IMPLEMENTASI KRIPTOGRAFI UNTUK KEMANAN DATA MENGGUNAKAN METODE *VIGENERE CIPHER*

TUGAS AKHIR

Disusun Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana

Jurusan Teknik Informatika

Oleh :

Nama : Bimaji Hernowo

NIM : 08523074

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

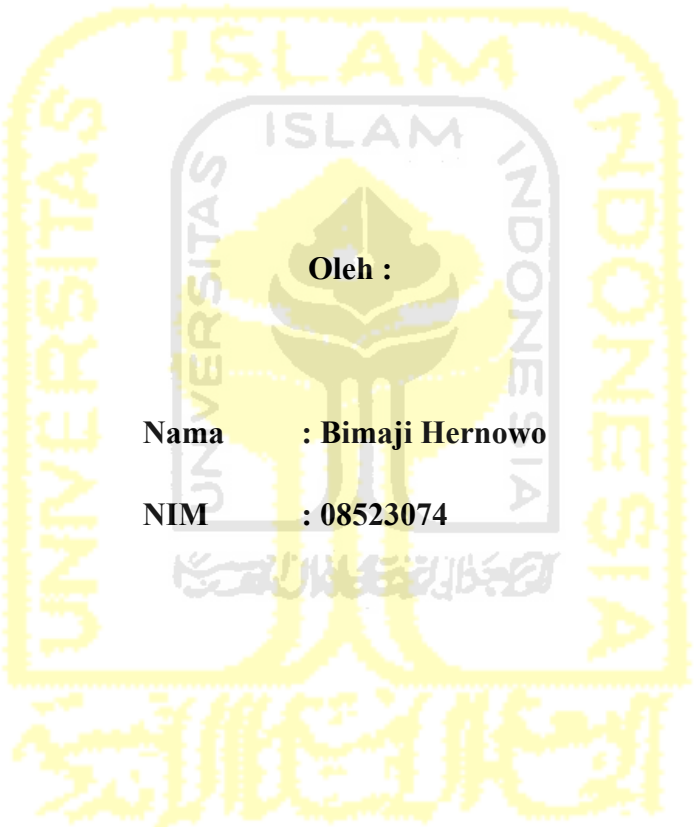
YOGYAKARTA

2012

LEMBAR PENGESAHAN PEMBIMBING

IMPLEMENTASI KRIPTOGRAFI UNTUK KEAMANAN DATA MENGGUNAKAN METODE *VIGENERE CIPHER*

TUGAS AKHIR



Oleh :

Nama : Bimaji Hernowo

NIM : 08523074

Yogyakarta, April 2012

Pembimbing,

Syarif Hidayat, S.Kom., M.I.T

LEMBAR PENGESAHAN KEASLIAN

HASIL TUGAS AKHIR

Saya yang bertanda tangan di bawah ini,

Nama : Bimaji Hernowo

No. Mahasiswa : 08523074

Menyatakan bahwa seluruh komponen dan isi dalam laporan tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbkti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, April 2012

Bimaji Hernowo

LEMBAR PENGESAHAN PENGUJI

IMPLEMENTASI KRIPTOGRAFI UNTUK KEAMANAN DATA MENGGUNAKAN METODE *VIGENERE CIPHER*

TUGAS AKHIR

Oleh :

Nama : Bimaji Hernowo

NIM : 08523074

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika Teknologi Industri Universitas Islam Indonesia

Syarif Hidayat, S.Kom., M.I.T _____

Ketua

R. Teduh Dirgahayu, S.T.,M.Sc,P.Hd _____

Anggota I

Affan Mahtarami, S.Kom., M.T. _____

Anggota II

Mengetahui

Ketua Jurusan Teknik Informatika

Universitas Islam Indonesia

Yudi Prayudi, S.Si,. M.Kom.

PERSEMBAHAN

Tuhanku, terima kasih untuk semua rencana besarmu,

Engkau memang nyata.

Bapak, Ibu, Mbak Wik, Bina.. Semoga ini bisa menjadi persembahan awalku untuk kalian, dan akan selalu aku beri segala yang terbaik untuk kalian.

Andin, jangan nakal, sekolah yg rajin.

Teman-teman terbaikku, Dini, Zaim, Adi, Sigit, Edi.

Affairs store crew, Mas Mario, Krista, Ocha, Sigit, Iboy, Dimas, Haris.

Tendos, who was truly and immensely inspiring to me.

Terimakasih untuk semua dukungan kalian.

YOU ROCK !

MOTTO

Hidup ga cukup hanya lihat ke bawah, hidup juga perlu lihat ke atas. Masih banyak orang yang ga seberuntung kita, dan ada banyak juga orang yang lebih beruntung daripada kita.

Love is a journey, not a destination.

I had to learn what I have got and what I am not.

Hidup tanpa target, kenapa tidak ?

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji dan syukur sudah selayaknya dipanjatkan kepada Allah yang telah menjadikan manusia beriman dan berilmu setelah sebelumnya berada dalam kondisi yang lemah dan diliputi kebodohan. Atas izin dan kehendak-Nya, tugas akhir berjudul "Implementasi Kriptografi Untuk Keamanan Data Menggunakan Metode Vigenere Cipher" ini akhirnya dapat terselesaikan. Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika di Universitas Islam Indonesia.

Dalam penulisan tugas akhir ini tidak lepas dari saran-saran, bimbingan, dukungan, serta bantuan dari berbagai pihak. Untuk itu, pada kesempatan ini penulis ingin mengucapkan terimakasih kepada :

1. Kedua orang tua, yang tak pernah putus memberikan dorongan, semangat, dan doanya.
2. Bapak Ir. Gumbolo HS, M.Sc., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Yudi Prayudi, S.Si, M.Kom., selaku Ketua Jurusan Teknik Informatika, Universitas Islam Indonesia.
4. Bapak Syarif Hidayat S.Kom., M.I.T., selaku Dosen Pembimbing Tugas Akhir. Terima kasih atas segala bantuan, dukungan, bimbingan, arahan dan pengetahuan yang telah diberikan kepada penulis dalam penyusunan skripsi ini.
5. Semua pihak yang tidak dapat disebutkan satu persatu.

Dalam penyelesaian tugas akhir ini penulis menyadari bahwa masih banyak terdapat kelemahan dan kekurangan. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun agar pada masa mendatang menjadi lebih baik.

Akhir kata semoga tugas akhir ini dapat berguna bagi para penuntun ilmu, para praktisi, dan seluruh masyarakat IT untuk tujuan kemaslahatan dan kepentingan bersama.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 10 April 2012

Penulis

ABSTRAKSI

Dengan semakin berkembangnya pemanfaatan teknologi informasi dalam membantu pekerjaan manusia di berbagai jenis kegiatan yang melibatkan komputer sebagai medianya, maka keamanan menjadi aspek yang sangat penting dalam sistem informasi. Beberapa informasi umumnya hanya ditujukan bagi segolongan orang tertentu, oleh karena itu keamanan data sangat dibutuhkan untuk mencegah informasi tersebut sampai pada pihak – pihak lain yang tidak berkepentingan.Teknik kriptografi dapat dimanfaatkan untuk menjamin keamanan dokumen elektronis. Salah satu yang dapat dimanfaatkan adalah enkripsi dan dekripsi data atau dengan kata lain menyandikan data sehingga hanya orang yang bersangkutan saja yang dapat mengetahui isi data tersebut.

Dalam penelitian ini, sistem dibangun menggunakan metode vigenere cipher**,** *Vigenere cipher* merupakan salah satu jenis algoritma klasik yang populer dan sering digunakan sebagai metode penyembunyian pesan (kriptografi). *Vigenere cipher* ini menggunakan teknik substitusi dalam pengenkripsian pesannya dimana setiap karakter plainteks pada pesan akan dienkripsi menjadi karakter lain pada cipherteks berdasarkan kunci yang digunakan. Algoritma ini termasuk ke dalam jenis cipher abjad majemuk atau lebih sering disebut sebagai *polyalphabetic subtitution cipher*. Algoritma ini merupakan bentuk pengembangan dari Caesar Cipher yang juga menggunakan metode substitusi karakter untuk melakukan enkripsi pesan. Tujuan utama dari algoritma enkripsi *Vigenere cipher* ini adalah untuk meminimalkan keterhubungan antara karakter plainteks dan karakter cipherteks yang merupakan kelemahan dari jenis substitusi alfabet tunggal seperti Caesar Cipher.

Hasil penelitian menunjukkan bahwa telah berhasil dibangun sebuah aplikasi kriptografi dengan menggunakan metode vigenere cipher dengan proses enkripsi dan dekripsi yang dilakukan per satuan *kilobyte* data. Aplikasi ini mampu menangani enkripsi dan dekripsi file gambar dan file teks.

Kata kunci : *Vigenere cipher,* enkripsi, dekripsi

TAKARIR

Array	: Kumpulan data yang dikelompokkan dalam satu variabel
Binary	: Biner
Ciphertext	: Data tersandi
Combo box	: Kotak dialog yang berisi kombinasi kontrol
Confidentiality	: Kerahasiaan
Data flow diagram	: Bentuk perancangan sistem
Default	: Keadaan awal
Dir list box	: Daftar direktori
Drive list box	: Daftar perangkat
File	: Data
File list box	: Daftar data
Flowchart	: Diagram alir
Input	: Masukan
Kilobyte	: Satuan memori, setara dengan 1024 byte
Listing code	: Baris kode bahasa pemrograman
Load	: Memuat
Message box	: Kotak pesan
Output	: Keluaran
Plaintext	: Data asli

Progress bar : Ilustrasi status proses yang sedang berjalan

Software developer : Pengembang perangkat lunak

Type : Tipe

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING ii
LEMBAR PERNYATAAN KEASLIAN iii
LEMBAR PENGESAHAN PENGUJI iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTTO vi
KATA PENGANTAR vii
ABSTRAKSI ix
TAKARIR x
DAFTAR ISI xii
DAFTAR GAMBAR xiv

BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Rumusan Masalah 2
1.3. Batasan Masalah 2
1.4. Tujuan Penelitian 3
1.5. Manfaat Penelitian 3
1.6. Metodologi Penelitian 3
1.7. Sistematika Penulisan 4

BAB II LANDASAN TEORI 6
2.1. Kriptografi 6
2.2. Steganografi 11
2.3. Algoritma Kriptografi Klasik 12
2.4. Vigenere Cipher 13
2.4.1. Teknik Substitusi Angka 13
2.4.2. Teknik Substitusi Huruf 15

BAB III METODOLOGI 19
3.1. Analisis Sistem 19
3.2. Kebutuhan Perangkat 19
3.3. Kebutuhan Input 20
3.4. Kebutuhan Proses 20
3.5. Kebutuhan Output 20
3.6. Data Flow Diagram (DFD) 21
3.7. Algoritma dan Flowchart 21
3.4. Perancangan Tampilan Program 25

BAB IV HASIL DAN PEMBAHASAN 28
4.1. Implementasi Sistem 28
4.2. Analisis Kerja Sistem 33
4.3. Pengujian Sistem 39

BAB V SIMPULAN DAN SARAN 58
5.1. Kesimpulan 58
5.2. Saran 58

DAFTAR PUSTAKA

DAFTAR GAMBAR

Gambar 2.1. Skema enkripsi dan dekripsi menggunakan kunci 8

Gambar 2.2. Bagan hubungan antara kriptografi dengan kriptanalisis 9

Gambar 2.3. Bujur sangkar vigenere 14

Gambar 2.4. Penggunaan bujur sangkar vigenere 16

Gambar 3.1 *Data Flow Diagram* sistem 20

Gambar 3.2 *Flowchart* proses enkripsi 22

Gambar 3.3 *Flowchart* proses dekripsi 23

Gambar 3.4 Rancangan antarmuka menu utama 24

Gambar 3.5 Rancangan antarmuka menu enkripsi 25

Gambar 3.6 Rancangan antarmuka menu dekripsi 25

Gambar 3.7 Rancangan antarmuka konfirmasi membuka file 26

Gambar 4.1 Halaman utama sistem 33

Gambar 4.2 Plainteks gambar 34

Gambar 4.3 Halaman enkripsi file 35

Gambar 4.4 Halaman jelajah 35

Gambar 4.5 Halaman enkripsi dengan tampilan hasil enkripsi 36

Gambar 4.6 Notifikasi proses enkripsi selesai 37

Gambar 4.7 Perbandingan plainteks dan cipherteks 37

Gambar 4.8 Halaman dekripsi file 38

Gambar 4.9 Konfirmasi proses dekripsi 38

Gambar 4.10 Tampilan hasil dekripsi dengan kunci yang salah 39

Gambar 4.11 Pengujian enkripsi menggunakan sistem..... 49
Gambar 4.12 Pengujian dekripsi menggunakan sistem..... 50
Gambar 4.13 Pesan kesalahan kata kunci belum terisi 50
Gambar 4.14 Pesan kesalahan duplikasi nama file 51
Gambar 4.15 Tampilan kelebihan karakter 52
Gambar 4.16 Tampilan pengurangan kelebihan karakter 53
Gambar 4.17 Ilustrasi pengujian dengan steganografi 53
Gambar 4.18 Tampilan aplikasi steganografi..... 54
Gambar 4.19 Perbandingan ukuran file sebelum dan sesudah steganografi 54
Gambar 4.20 Perbandingan file sebelum dan sesudah didekripsi 55
Gambar 4.21 Tampilan pengujian file steganografi setelah dekripsi 55
Gambar 4.22 Grafik kebutuhan waktu enkripsi 56
Gambar 4.23 Grafik kebutuhan waktu dekripsi 57

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini penggunaan komputer untuk pengiriman data melalui saluran komunikasi sudah merupakan hal yang jamak. Namun sekarang ini banyak orang yang tidak bertanggung jawab dengan melakukan sabotase terhadap pengiriman data melalui jaringan. Hal ini mengakibatkan perlu adanya tingkat keamanan yang lebih baik dalam pengiriman dala melalui jaringan. Untuk mencegah adanya penyadapan data pada waktu pengiriman, digunakanlah teknik kriptografi untuk menyandikan data, yaitu dengan cara mengubah data asli (*plaintext*) menjadi data yang tersandi (*ciphertext*) yang isinya berbeda dari dari data aslinya. Dengan teknik kriptografi yang menggunakan proses enkripsi dan dekripsi, maka suatu data dapat diubah ke bentuk yang tidak dimengerti oleh orang awam dan dapat dikembalikan lagi ke bentuk data semula.

