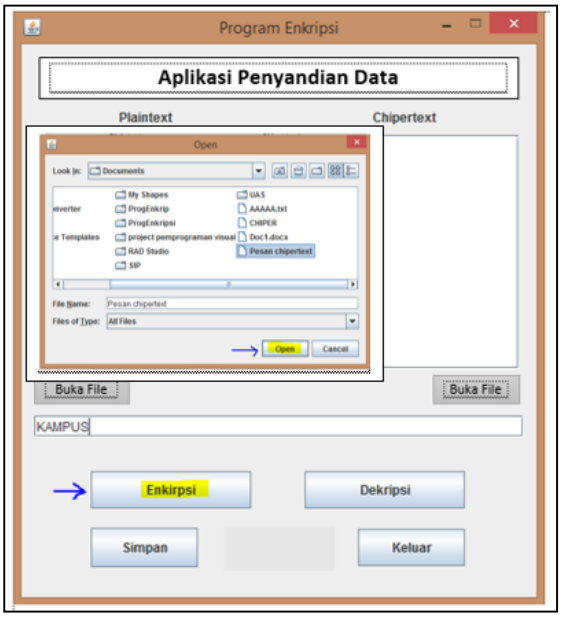
Tahapan *architecture disign* pada aplikasi ini adalah sebagai berikut:



Gambar 4. Desain awal aplikasi

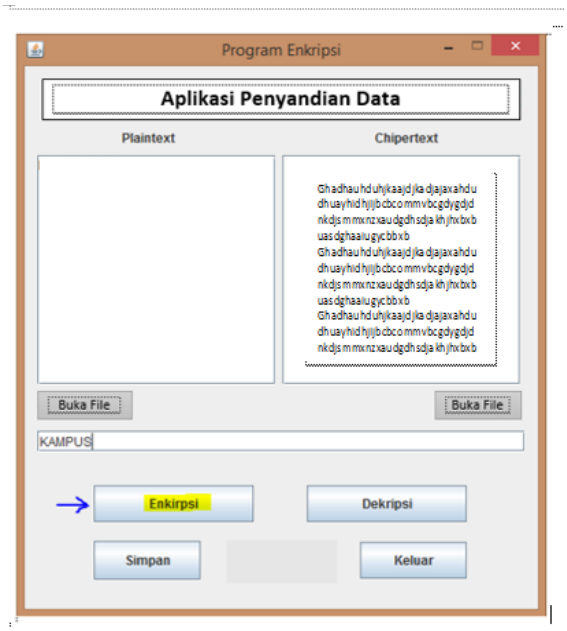
Design dari aplikasi ini terdapat inputan *plaintext* dan *chipertext* yang berfungsi untuk tempat *input/open file* yang akan di proses, kemudian inputan kunci yang digunakan untuk penyandian. *Enkripsi* dan *dekripsi* adalah *button* yang digunakan untuk proses enkripsi ataupun dekripsi.

Tahapan selanjutnya adalah *testing* perangkat lunak, hasil dari uji coba perangkat lunak adalah sebagai berikut:



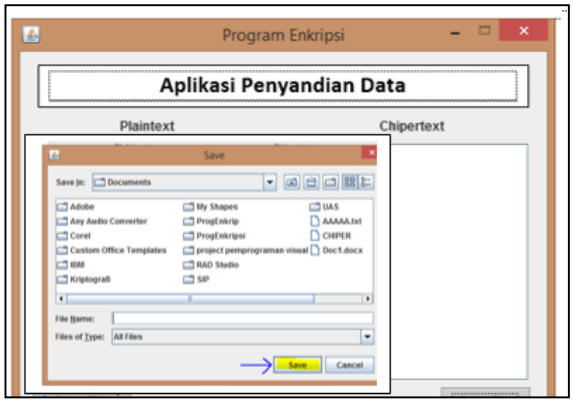
Gambar 5. Proses *enkripsi*

Proses *enkripsi* dilakukan dengan cara *input file* ataupun dengan buka *file*, kemudian mencari lokasi *file* yang akan disandikan. Kemudian masukkan kunci yang akan digunakan untuk menyandikan pesan dan pilih proses enkripsi, hasil chipertextnya dapat dilihat pada gambar 6 berikut:



Gambar 6. Hasil *Chipertext*

Hasil Chipertext dapat disimpan pada folder komputer dengan format file yang sama dengan file yang disandikan, proses penyimpanan file dapat dilihat pada gambar 7 berikut ini:



Gambar 7. Proses *save file*

Alur proses dekripsi pada dasarnya sama dengan proses enkripsi hanya saja *file* yang di inputkan pada *chipertext* adalah hasil *file* yang sudah terenkripkan.

Proses enkripsi dan dekripsi pada aplikasi penyandian ini diujikan pada 4 *file* dengan ukuran yang berbeda, hasil yang di dapatkan dapat dilihat pada tabel 1 berikut:

Tabel 1. Hasil Percobaan Penyandian

No	*Text file*	File Steganografi	*File* enkripsi	*File* dekripsi
1	143 kb	237 kb	380kb	143 kb
2	567 kb	237 kb	804 kb	567 kb
3	41 kb	237 kb	278 kb	41 kb
4	87 kb	237 kb	324 kb	87 kb

Percobaan pertama dilakukan pada file .docx dengan ukuran 143 kb yang disisipkan pada *file* .txt yang merupakan *file* steganografi untuk menyamarkan pesan, hasil yang didapat adalah *file* dalam bentuk .docx dengan ukuran gabungan antara *file* .docx dengan *file* .txt.

Hasil yang sama juga didapat dari ketiga pecobaan selanjutnya. Hasil enkripsi merupakan file yang sama dengan *file* awal yang akan disandikan hanya berbeda pada ukuran *file*, dengan demikian hasil penyandian data tersebut tidak dicurigai bahwa file yang telah terenkripsi mengandung file yang telah disamarkan oleh file steganografi.

4. KESIMPULAN

Penelitian tentang rekayasa perangkat lunak aplikasi penyandian data ini menghasilkan sebuah aplikasi yang dapat digunakan untuk menyandikan file- file untuk dapat membantu mengamankan data penting yang ingin kita lindungi.

DAFTAR PUSTAKA

[1] Sasongko, Jati. 2005. Pengamanan Data dan Informasi menggunakan Kriptografi Klasik.Jurnal Teknologi Informasi DINAMIK, Vol. X No. 3.

[2] Fairuzabadi, Muhammad. 2010. Implementasi Kriptografi Klasik menggunakan Borland Delphi. Jurnal Dinamika Informatika, Vol. 4 No.2.

[3] Nurnawati, Erna Kumalasari. 2008. Analisis Kriptografi dengan menggunakan Algoritma Vigenere Chiper dengan Mode Operasi Chiper Block Chainning (CBC). Seminar Nasional Aplikasi dan SAINS IST AKPRIND Yogyakarta.

[4] Sitorus, michael. 2015. Teknik Steganografy dengan Metode Least Significant Bit (LSB). Jurnal Ilmiah Fakultas Teknik LIMIT'S, Vol. 11 No. 2

[5] Fanani, Mukhamad faiz, dkk. 2015. Seminar Nasional Sistem Informasi Indonesia.

[6] Renaldi Munir. 2006. Diktat Kuliah Studi Teknik Informatika, Institut Teknologi Bandung.