Salah satu aspek keamanan yang perlu dijamin dalam suatu data, baik itu konvensional maupun digital, adalah kerahasiaannya (*confidentiality*). Kerahasiaan merupakan aspek yang digunakan untuk menjaga informasi dari semua pihak yang tidak memiliki kewenangan untuk mengaksesnya. Dengan demikian maka informasi hanya akan dapat diakses oleh pihak-pihak yang berhak saja.

Cukup banyak metode yang dapat digunakan dalam teknik kriptografi, baik itu klasik maupun modern. Untuk metode kriptografi klasik diantaranya adalah Caesar Cipher, Kode Geser, Hill Cipher, Vignere Cipher, Playfair Cipher, Stream Cipher. Sedangkan untuk kriptografi modern diantaranya RSA, Algoritma ElGamal, Algoritma A5, GOST, Blowfish, Algoritma Twofish.

Dari latar belakang untuk permasalahan diatas, penulis bermaksud untuk membangun dan mengimplementasikan teknik kriptografi untuk memenuhi kebutuhan keamanan data dengan menggunakan metode Vigenere Cipher yang merupakan salah satu metode kriptografi klasik dengan karakteristik cipher abjad majemuk atau lebih sering disebut sebagai *polyalphabetic substitution cipher* yang cukup mudah untuk dipelajari, dipahami, dan dikembangkan.

1.2 Rumusan Masalah

Dalam pelaksanaan penelitian tugas akhir ini terdapat beberapa permasalahan yang menjadi titik utama pembahasan, diantaranya adalah sebagai berikut:

1. Membangun sistem yang dapat menjaga kerahasiaan data menggunakan algoritma Vigenere Cipher menggunakan pemrograman Visual basic 6.0.
2. Bagaimana metode enkripsian dan dekripsi file yang digunakan pada algoritma Vigenere Cipher untuk menyamarkan data aslinya.

1.3 Batasan Masalah

Dari rumusan masalah yang ada maka dapat diberikan batasan-batasan sehingga pembahasannya lebih terarah. Adapun batasan-batasan masalah yang menjadi acuan dalam tugas akhir ini adalah sebagai berikut :

1. Aplikasi melakukan enkripsi data teks (txt, rtf), dan data gambar (jpeg, bmp, png, gif) dengan menggunakan metode Vigenere Cipher dengan menggunakan kunci tertentu.
2. Proses enkripsi dilakukan dengan menggunakan 256 karakter yang ada dalam ASCII untuk memperluas cakupan jumlah karakter yang dapat dipakai.

1.4 Tujuan Penelitian

Tujuan yang hendak dicapai dari penelitian ini adalah :

1. Menganalisis bagaimana cara kerja algoritma Vigenere Cipher dalam memberikan layanan kerahasiaan data
2. Membangun suatu program yang dapat digunakan untuk enkripsi dan dekripsi data dengan menggunakan algoritma Vigenere Cipher.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Memenuhi kebutuhan akan keamanan suatu data, sehingga menjamin kerahasiaan, integritas data, dan otentikasi data itu sendiri
2. Permasalahan terhadap keamanan pada komunikasi data antar jaringan dapat diatasi
3. Dapat menjadi wacana baru dalam hal perbaikan terhadap metode kriptografi dan bisa memberikan wawasan untuk pengembangan penelitian selanjutnya

1.6 Metodologi Penelitian

Metodologi penyusunan tugas akhir ini merupakan langkah-langkah kerja yang perlu dilakukan agar penyusunan tugas akhir menjadi lebih mudah dan terarah. Metodologi yang digunakan dalam penyusunan tugas akhir ini adalah :

1. Analisis masalah

   Metode ini untuk menganalisis masalah-masalah yang telah terjadi, yang menyebabkan mengapa dilakukan penelitian ini

2. Analisis kebutuhan

   Metode untuk menganalisis kebutuhan sistem aplikasi antara lain dari spesifikasi perangkat keras, analisis perangkat lunak meliputi input, output, dan langkah-langkah yang dibutuhkan serta sistem antarmuka yang akan digunakan

3. Perancangan

   Metode perancangan berisi tentang rancangan dari alur sistem dan perancangan antarmukanya.

4. Impelementasi

   Metode implementasi menerapkan hasil rancang yang telah disetujui terhadap aplikasi yang akan dibuat

5. Pengujian

   Metode pengujian dilakukan untuk menguji aplikasi yang telah selesai dibuat sesuai rancangan sebelum aplikasi tersebut digunakan

1.7 Sistematika Penulisan

Dalam penyusunan laporan tugas akhir ini dibagi menjadi beberapa bab pokok. Adapun gambaran secara umum sebagai berikut :

BAB I Pendahuluan, berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, sistematika penulisan.

BAB II Landasan Teori, membahas masalah yang berhubungan dengan perancangan aplikasi kriptografi untuk keamanan data serta teori lainnya yang mendukung pembuatan aplikasi.

BAB III Metodologi, berisi tahapan-tahapan dalam perancangan sistem yang meliputi analisis kebutuhan sistem, perancangan alur sistem, dan perancangan tampilan sistem.

BAB V Hasil dan Pembahasan, berisi implementasi dari sistem yang dibangun, analisis kinerja sistem, pengujian sistem, serta evaluasi terhadap kesesuaian sistem dengan kebutuhan dan hasil yang diharapkan.

BAB VI Simpulan dan Saran, berisi kesimpulan yang diperoleh selama perancangan aplikasi dan saran yang diberikan untuk pengembangan aplikasi lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, crypto dan graphia. *Crypto* berarti *secret* (rahasia) dan *graphia* yang berarti *writing* (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. (Dony Ariyus, 2008, h.13)

Dalam perkembangannya, kriptografi juga digunakan untuk mengidentifikasi pengiriman pesan dan tanda tangan digital dan keaslian pesan dengan sidik jari digital.

Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminologi. Beberapa istilah yang harus diketahui yaitu :

1. Pesan, plaintext, dan ciphertext

   Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks *(plaintext)* atau teks-jelas (*clear text*). Agar pesan tidak dapat dimengerti maknanya oleh pihak lain yang tidak berkepentingan, maka pesan perlu disandikan kebentuk lain yang tidak dapat dipahami.bentuk pesan yang tersandi disebut cipherteks *(ciphertext)* atau kriptogram (*cryptogram*). Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar dapat diterima dan bisa dibaca.

2. Pengirim dan penerima

   Komunikasi data melibatkan pertukaran pesan antar dua entitas. Pengitim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*sender*) adalah entitas yang menerima pesan. Pengirim tentu menginginkan pesan dapat

dikirim secara aman, yaitu pengirim yakin bahwa pihak lain tidak dapat membaca isi pesan yang dikirim. Solusinya adalah dengan cara menyandikan pesan menjadi *ciphertext.*

3. Enkripsi dan dekripsi

   Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering.* Sedangkan proses mengembalikan cipherteks menjadi plainteks disebut dekripsi (*decryption*) atau *deciphering.*

4. Cipher dan kunci

   Algoritma kriptografi disebut juga cipher, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk *enciphering* dan *deciphering.*

   Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi cipherteks. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan *P* menyatakan plainteks dan *C* menyatakan cipherteks, maka fungsi enkripsi *E* memetakan *P* ke *C.*

   $$E(P)=C$$

   Dan fungsi dekripsi D memetakan C ke P

   $$D(C)=P$$

   Karena proses enkripsi kemudian dekripsi mengembalikan pesan kepesan asal, maka kesamaan berikut harus benar ,

   $$D(E(P))=P$$

   Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan

untuk transformasi *enciphering* dan *deciphering*. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan *K*, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai berikut :

$$E_K(P)=C \text{ dan } D_K(C)=P$$

Dan kedua fungsi ini memenuhi

$$D_K(E_K(P))=P$$

Keterangan :

P = Plainteks

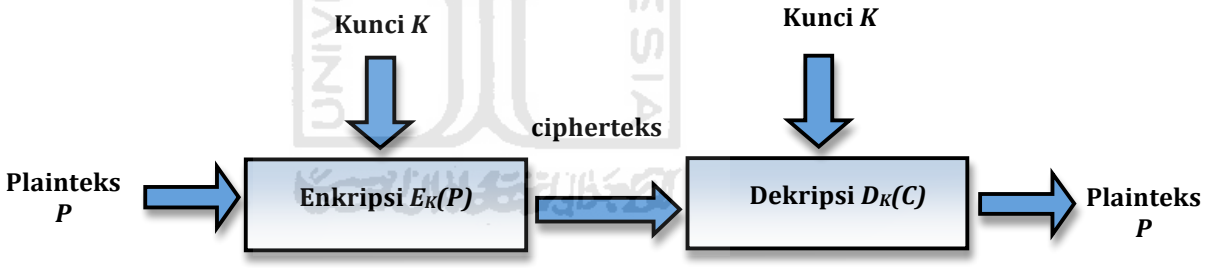
C = Cipherteks

K = Kunci

E_K = Proses enkripsi menggunakan kunci *K*

D_K = Proses dekripsi menggunakan kunci *K*

Skema enkripsi dan dekripsi dengan menggunakan kunci diperlihatkan pada gambar dibawah ini :



Gambar 2.1 Skema enkripsi dan dekripsi menggunakan kunci

5. Sistem Kriptografi

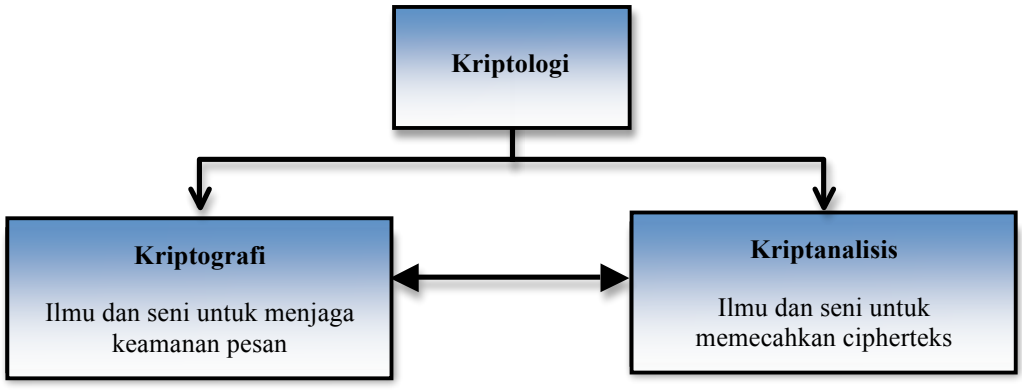
Kriptografi membentuk sistem yang dinamakan sistem kriptografi. Sistem kriptografi (*Cryptosystem*) adalah kumpulan yang terdiri dari algoritma kriptografi, semua plainteks dan cipherteks yang mungkin, dan kunci.

6. Penyadap

Penyadap (*eavedropppers*) adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadapan adalah untuk mendapatkan infromasi sebanyak-banyaknya mengenai sistem kriptografi yang digunakan untuk berkomunikasi dengan maksud untuk memecahkan cipherteks.

7. Kriptanalisis dan kriptologi

Kriptografi berkembang sedemikian rupa sehingga melahirkan bidang yang berlawanan yaitu kriptanalisis. Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan cipherteks menjadi plainteks tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptanalis. Jika seorang kriptografer (*cryptographer*) mentransformasikan plainteks menjadi cipherteks dengan suatu algoritma dan kunci maka sebaliknya seorang kriptanalis berusaha untuk memecahkan cipherteks tersebut untuk menemukan plainteks atau kunci. Kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalisis. Baik kriptografi maupun kriptanalisis keduanya saling berkaitan, dpat dilihat seperti gambar dibawah ini :



Gambar 2.2. Bagan hubungan antara kriptografi dengan kriptanalisis

Dari paparan awal dapat dirangkumkan bahwa kriptografi bertujuan untuk memeberi layanan kemanan. Yang dinamakan aspek-aspek keamanan sebagai berikut :

1. Kerahasiaan (*confidentiality*)

   Adalah layanan yang ditujukan unutk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak. Di dalam kriptografi layanan ini direalisasikan dengan menyandikan plainteks menjadi cipherteks. Misalnya pesan "harap datang pukul 8" disandikan menjadi "trxC#45motyptre!%".

2. Integritas data (*data integrity*)

   Adalah layanan yang menjamin bahwa pesan masih asli/utuh atau belum pernah dimanipulasi selama pengririman.

3. Otentikasi (*authentication*)

   Adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasikan kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengidentifikasi kebenaran sumber pesan (*data origin authentication*). Dua pihak yang saling berkomunikasi harus dapat mengotentikasi satu sama lain sehingga ia dapat memastikan sumber pesan.

4. Nirpenyangkalan (*non-repudiation*)

   Adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

   (Rinaldi Munir, 2006, h.3)

2.2 Steganografi

Steganografi adalah seni dan ilmu menulis atau menyembunyikan pesan tersembunyi dengan suatu cara sehingga selain si pengirim dan si penerima, tidak ada seorangpun yang mengetahui atau menyadari bahwa ada suatu pesan rahasia. Kata steganografi (steganografi) berasal dari bahasa Yunani steganos, yang artinya "tersembunyi atau terselubung", dan graphein, "menulis".

Teknik steganografi meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia (teks atau gambar) di dalam file-file lain yang mengandung teks, image, bahkan audio tanpa menunjukkan ciri-ciri perubahan yang nyata atau terlihat dalam kualitas dan struktur dari file semula. Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi.

Metode yang digunakan dalam steganografi bisa bermacam-macam. Begitu pula medium perantaranya, bisa bermacam-macam juga, sesuai dengan perkembangan masa. Pada zaman dahulu, teknik steganografi sederhana sudah banyak digunakan untuk merahasiakan pesan. Seperti Leonardo Da Vinci yang konon menyembunyikan pesan-pesan tertentu di balik karya-karya lukisannya. Era digital yang merambah kehidupan manusia modern telah berhasil membawa perubahan dalam implementasi teknik steganografi. Sekarang, semuanya serba digital. Data-data tak lagi disimpan dalam bentuk kertas, namun tersimpan rapi dalam bentuk file-file digital. Maka, steganografi pun mengikuti arus perkembangan ini. Kini, pesan rahasia disembunyikan di dalam suatu file teks, gambar, audio, atau bahkan video. Memang tidak kasat mata. Namun dapat mudah ditemukan oleh tools tertentu yang mengenali polanya.

2.3 Algoritma Kriptografi Klasik

Sebelum komputer ada, kriptografi dilakukan dengan menggunakan pensil dan kertas. Algoritma kriptografi (*cipher*) yang digunakan saat itu, dinamakan juga algoritma klasik, adalah berbasis karakter, yaitu enkripsi dan dekripsi dilakukan pada setiap karakter pesan. Semua algoritma klasik termasuk ke dalam sistem kriptografi simetris dan digunakan jauh sebelum kriptografi kunci publik ditemukan.

Kriptografi klasik memiliki beberapa ciri :

1. Berbasis karakter
2. Menggunakan pena dan kertas saja, belum ada komputer
3. Termasuk ke dalam kriptografi kunci simetris

Tiga alasan mempelajari algoritma klasik :

1. Memahami konsep dasar kriptografi
2. Dasar algoritma kriptografi modern
3. Memahami kelamahan sistem kode

(Dony Ariyus, 2008, h.49)

Pada dasarnya, algoritma kriptografi dapat dikelompokkan ke dalam dua macam cipher, yaitu :

1. Cipher substitusi (*substitution cipher*)

   Di dalam cipher substitusi setiap unit plainteks diganti dengan satu unit cipherteks. Satu unit di sini berarti satu huruf, pasangan huruf, atau dikelompokkan lebih dari dua huruf. Algoritma substitusi tertua yang diketahui adalah *Caesar cipher* yang digunakan oleh kaisar Romawi, Julius Caesar untuk mengirimkan pesan yang dikirimkan kepada Gubernurnya.

2. Cipher transposisi (*transposition cipher*)

   Pada cipher transposisi, huruf-huruf di dalam plainteks tetap saja, hanya saja urutan diubah. Dengan kata lain algoritma ini melakukan transpose terhadap rangkaian karakter di dalam teks. Nama lain untuk metode ini adalah

permutasi atau pengacakan (*scrambling*) karena transpose setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut.

2.4 Vigenere Cipher

Vigenere cipher mungkin adalah contoh terbaik dari cipher alfabet-majemuk. Algoritma ini dipublikasikan oleh diplomat perancis, Blaise de Vigenere pada abad 16, meskipun Giovan Batista Belaso telah menggambarkannya pertama kali pada tahun 1553 seperti ditulis di dalam bukunya La Cifra del Sig. Vigenere cipher dipublikasikan pada tahun 1586, tetapi algoritma tersebut baru dikenal luas 200 tahun kemudian yang oleh penemunya tersebut dinamakan vigenere cipher. Cipher ini berhasil dipecahkan oleh Babbage dan Kasiski pada pertengahan abad 19.

Vigenere cipher sangat dikenal karena mudah dipahami dan diimplementasikan. Bila pada teknik substitusi lainnya setiap teks-kode selalu mengganti nilai setiap teks-asli tertentu, maka pada teknik substitusi Vigenere setiap teks-kode bisa memiliki banyak kemungkinan teks-asli. Teknik dari substitusi bisa dilakukan dengan angka dan huruf.

2.4.1 Teknik substitusi angka

Teknik substitusi Vigenere dengan menggunakan angka dilakukan dengan menukarkan huruf dengan angka. (Dony Ariyus, 2008, h. 65)

A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
0	1	2	3	4	5	6	7	8	9	10	11	12

N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
13	14	15	16	17	18	19	20	21	22	23	24	25

Teknik angka pada metode ini adalah deretan plainteks dan kunci diterjemahkan kedalam angka, kemudian plainteks dan kunci dijumlahkan, hasilnya di modulo dengan 26 untuk mendapatkan chipertextnya, dengan rumus enkripsi:

$C_i = (p_i + k_i) \bmod 26 \qquad (1 \leq i \leq t)$

Yang dalam hal ini,

p_i : plainteks

c_i : cipherteks

k_i : kunci

Sedangkan rumus untuk dekripsi adalah :

$p_i = (c_i - k_i) \bmod 26 \qquad (1 \leq i \leq t)$

Contoh :

Plainteks : PLAINTEKS

Kunci : SONY

Cara menyelesaikannya adalah ulang kunci sesuai dengan panjang plainteks, kemudian ubah semua huruf plainteks menjadi angka sehingga didapat deretan angka :

P	L	A	I	N	T	E	K	S
15	11	0	8	13	19	4	10	18
S	O	N	Y	S	O	N	Y	S
18	14	13	24	18	14	13	24	18

Setelah semua sudah berubah menjadi angka, maka jumlahkan plainteks dengan kunci yang bersesuain. Kemudian di modulo dengan 26.

P	=	15	+	18	mod	26	=	7		T	=	19	+	14	mod	26	=	7
L	=	11	+	14	mod	26	=	25		E	=	4	+	13	mod	26	=	17
A	=	0	+	13	mod	26	=	13		K	=	10	+	24	mod	26	=	8
I	=	8	+	24	mod	26	=	6		S	=	18	+	18	mod	26	=	10
N	=	13	+	18	mod	26	=	5										

Langkah selanjutnya, ubah kembali angka hasil enkripsi menjadi huruf, sehingga didapat cipherteksnya adalah **HZNGFHRVL.**

2.4.2 Teknik substitusi huruf

Untuk mengenkripsi pesan dengan kode Vigenere digunakan *tabula recta* atau disebut juga bujursangkar Vigenere seperti gambar dibawah ini.

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

Gambar 2.3 Bujur sangkar vigenere

Tabula recta digunakan untuk memperoleh teks-kode dengan menggunakan kunci yang sudah ditentukan. Jika panjang kunci lebih pendek daripada panjang teks-asli maka pengguaan kunci diulang. Secara matematis enkripsi dengan Vigenere cipher bisa dinyatakan dengan :

E(pi) = V(pi, k(i mod m))

dengan :

pi = huruf ke-I dalam teks-asli

kn = huruf ke-n dalam kunci

m = panjang kunci, dan

V(x,y) = huruf yang tersimpan pada baris x dan kolom y pada *tabula recta*.

Contoh vigenere cipher adalah sebagai berikut :

Teks-asli : "KEAMANAN DATA MENGGUNAKAN CIPHER VIGENERE"

Kunci : "KRIPTOGRAFI"

Dengan menggunakan algoritma Vigenere cipher maka akan didapat teks-kode "UVIBTBGE DFBK WVVVZCTRKFV FZOTGSXHE HQZYMP".

Cara menentukan teks-kode pada sistem ini, pada tabula recta bisa dilihat bahwa posisi horizontal merupakan teks-asli dan pada posisi vertikal adalah kunci. Jika teks-asli huruf "K" maka lihat posisi huruf K pada teks-asli tabula recta dan posisi huruf "K" pada posisi kunci jika huruf pertama kunci juga kebetulan "K". Jika sudah ditemukan, tarik garis lurus ke bawah dari teks-asli dan garis lurus ke samping dari posisi kunci maka akan ditemukan huruf "U". Huruf U inilah yang akan menjadi teks-kode. Begitu pula seterusnya sampai seluruh karakter di teks-asli terpenuhi.

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

Gambar 2.4 Penggunaan bujursangkar vigenere

Untuk mendekripsi pesan, penerima harus mengetahui kunci yang digunakan lalu mencari baris huruf manakah yang menghasilkan huruf pada teks-kode jika kolomnya adalah huruf yang bersesuaian pada kunci. Misalnya, pada huruf pertama teks-kode (D), huruf yang bersesuaian pada kunci yang digunakan adalah D. Dengan melihat tabula recta, huruf D pada tabel untuk baris huruf D ada pada kolom huruf A, karena itu huruf pertama teks-asli adalah huruf A.

Salah satu kelebihan Vigenere cipher adalah sulitnya melakukan kriptanalisis dengan metode analisis frekuensi karena dua huruf yang sama dalam teks-kode belum tentu bisa dideskripsikan menjadi dua huruf yang sama dalam teks-asli.

Kelemahan utama Vigenere cipher adalah kuncinya yang pendek dan penggunaanya yang berulang-ulang. Jika kriptanalis dapat menentukan panjang kunci saja maka teks-kode dapat diperlakukan seperti rangkaian beberapa kode Kaisar. (Dony Ariyus, 2008, h. 67)

BAB III

METODOLOGI

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, dan hambatan-hambatan yang terjadi serta kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan-perbaikannya.

3.2 Kebutuhan Perangkat

Dalam perancangan suatu sistem diperlukan suatu *software developer*, yaitu sebuah perangkat lunak yang dapat digunakan untuk membangun program. Pemilihan jenis *software developer* sendiri disesuaikan dengan desain dan rancangan program. Perangkat keras juga menjadi hal yang penting dalam perancangan suatu sistem, karena di perangkat inilah *software* akan berjalan untuk kemudian digunakan untuk membangun suatu program.

Perangkat keras yang digunakan dalam pembuatan program ini adalah:

1. Processor Intel Pentium 4
2. RAM 512
3. Hard Disk 80 GB
4. Keyboard dan Mouse
5. Monitor 15”

Perangkat lunak yang digunakan dalam pembuatan program ini adalah:

1. Sistem Operasi Windows XP Professional Service Pack 2
2. Microsoft Visual Basic 6.0

3.3 Kebutuhan Input

File-file masukan yang dibutuhkan untuk proses enkripsi adalah file plainteks yang berupa file gambar dengan ekstensi JPEG, GIF, BMP, dan BMP, serta file teks dengan ekstensi TXT, dan RTF. Sedangkan file-file masukan yang dibutuhkan selama proses dekripsi adalah file yang telah terkunci dengan ekstensi LCK.

3.4 Kebutuhan Proses

Terdapat dua proses utama dalam program ini, yaitu proses enkripsi dan dekripsi. Proses enkripsi merupakan proses pengamanan data yang dikirimkan agar terjaga kerahasiannya. Selama proses enkripsi pesan asli atau plainteks akan diubah menjadi kode-kode yang tidak dimengerti. Enkripsi dilakukan dengan menggunakan 256 karakter yang ada dalam ASCII. Selama membaca isi pesan asli untuk selanjutnya dienkripsikan, proses pengambilan karakter dilakukan per kilobyte data, hal ini bertujuan untuk mempercepat proses enkripsi.

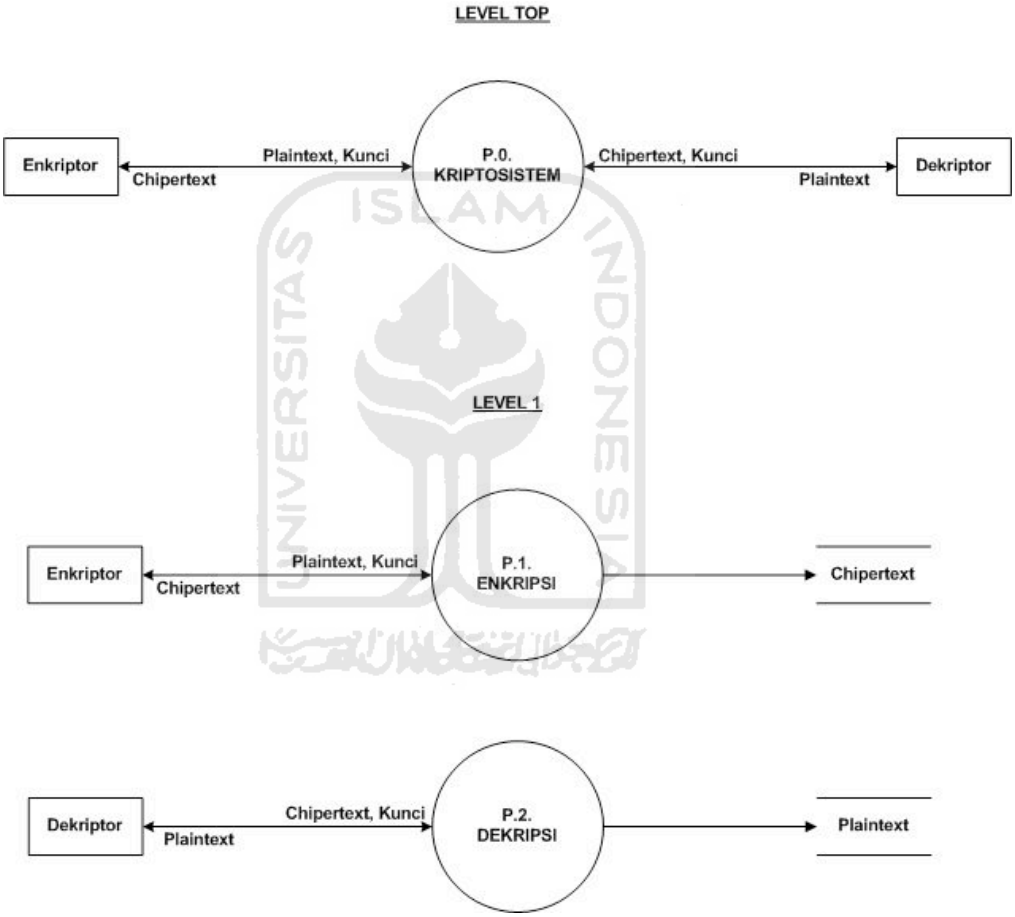
Proses dekripsi merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya. Semua file terkunci bisa didekripsikan, namun hanya dekripsi dengan kunci yang tepat saja yanga meng akan menghasilkan output file yang sesuai dengan bentuk asalnya.

3.5 Kebutuhan Output

Output yang dihasilkan oleh proses enkripsi adalah file terkunci dengan ekstensi LCK. Sedangkan output yang dihasilkan oleh proses dekripsi adalah file dengan ekstensi JPEG, BMP, GIF, TXT, RTF sesuai dengan data aslinya sebelum dienkripsi.

3.6 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana sistem disimpan, proses apa saja yang menghasilkan data tersebut, dan interaksi data yang tersimpan dan proses yang dikenakan pada data tersebut. *Data Flow Diagram* yang digunakan dalam merancang sistem ini adalah :



Gambar 3.1 *Data Flow Diagram* sistem

3.7 Algoritma dan Flowchart

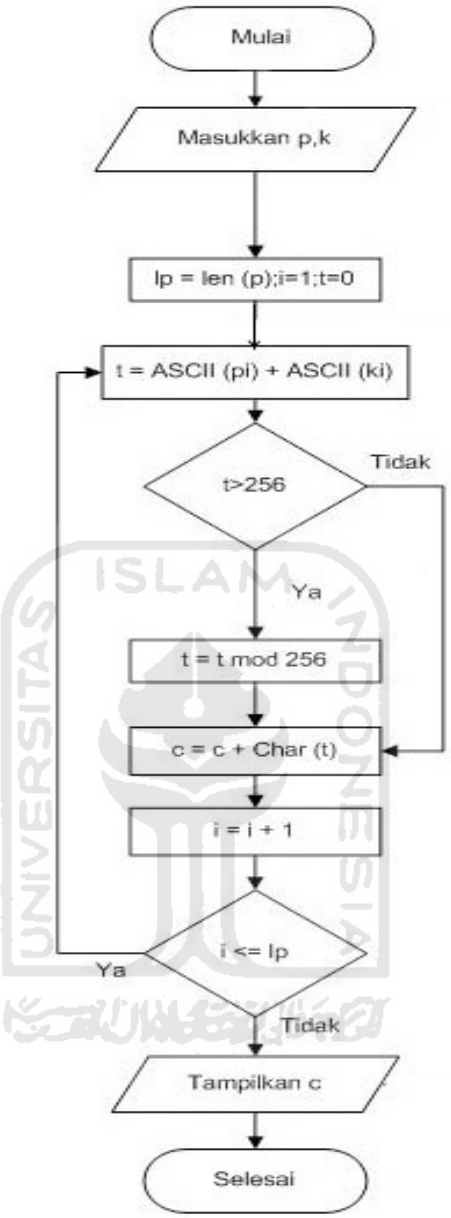
Algoritma pemrograman adalah urutan langkah-langkah logis untuk membangun sebuah program yang disusun secara sistematis. Langkah-langkah dalam membangun sebuah program berkaitan dengan desain dan rancangan program. *Flowchart* adalah gambaran dalam bentuk diagram

alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut.

Algoritma yang digunakan untuk enkripsi adalah :

1. Masukkan Plainteks(P) dan Kunci(K)
2. Untuk inisialisasi awal lp=len(P), i=1 dan t=0
3. Kerjakan langkah 6 sampai 9 selama i ≤ lp
4. Tentukan t=ASC(Pi) + ASC(Ki)
5. Jika t>256 kerjakan langkah 6,jika tidak kerjakan langkah 7
6. Tentukan t=t mod 256
7. Tentukan C=C + CHR(t)
8. i = i + 1
9. Tampilkan C

Dari algoritma diatas tentang enkripsi dan dekripsi, maka *flowchart* dari program adalah sebagai berikut :



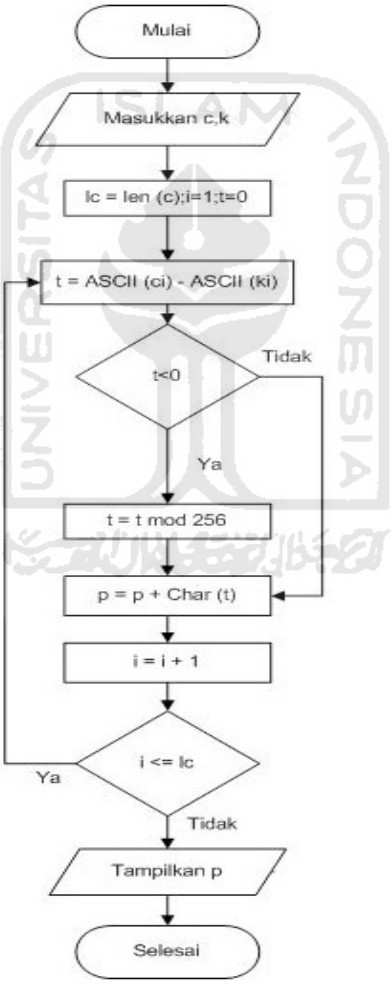
Gambar 3.2 *Flowchart* proses enkripsi

Algoritma yang digunakan untuk proses dekripsi adalah :

1. Masukkan Cipherteks(C) dan Kunci(K)
2. Untuk inisialisasi awal lc=len(C), i=1 dan t=0
3. Kerjakan langkah 4 sampai 7 selama i ≤ lc

4. Tentukan t=ASC(Ci) - ASC(Ki)
5. Jika t<0 kerjakan langkah 6, jika tidak kerjakan langkah 7
6. Tentukan t=t mod 256
7. Tentukan P=P + CHR(t)
8. i = i + 1
9. Tampilkan P

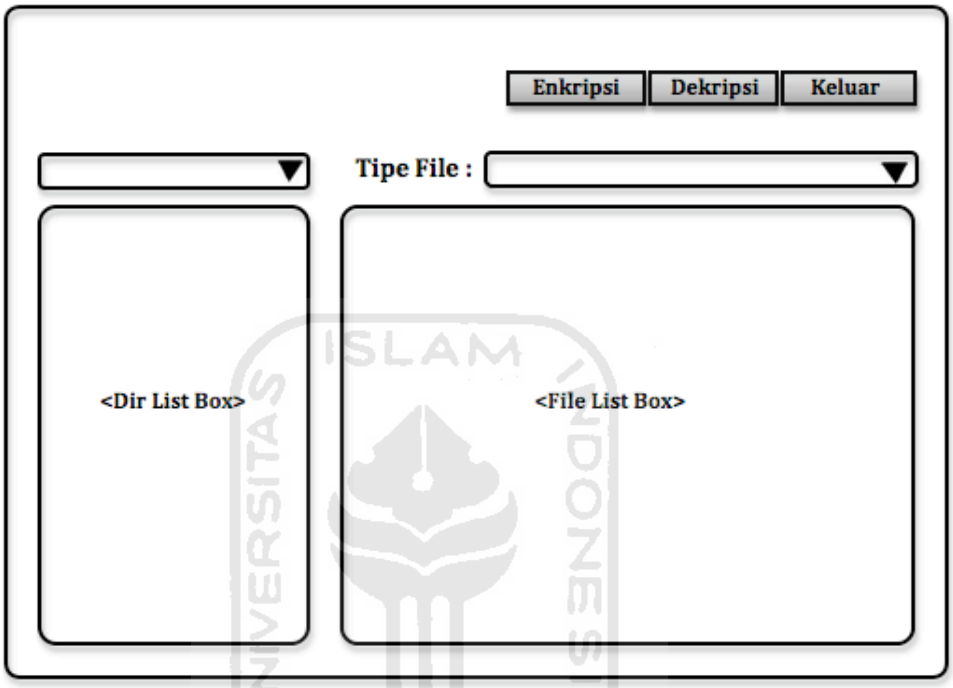
Dari algoritma diatas tentang enkripsi dan dekripsi, maka *flowchart* dari program adalah sebagai berikut :



Gambar 3.3 *Flowchart* proses dekripsi

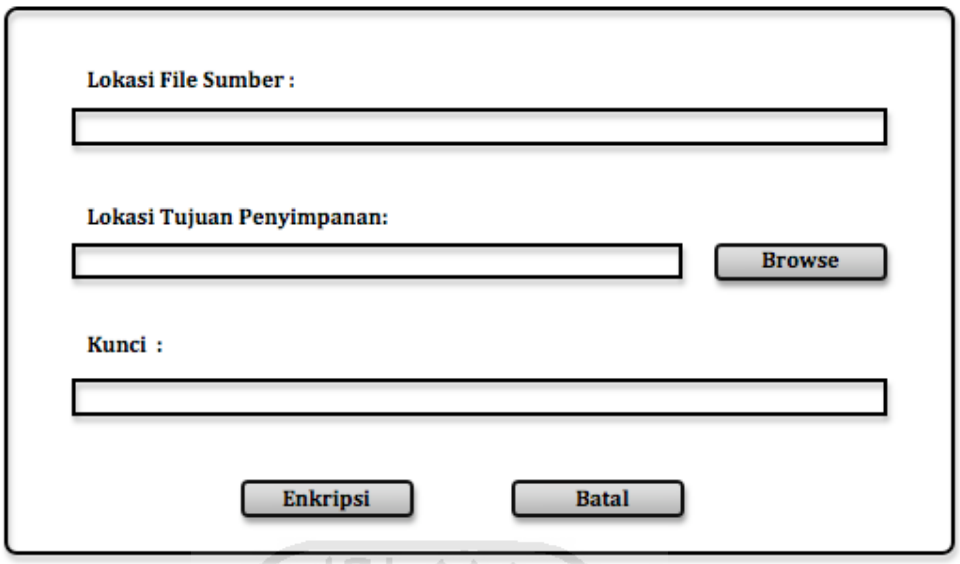
3.8 Perancangan Tampilan Program

Menu utama digunakan untuk menentukan file yang akan diproses, tersedia filter yang digunakan untuk menyaring tipe file tertentu sesuai dengan yang ditentukan.



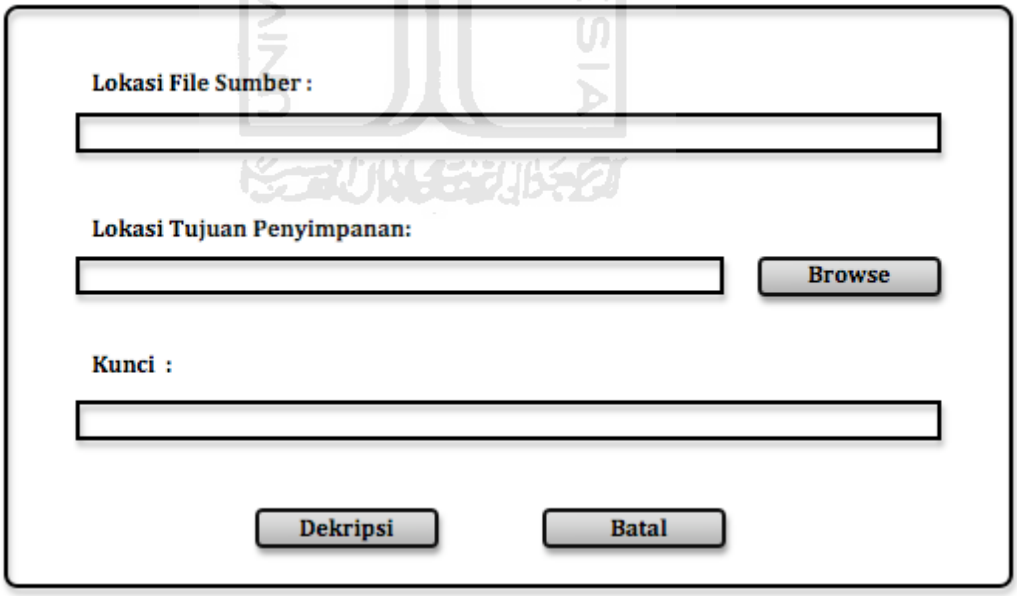
Gambar 3.4 Rancangan antarmuka menu utama

Form enkripsi digunakan untuk enkripsi file, untuk melakukan enkripsi diwajibkan untuk mengisi kunci enkripsi terlebih dahulu.



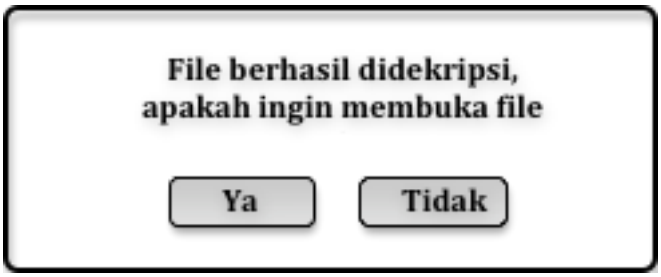
Gambar 3.5 Rancangan antarmuka menu enkripsi

Form dekripsi digunakan untuk dekripsi file yang sebelumnya sudah dienkripsikan, untuk melakukan dekripsi juga diwajibkan untuk mengisi kunci dekripsi terlebih dahulu.



Gambar 3.6. Rancangan antarmuka menu dekripsi

Message Box menampilkan konfirmasi pertanyaan untuk menampilkan hasil dari file dekripsi secara otomatis sesuai dengan tipe file.



Gambar 3.7. Rancangan antar muka konfirmasi membuka file

BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Tugas yang pertama kali dikerjakan sistem adalah *load file*, proses ini bertujuan untuk menampung file sebelum dilakukan proses enkripsi dan dekripsi. Setiap file yang diload dibaca isinya kemudian isi tersebut ditampung ke dalam bentuk *array*. Berikut ini adalah prosedur yang digunakan untuk membaca file inputan.

```
Private Sub FileOpen(KFileName As String)
On Error Resume Next
Dim FileNum As Integer
Dim temp As String
FileNum = FreeFile
Open KFileName For Binary As #FileNum
FileSize = LOF(FileNum)
ReDim ArrayFile(FileSize)
Get #FileNum, , ArrayFile
Close #FileNum
End Sub
```

Setelah *load file* kemudian sistem membaca kunci dan mengubahnya ke dalam bentuk array. Isi dari variabel kunci diubah ke dalam bentuk ASCII.

```
maksKunci = Len(Trim(txtKunci.Text))
ReDim kunci(maksKunci)
For i = 1 To maksKunci
  kunci(i) = Asc(Mid(Trim(txtKunci.Text), i, 1))
Next i
```

Proses enkripsi dan dekripsi dilakukan per 1 *kilobyte* data, setiap ukuran file di *DIV* dengan 1024, ini bertujuan untuk mempercepat proses perulangan dan digunakan untuk kebutuhan *progress bar.*

```
maksimal = (FileSize \ 1024) + 1
```

Proses perulangan dilakukan untuk pengambilan karakter untuk kemudian dikonversikan dari plainteks menjadi cipherteks maupun sebaliknya. Proses perulangan dimulai dari 0 agar dapat menghandle file dibawah 1024.

```
Private Function Enk_vigenere(ByVal maksimal As Integer) As
String

 Dim temporary As String
 temporary = Empty
 For i = 0 To maksimal - 2
   awal = (i * 1024) + 1
   akhir = (i * 1024) + 1024
   chiper = konvert(awal, akhir)
   temporary = temporary + chiper
   pg.Value = pg.Value + 1
 Next i
 awal = (i * 1024) + 1
 akhir = FileSize
 chiper = konvert(awal, akhir)
 temporary = temporary + chiper
 pg.Value = pg.Value + 1
 Enk_vigenere = temporary
 End Function
```

Funsi Konvert digunakan untuk konversi dari plainteks menjadi cipherteks maupun sebaliknya. Setiap karakter hasil perhitungan di*modulus* dengan 256 agar nilai tetap berada pada jangkauan 0-255 karakter. Prinsip

penggunaan kunci pada metode Vigenere Cipher adalah menyusun kunci bersesuaian dengan plainteks yang ada di atasnya. Apabila telah sampai di akhir kunci, penyusunan kunci diulang kembali sampai seluruh plainteks telah memiliki karakter kunci masing-masing.

```
    Private Function konvert(ByVal awal As Long, ByVal akhir
As Long) As String

 Dim nilaiplain, nilaiKunci As Integer
 Dim hitung As Byte
 Dim hasil As String
 hitung = 1
 For i = awal To akhir
       nilaiplain = ArrayFile(i)
       nilaiKunci = kunci(hitung)
       hasil = hasil + Chr((nilaiplain + nilaiKunci) Mod 256)
       If hitung = maksKunci Then
             hitung = 1
                   Else
             hitung = hitung + 1
       End If
 Next i
 konvert = hasil
 End Function
```

Perbedaan mendasar antara proses enkripsi dan dekripsi adalah penggunaan operator, pada enkripsi menggunakan operator + sedangkan pada dekripsi menggunakan operator -. Khusus untuk dekripsi selama proses dekripsi akan ditampung 5 karakter awal hasil dekripsi, proses ini ditujukan untuk penentuan tipe dari file tersebut.

```
   plain = ((nilaiplain - nilaiKunci) + 256) Mod 256
       If i <= 5 Then
```

```
            kode(i) = plain
        End If
    hasil = hasil + Chr(plain)
```

Selama proses dekripsi akan dilakukan pengecekan tipe file, proses ini menggunakan kode yang merupakan 5 karakter awal yang sebelumnya telah diambil.

```
Private Function cek_type() As String

 If kode(1) = 255 And kode(2) = 216 And kode(3) = 255 Then
 cek_type = ".jpg"
 ElseIf kode(1) = 66 And kode(2) = 77 Then
 cek_type = ".bmp"
 ElseIf kode(1) = 137 And kode(2) = 80 And kode(3) = 78 Then
 cek_type = ".png"
 ElseIf kode(1) = 71 And kode(2) = 73 And kode(3) = 70 Then
 cek_type = ".gif"
 ElseIf kode(1) = 123 And kode(2) = 92 And kode(3) = 114 Then
 cek_type = ".rtf"
 Else
 cek_type = ".txt"
 End If
End Function
```

Setelah proses enkripsi dan dekripsi berhasil maka sistem akan otomatis menyimpan file hasil konversi. Lokasi default penyimpanan file adalah sama dengan lokasi file awal, namun lokasi penyimpanan dapat diubah sesuai kebutuhan. Berikut ini adalah fungsi untuk menyimpan file.

```
Function SimpanFile(FileName, txtFile As RichTextBox) As
Boolean

 On Error Resume Next
 Dim strContents As String
```

```
Dim FileNum As Integer
Dim status As Boolean
status = False
Screen.MousePointer = 11
FileNum = FreeFile
strContents = txtFile.Text
If Dir(FileName) <> "" Then
x = MsgBox("nama File sudah ada, apakah ingin direplace?",
vbExclamation + vbYesNo, "Peringatan")
If x = vbYes Then
    Open FileName For Output As #FileNum
    Print #FileNum, strContents
    Close #FileNum
    status = True
  End If
Else
  Open FileName For Output As #FileNum
  Print #FileNum, strContents
  Close #FileNum
  status = True
End If
Screen.MousePointer = 0
SimpanFile = status
End Function
```

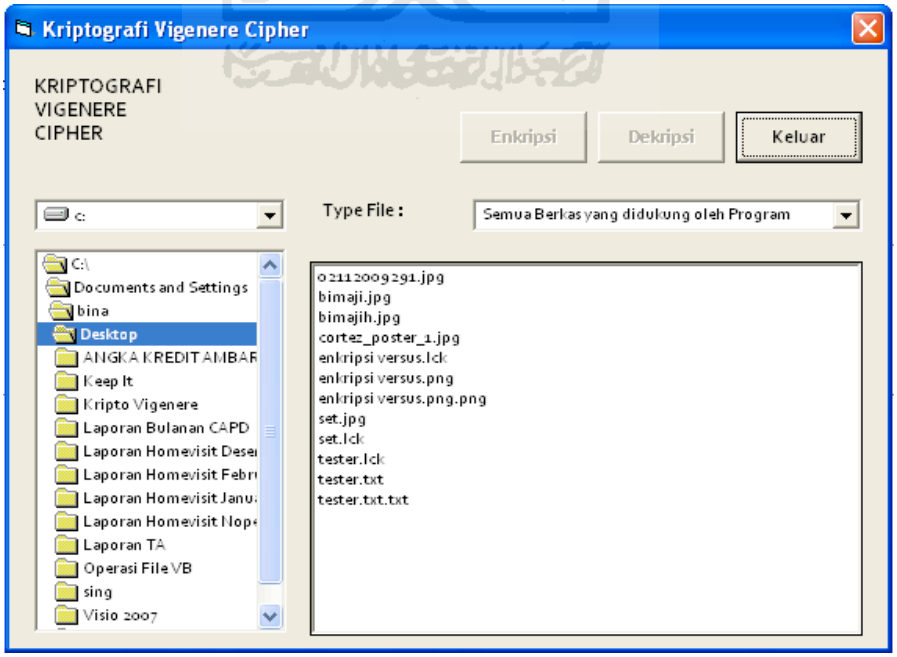
Khusus untuk dekripsi, setelah file disimpan maka file dapat langsung dibuka tanpa harus membuka lokasi file tersebut secara manual. Proses ini menggunakan fungsi *shellexecute.*

```
x = MsgBox("Dekripsi File telah selesai" & Chr(10) & _
"Ingin Membuka File " & Mid(FrmMain.File1.FileName, 1,
Len(FrmMain.File1.FileName) - 4) & "." & _
```

```
Right(txtFileTujuan.Text, 3) & "?", vbInformation + vbYesNo,
"Konfirmasi")

If x = vbYes Then

 ShellExecute 0, vbNullString, Trim(txtFileTujuan.Text),
vbNullString, vbNullString, vbNormalFocus

End If
```

4.2 Analisis Kerja Sistem

Program terdiri dari 3 menu utama, yaitu menu utama, enkripsi, dan dekripsi. Saat pertama kali membuka program, akan ditampilkan menu utama yang berisikan *Drive List Box* untuk menentukan lokasi *drive, Directory List Box* untuk menentukan letak direktori, dan *File List Box* untuk menentukan file yang akan diproses. Fungsi utama halaman ini adalah untuk menentukan lokasi file yang nantinya akan dienkripsi atau didekripsi. Untuk mempermudah pemakaian disediakan pula *Combo Box* yang berperan sebagai filter untuk menampilkan berkas sesuai tipe yang disediakan.



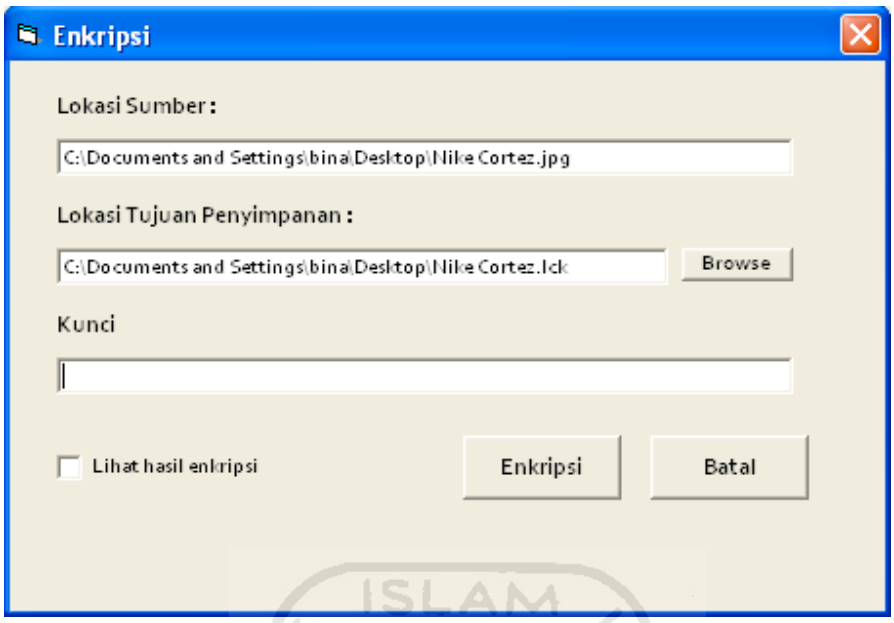
Gambar 4.1 Halaman utama sistem

Untuk menggunakan program pertama kali harus memilih file untuk dienkripsi atau dekripsi. Dalam pilihan *Type File* disediakan 4 pilihan, yaitu:

1. Berkas Gambar (*.jpg; *.bmp; *.png; *.gif)
2. Berkas Teks (*.txt; *.rtf)
3. Berkas Terkunci (*.lck)
4. Semua berkas yang didukung oleh program

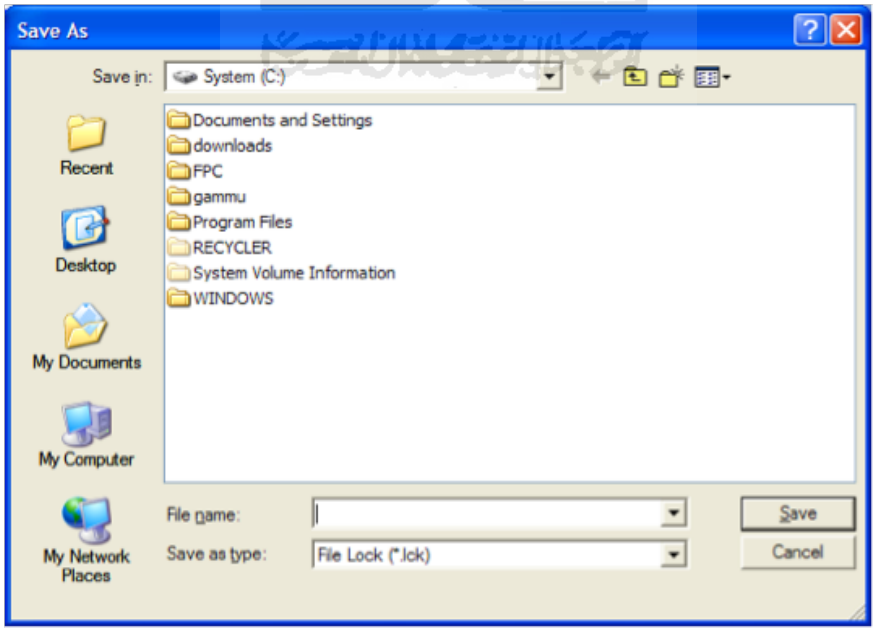
Apabila pengguna memilih file dengan ekstensi lck maka tombol dekripsi akan aktif, sedangkan untuk file lainnya maka tombol enkripsi yang akan aktif. Untuk mengenkripsi file, tentukan filenya kemudian tekan tombol enkripsi. Dalam pengujian ini akan digunakan plainteks berupa file gambar dengan nama file *Nike Cortez.jpg*.

Gambar 4.2 Plainteks gambar



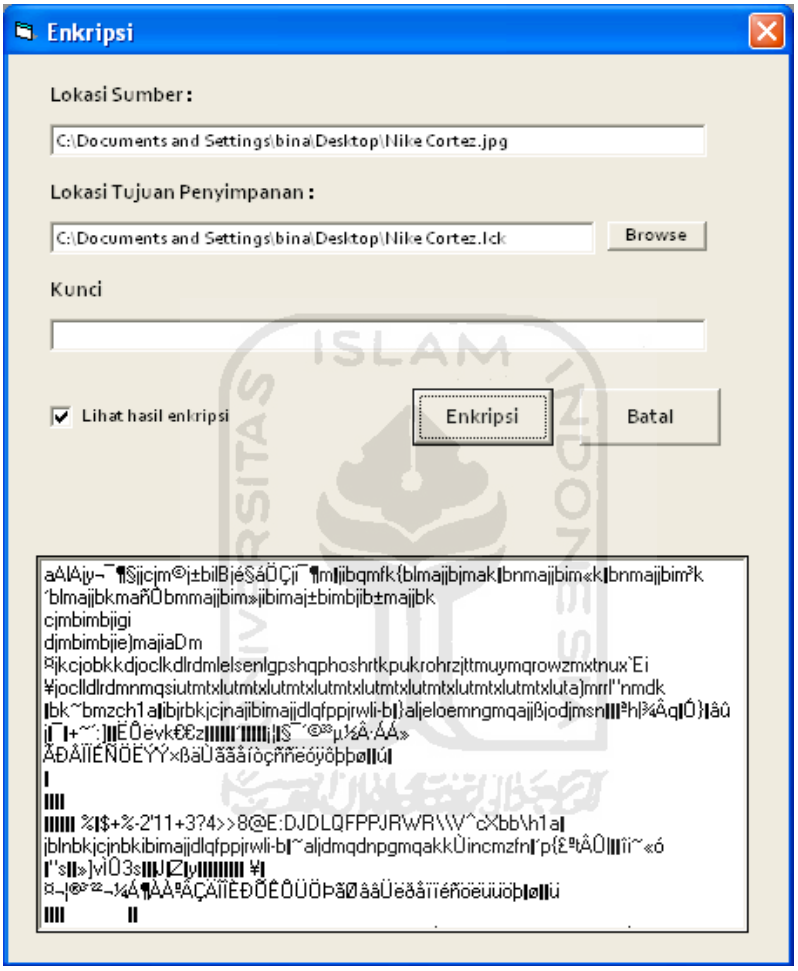
Gambar 4.3 Halaman Enkripsi File

Di dalam halaman Enkripsi pada gambar 4.3, file yang dipilih pada menu utama akan ditetapkan sebagai file sumber. *Defaultnya* lokasi tujuan penyimpanan adalah sama dengan lokasi sumber file tersebut, namun lokasi tujuan penyimpanan bisa diubah dengan memilih tombol jelajah.



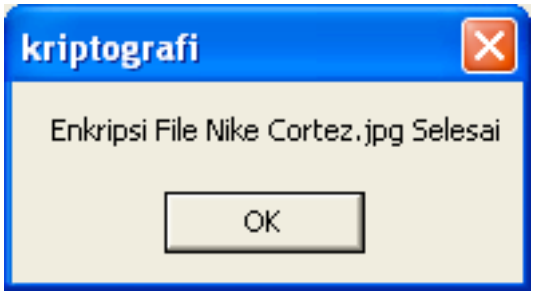
Gambar 4.4. Halaman jelajah

Sebelum file dienkripsi diharuskan untuk memasukkan kunci enkripsi, kunci bersifat simetris, sehingga kunci yang digunakan untuk enkripsi nantinya akan digunakan juga untuk dekripsi file. Pengguna bisa melihat hasil enkripsi dengan mencentang pilihan Lihat Hasil Enkripsi.



Gambar 4.5 Halaman enkripsi dengan tampilan hasil enkripsi

File lck hasil enkripsi akan langsung disimpan sesuai dengan lokasi yang telah ditentukan sebelumnya. Akan muncul notifikasi bahwa file berhasil di enkripsi.

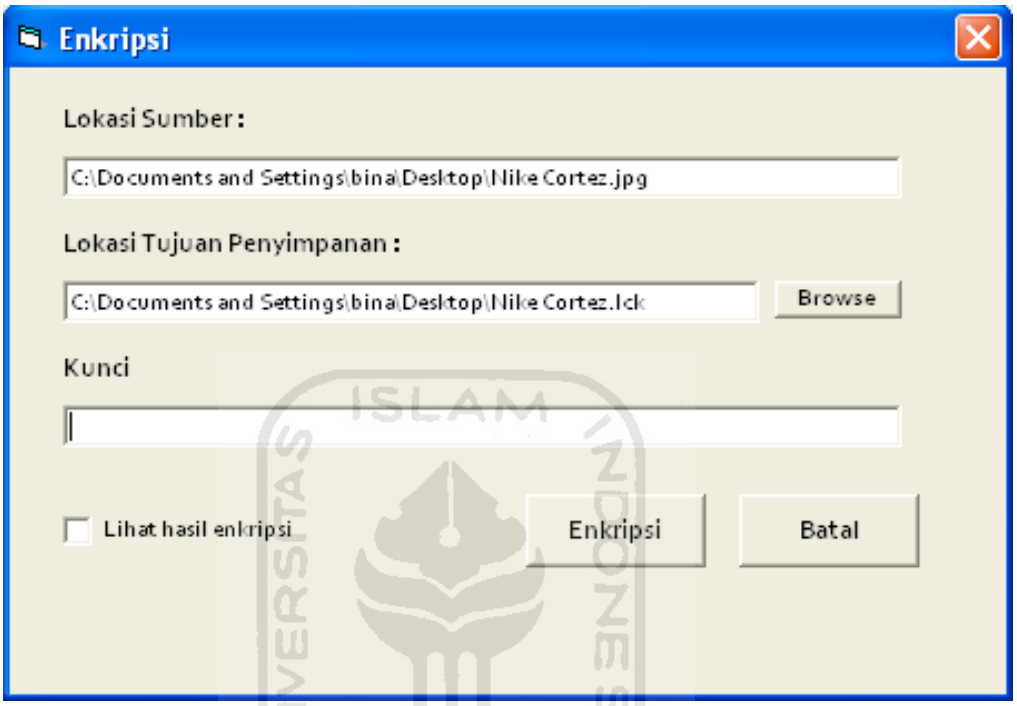


Gambar 4.6 Notifikasi proses enkripsi selesai

Proses enkripsi akan menghasilkan output cipherteks berupa file dengan ekstensi lck. File awal yang berupa file gambar telah berhasil terenkripsi, dan apabila file lck tersebut dibuka akan menampilkan isi berupa deretan karakter-karakter yang sudah tidak bermakna dan susah untuk dipahami.

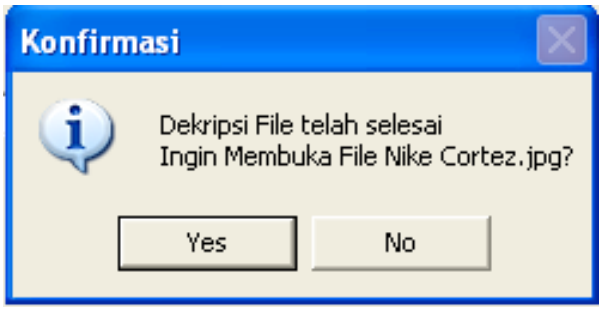
Gambar 4.7 Perbandingan plainteks dan cipherteks

Menu dekripsi hampir sama dengan menu enkripsi, baik dari file sumber, file tujuan, maupun cara memasukkan kunci, yang membedakan hanyalah tipe file yang bisa didekripsi yaitu file dengan tipe lck.



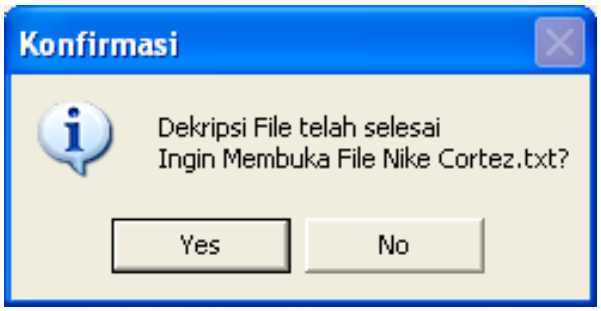
Gambar 4.8 Halaman dekripsi file

Untuk menghasilkan output yang sesuai dengan tipe file semula, pengguna harus memasukkan kunci yang sama dengan kunci enkripsi. Penentuan tipe file setelah didekripsi ditentukan secara otomatis dengan melihat isi dari file hasil dekripsi. Pengguna juga dapat langsung membuka file hasil dekripsi.



Gambar 4.9 Konfirmasi proses dekripsi

File lck akan tetap dapat terdekripsi dengan menggunakan kunci apa saja, kunci dekripsi yang berbeda dengan kunci enkripsi akan menghasilkan *output* berupa file teks.



Gambar 4.10 Tampilan hasil dekripsi dengan kunci yang salah

4.3 Pengujian Sistem

Dalam tahap ini dilakukan pengujian terhadap sistem yang telah dibangun untuk mengetahui apakah sistem ini telah sesuai dengan kebutuhan pengguna. Berikut ini adalah pembahasan prosedur enkripsi dan dekripsi untuk membandingkan hasil perhitungan secara manual dengan hasil perhitungan menggunakan sistem, dengan contoh kasus menggunakan

plainteks berupa file teks. Dalam pengujian ini ditentukan teks ‘kriptografi’ dengan menggunakan kunci ‘bimaji’.

Plainteks : kriptografi

Kunci : bimaji

Proses enkripsi 1

P(1) : k

ASCII P(1) : 107

K(1) : b

ASCII K(1) : 98

Enkripsi : 107 + 98 = 205

C(1) : Char (205) = Í

Proses enkripsi 2

P(2) : r

ASCII P(2) : 114

K(2) : i

ASCII K(2) : 105

Enkripsi : 114 + 105 = 219

C(2) : Char (219) = Û

Proses enkripsi 3

P(3) : i

ASCII P(3) : 105

K(3) : m

ASCII K(3) : 109

Enkripsi : 105 + 109 = 214

C(3) : Char (214) = Ö

Proses enkripsi 4

P(4) : p

ASCII P(4) : 112

K(4) : a

ASCII K(4) : 97

Enkripsi : 112 + 97 = 209

C(4) : Char (209) = Ñ

Proses enkripsi 5

P(5) : t

ASCII P(5) : 116

K(5) : j

ASCII K(5) : 106

Enkripsi : 116 + 106 = 222

C(5) : Char (222) = Þ

Proses enkripsi 6

P(6) : 0

ASCII P(6) : 111

K(6) : i

ASCII K(6) : 105

Enkripsi : 111 + 105 = 216

C(6) : Char (216) = Ø

Proses enkripsi 7

P(7) : g

ASCII P(7) : 103

K(7) : b

ASCII K(7) : 98

Enkripsi : 103 + 98 = 201

C(7) : Char (201) = É

Proses enkripsi 8

P(8) : r

ASCII P(8) : 114

K(8) : i

ASCII K(8) : 105

Enkripsi : 114 + 105 = 219

C(8) : Char (219) = Û

Proses enkripsi 9

P(9) : a

ASCII P(9) : 97

K(9) : m

ASCII K(9) : 109

Enkripsi : 97 + 109 = 206

C(9) : Char (206) = Î

Proses enkripsi 10

P(10) : f

ASCII P(10) : 102

K(10) : a

ASCII K(10) : 97

Enkripsi : 102 + 97 = 199

C(10) : Char (199) = Ç

Proses enkripsi 11

P(11) : i

ASCII P(11) : 105

K(11) : j

ASCII K(10) : 106

Enkripsi : 105 + 106 = 211

C(11) : Char (211) = Ó

Cipherteks=C(1)+C(2)+C(3)+C(4)+C(5)+C(6)+C(7)+C(8)+C(9)+C(10)+C(11)

= ÍÛÖÑÞØÉÛÎÇÓ

Cipherteks = ÍÛÖÑÞØÉÛÎÇÓ

Kunci = bimaji

Proses dekripsi 1

C(1) : Í

ASCII C(1) : 205

K(1) : b

ASCII K(1) : 98

Dekripsi : 205 - 98 = 107

P(1) : Char (107) = k

Proses dekripsi 2

C(2) : Û

ASCII C(2) : 219

K(2) : i

ASCII K(2) : 105

Dekripsi : 219 - 105 = 114

P(2) : Char (114) = r

Proses dekripsi 3

C(3) : Ö

ASCII C(3) : 214

K(3) : m

ASCII K(3) : 109

Dekripsi : 214 - 109 = 105

P(3) : Char (105) = i

Proses dekripsi 4

C(4) : Ñ

ASCII C(4) : 209

K(4) : a

ASCII K(4) : 97

Dekripsi : 209 - 97 = 112

P(4) : Char (112) = p

Proses dekripsi 5

C(5) : Þ

ASCII C(5) : 222

K(5) : j

ASCII K(5) : 106

Dekripsi : 222 - 106 = 116

P(5) : Char (116) = t

Proses dekripsi 6

C(6) : Ø

ASCII C(6) : 216

K(6) : i

ASCII K(6) : 105

Dekripsi : 216 - 105 = 111

P(6) : Char (111) = o

Proses dekripsi 7

C(7) : É

ASCII C(7) : 201

K(7) : b

ASCII K(7) : 98

Dekripsi : 201 - 98 = 103

P(7) : Char (103) = g

Proses dekripsi 8

C(8) : Û

ASCII C(8) : 219

K(8) : i

ASCII K(8) : 105

Dekripsi : 219 - 105 = 114

P(8) : Char (114) = r

Proses dekripsi 9

C(9) : Î

ASCII C(9) : 206

K(9) : m

ASCII K(9) : 109

Dekripsi : 206 - 109 = 97

P(9) : Char (97) = a

Proses dekripsi 10

C(10) : Ç

ASCII C(10) : 199

K(10) : a

ASCII K(10) : 97

Dekripsi : 199 - 97 = 102

P(10) : Char (102) = f

Proses dekripsi 11

C(11) : Ó

ASCII C(11) : 211

K(11) : j

ASCII K(11) : 106

Dekripsi : 211 - 106 = 105

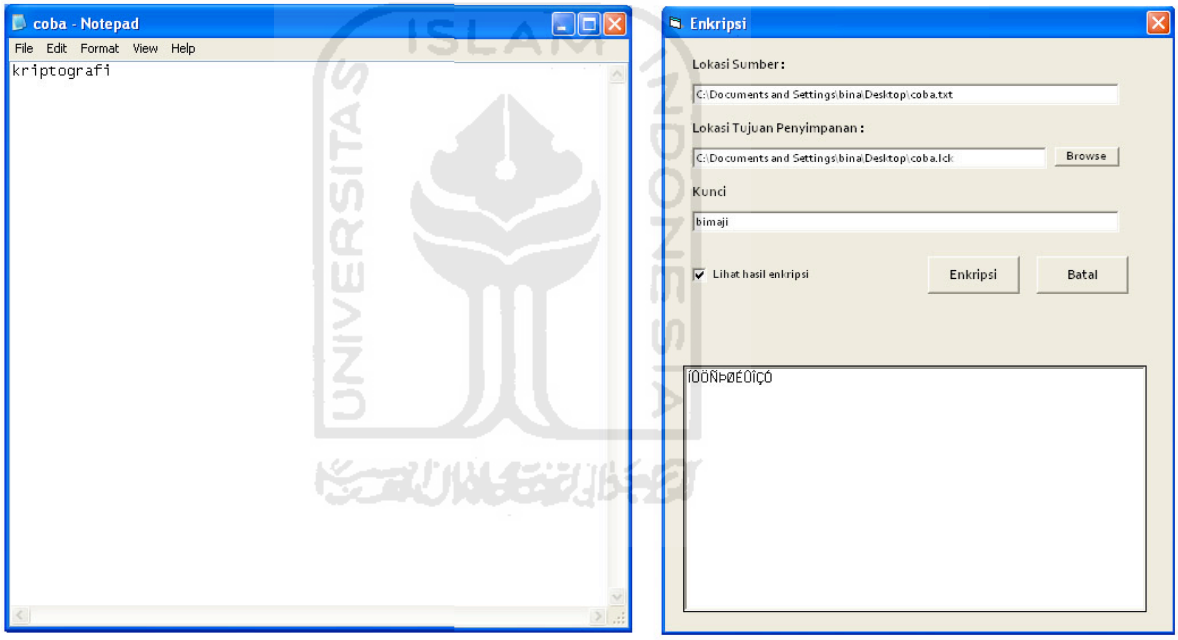
P(11) : Char (105) = i

Plainteks = P(1)+P(2)+P(3)+P(4)+P(5)+P(6)+P(7)+P(8)+P(9)+P(10)+P(11)

= kriptografi

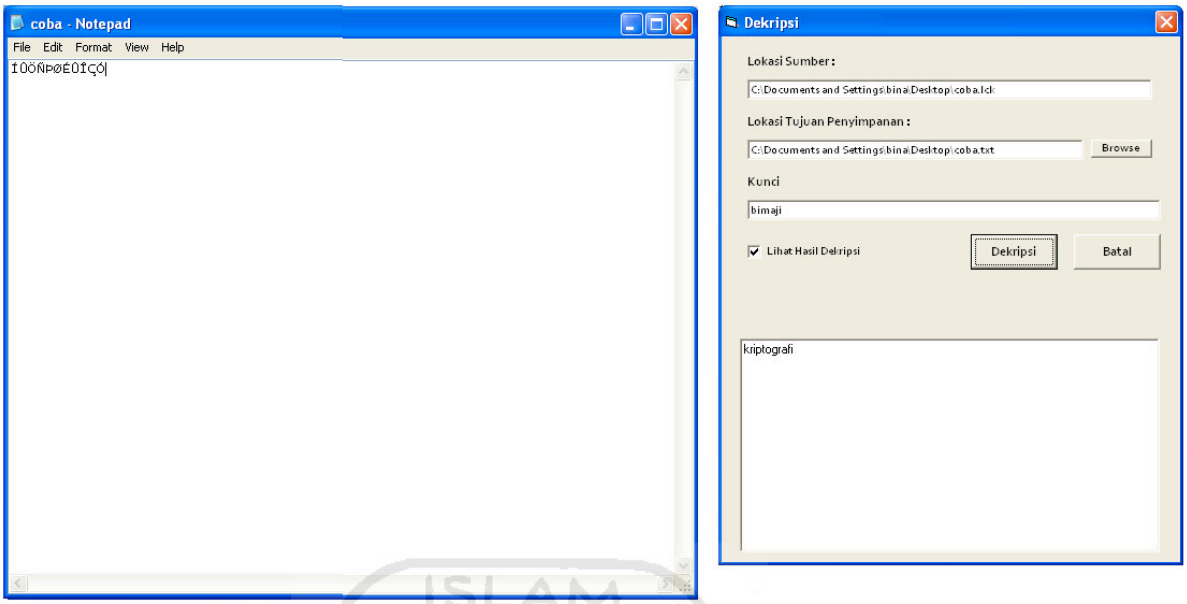
Dari hasil pengujian manual seperti di atas dengan menggunakan plainteks “kriptografi” dan kunci “bimaji”, maka dihasilkan cipherteks “ÍÛÖÑÞØÉÛÎÇÓ”. Pada proses dekripsi dengan menggunakan kunci yang sama maka dihasilkan plainteks yang sama yaitu “kriptografi”.

Untuk memastikan sistem sudah berjalan sesuai dengan kebutuhan maka pengujian juga akan dilakukan dengan menggunakan program. Pengujian menggunakan plainteks berupa file teks dengan nama *coba.txt*, file ini berisi sebuah kata “kriptografi”, kemudian file ini akan dienkripsi dan didekripsi menggunakan kunci “bimaji”.



Gambar 4.11 Pengujian enkripsi menggunakan sistem

Berdasarkan hasil enkripsi menggunakan program, cipherteks yang dihasilkan adalah “ÍÛÖÑÞØÉÛÎÇÓ”, sama dengan hasil pengujian yang dilakukan secara manual sebelumnya. Selanjutnya file tersebut didekripsi menggunakan kunci yang sama, menghasilkan plainteks yang sesuai dengan hasil pengujian manual sebelumnya dan sesuai dengan data aslinya.



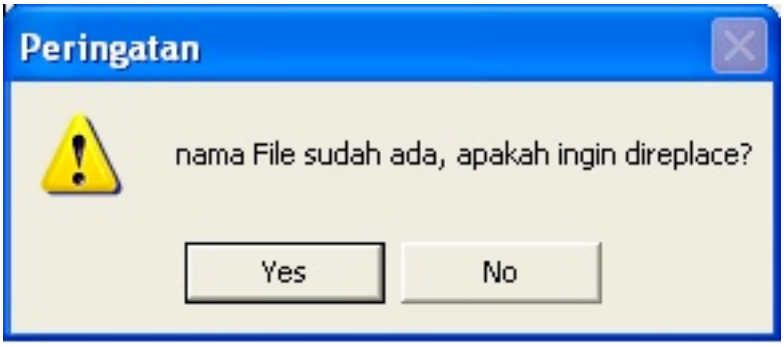
Gambar 4.12 Pengujian dekripsi menggunakan sistem

Penanganan kesalahan pada proses enkripsi dan dekripsi diletakkan pada tombol Enkripsi pada form enkripsi dan tombol Dekripsi pada form dekripsi pesan kesalahan akan keluar dengan kondisi seperti di bawah ini :

1. Kata kunci belum terisi (gambar 4.).
2. Terdapat duplikat nama file, nama file sudah pernah digunakan sebelumnya (gambar 4.).

Berikut ini adalah tampilan-tampilannya :

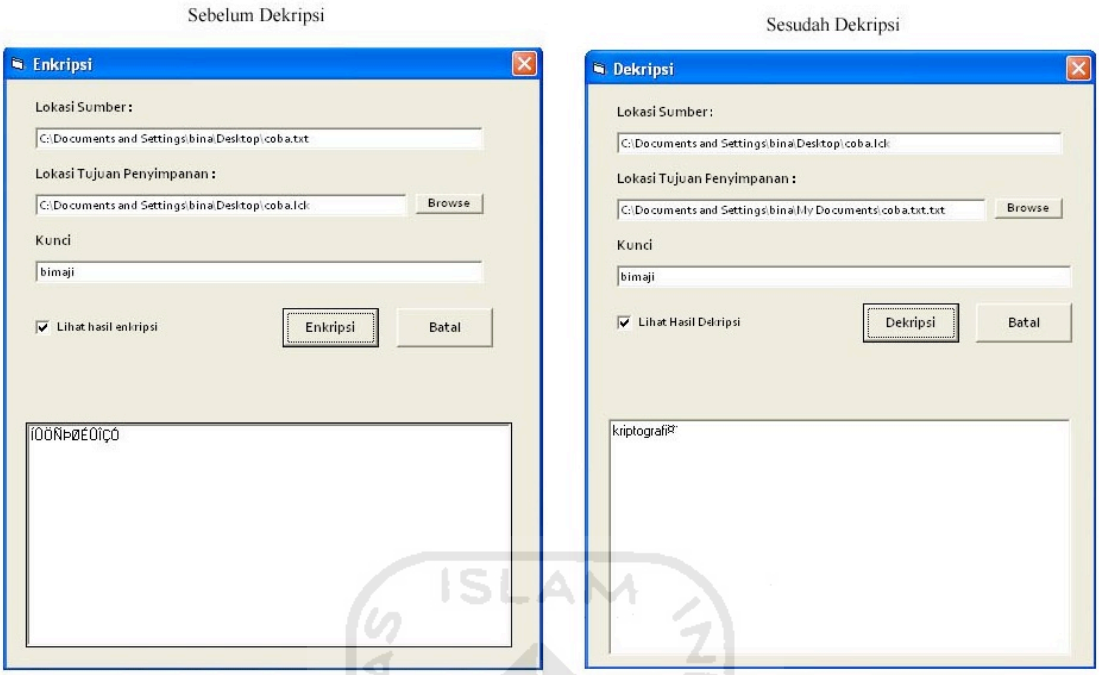
Gambar 4.13 Pesan kesalahan kata kunci belum terisi



Gambar 4.14 Pesan kesalahan duplikasi nama file

Metode pengambilan file yang digunakan selama proses enkripsi dan dekripsi adalah dengan menggunakan metode *binary*, yaitu dilakukan per satuan kilobyte data, kelebihan metode ini adalah mampu mempercepat proses enkripsi dan dekripsi, pemrosesan data akan berjalan dengan lebih cepat dan lebih ringan. Namun metode ini memiliki kekurangan yaitu akan selalu ada tambahan karakter di akhir. Untuk file gambar pertambahan karakter ini tidak begitu berpengaruh, namun untuk file teks akan sangat terlihat, karena di setiap akhir isi pesan pasti akan ada tambahan karakter.

Dari contoh kasus diatas telah ditentukan plainteks berupa data teks dengan isi pesan “kriptografi”, kemudian dienkripsi menggunakan kunci “bimaji” menghasilkan cipherteks “ÍÛÖÑÞØÉÛÎÇÓ”. Cipherteks tersebut kemudian didekripsi menggunakan kunci yang sama dan menghasilkan plainteks “kriptografi¤¨”. Terdapat kelebihan 2 karakter yaitu karakter “¤” dan karakter “¨”.

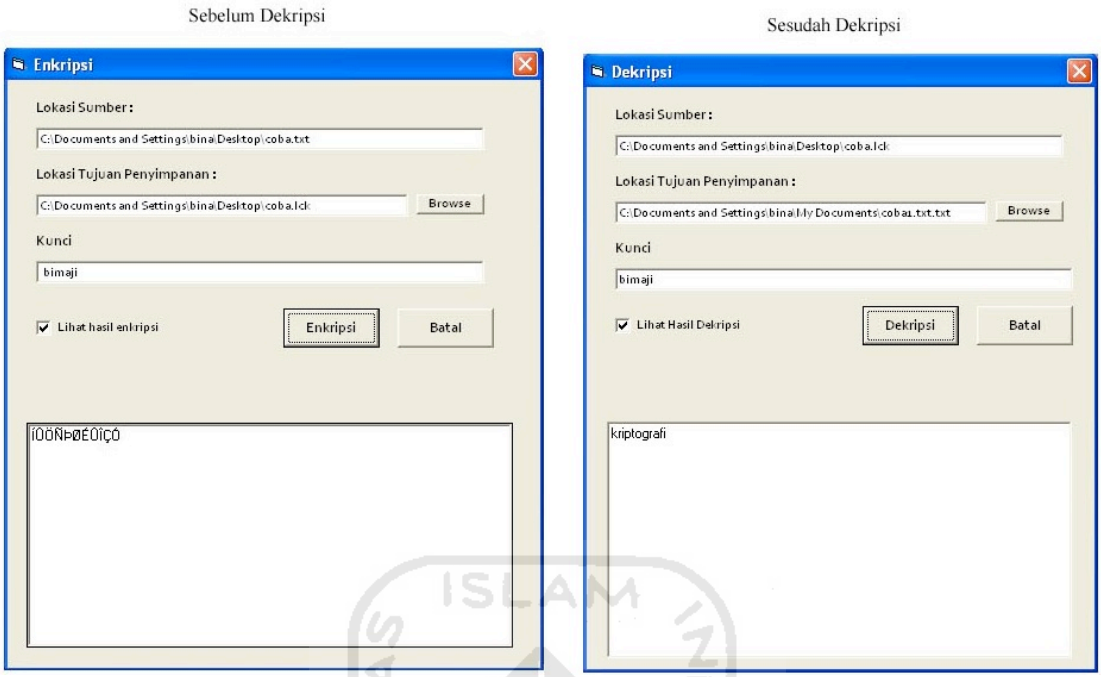


Gambar 4.15 Tampilan kelebihan karakter

Untuk menangani kondisi tersebut maka ditambahkan *listing code* untuk menghilangkan kelebihan karakter tersebut. Tujuannya adalah untuk mengembalikan isi file seperti semula, sehingga tidak ada tambahan karakter yang mungkin menimbulkan kerancuan.

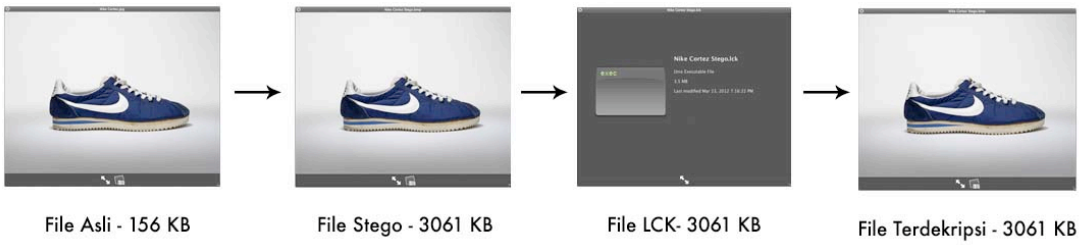
```
txtEnkripsi.Text=Mid(txtEnkripsi.Text,1,Len(txtEnkripsi.Text)
-2)
```

Setiap proses dekripsi selesai maka isi dari file tersebut akan dikurangi dengan 2 karakter, karena sebagian besar karakter tambahan yang dihasilkan adalah sebanyak 2 karakter.



Gambar 4.16 Tampilan pengurangan kelebihan karakter

Pengujian juga dilakukan terhadap file steganografi, file gambar berisikan pesan yang disembunyikan, dalam pengujian ini digunakan file gambar *Nike Cortez.jpeg* dengan pesan teks tersembunyi yang berisi "bimaji hernowo". Pengujian ini bertujuan untuk memastikan pesan yang tersembunyi tersebut tidak akan hilang atau rusak selama proses enkripsi. Aplikasi steganografi yang digunakan adalah aplikasi Quick Stego.



Gambar 4.17 Ilustrasi pengujian dengan steganografi

Untuk menggunakan aplikasi steganografi, pertama kali diharuskan memilih file gambar yang akan digunakan untuk menyamarkan pesan

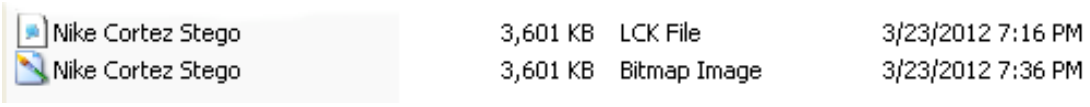
rahasia, kemudian menentukan file pesan yang akan disembunyikan didalamnya.

Gambar 4.18 Tampilan aplikasi steganografi

Output dari aplikasi ini berupa file gambar baru dengan ekstensi .bmp, ukuran filenya pun berubah, dari yang semula 153 KB menjadi 3061 KB, hampir 20 kali lipat dari file aslinya.

Gambar 4.19 Perbandingan ukuran file sebelum dan sesudah steganografi

Kemudian file hasil steganografi tersebut dienkripsi dan selanjutnya didekripsi menggunakan sistem kriptografi.

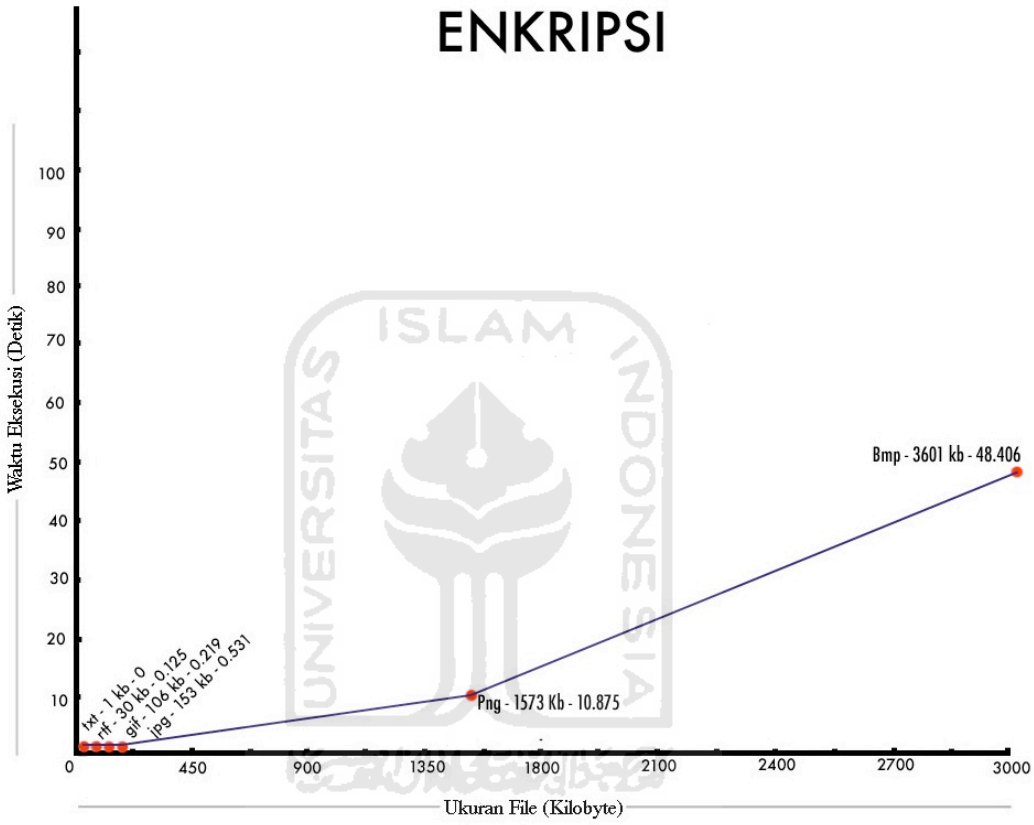


Gambar 4.20 Perbandingan file sebelum dan sesudah didekripsi

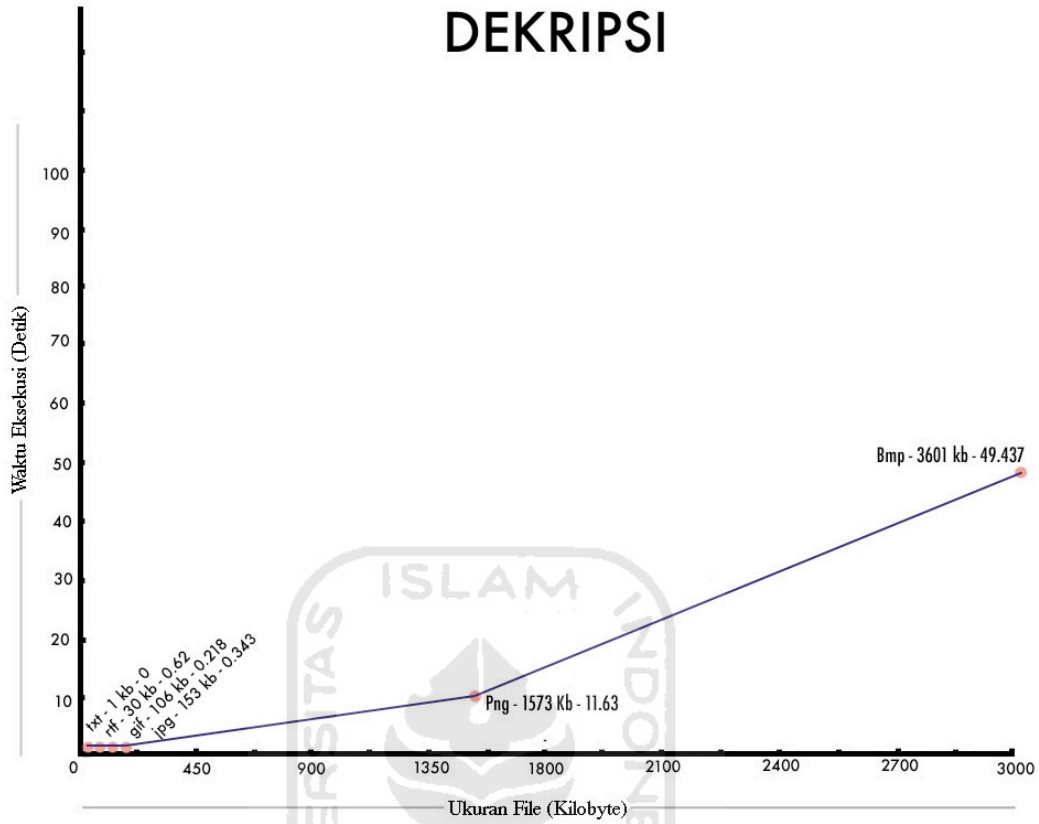
Untuk memastikan pesan yang tersembunyi tersebut tidak hilang atau rusak, maka digunakan lagi aplikasi steganografi tadi. Berdasarkan hasil pengujian ternyata pesan yang tadi disembunyikan tidak hilang atau rusak, masih tetap dapat dimunculkan dan isinya pun tetap sama.

Gambar 4.21 Tampilan pengujian file steganografi setelah dekripsi

Pengujian juga dilakukan untuk mengetahui kebutuhan waktu selama proses berjalan, sistem memiliki 2 proses utama yaitu proses enkripsi dan dekripsi. Tercatat bahwa kebutuhan waktu antara proses enkripsi dan dekripsi memiliki porsi waktu yang tidak jauh berbeda.



Gambar 4.22 Grafik kebutuhan waktu enkripsi



Gambar 4.23 Grafik kebutuhan waktu dekripsi

Pengujian waktu juga digunakan pada file dengan ukuran yang sama namun dengan jenis yang berbeda, pada pengujian ini diuji 2 buah file gambar GIF dan PNG dengan resolusi yang sama 480 pixel x 640 pixel dan dengan ukuran yang hampir sama pula yaitu 115 KB dan 121 KB. Hasil pengujian menunjukkan bahwa walaupun ukuran file dan resolusi gambar sama namun kebutuhan waktu antara kedua file tersebut berbeda, tercatat untuk enkripsi dan dekripsi file GIF membutuhkan waktu 0 detik 281 milidetik dan 0 detik 265 milidetik, sedangkan untuk file JPEG dibutuhkan waktu 0 detik 250 milidetik dan 0 detik 234 milidetik. Sementara untuk file teks tidak ada perbedaan waktu, file TXT dan RTF dengan ukuran file yang sama membutuhkan waktu yang sama pula.

BAB V
SIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan dan evaluasi dari bab-bab terdahulu dan teori yang ada, dan setelah dilakukannya penelitian dan pengujian, maka penulis dapat mengambil kesimpulan sebagai berikut :

1. Proses enkripsi dan dekripsi dilakukan per 1 *kilobyte* data untuk mempercepat proses perulangan dan digunakan untuk penggunaan *progress bar*.
2. Program kriptografi yang dibuat dapat melakukan enkripsi dan dekripsi berkas gambar (*.jpg, *.bmp, *.png, *.gif), berkas teks (*.txt, *.rtf), dan berkas terenkripsi (*.lck).
3. Kelemahan program ini adalah hanya dapat menyamarkan isi dari file asli tanpa menyamarkan ukuran file aslinya.

5.2 Saran

Saran yang dapat diberikan untuk pengembangan program ini adalah :

1. Mengembangkan penggunaan metode untuk meningkatkan keamanan, contohnya dengan menggunakan lebih dari 1 metode kriptografi.
2. Memperbanyak dukungan file
3. Dari kelemahan dalam enkripsi file dapat dibuat metode penyamaran ukuran file, sehingga bukan hanya isi filenya saja yang disamarkan.

DAFTAR PUSTAKA

Ariyus, Dony. 2008. *Pengantar Ilmu kriptografi: Teori, Analisis, dan Implementasi*.Yogyakarta: penerbit Andi Yogyakarta**.**

Munir, Rinaldi. *Kriptografi*. Program Studi Teknik Informatika. 2006.