KRIPTOGRAFI MENGGUNAKAN METODE VIGNERE CHIPER

LAPORANTUGASAKHIR

OLEH

AGUNG TRIWIBOWO

06.203.0831

PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2011

KRIPTOGRAFI MENGGUNAKAN METODE VIGNERE CHIPER

LAPORAN TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana S1

Pada Prodi Teknik Elektro Fakultas Teknologi Industri Universitas Islam

Sultan Agung Semarang

Disusun Oleh:

Agung Triwibowo

06.203.0831

JURUSANTEKNIKELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM SULTAN AGUNG

SEMARANG

2011

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul " Kriptografi Klasik Menggunakan Metode Vignere Cipher" ini disusun oleh :

Nama : Agung Triwibowo

NIM : 062030831

Program studi : Teknik Elektro

Telah disahkan dan disetujui oleh dosen pembimbing pada :

Hari :

Tanggal :

Pembimbing I	Pembimbing II
Ir. Ida Widihastuti, MT	Bustanul Arifin, ST, MT

Mengetahui,

Ka. Prodi. Teknik Elektro

Agus Suprajitno, ST, MT

FAKULTAS TEKNOLOGI INDUSTRI
Universitas Islam Sultan Agung (UNISSULA)
Jl. Raya kaligawe Km.4 Telp. 024-6583584 Psw. 340 Faks. 024-6582455
Semarang 50112 http://www.unissula.ac.id

LEMBARPENGESAHANPENGUJI

Laporan Tugas Akhir dengan judul "Kriptografi Menggunakan Metode Vignere Cipher" ini telah dipertahankan didepan Penguji Sidang Tugas Akhir pada :

Hari :

Tanggal :

Tim penguji	Tanda tangan
Eka Nuryanto Bs, ST, MT Ketua	.....
M.Khosyi'in,ST,MT Anggota 1	.....
Ir. Agus Adhi Nugroho,MT Anggota 2	.....

MOTTO DAN PERSEMBAHAN

Motto

- *"Sebaik-baiknya orang adalah orang yang berguna untuk orang lain"*
- *"Sebelum mati peliharalah nama baik, karena nama baik adalah umur manusia yang kedua"*
- *"Ilmu yang tidak diamalkan membahayakan bagi orang yang mempunyai ilmu dan beramal tanpa dilandasi ilmu adalah sesat (menyesatkan bagi yang beramal)"*
- *"kerja anda hari ini menentukan hidup anda hari esok"*

Persembahan

Dengan penuh rasa syukur atas segala restunya, penulis persembahkan kepada :

1. Bapak dan ibu tercinta
2. Kakak dan adik tersayang

KATA PENGANTAR

Assalamu'alaikum. Wr. Wb.

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan taufiq, hidayah dan inayah-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir. Laporan ini disusun untuk memenuhi salah satu syarat yang harus dipenuhi oleh mahasiswa dalam menyelesaikan studi di Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Sultan Agung (UNISSULA) Semarang.

Penulis menyadari bahwa dalam penyusunan laporan ini tidak terlepas dari bantuan, bimbingan dan motivasi berbagai pihak. Penulis ingin mengucapkan rasa terima kasih kepada :

1. Bapak Ir. Sukarno selaku Dekan Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Sultan Agung (UNISSULA) Semarang.
2. Bapak Dedi Nugroho, ST, MT selaku Ketua Program Studi Jurusan Teknik Elektro Fakultas Teknologi Industri UNISSULA.
3. Ibu Ir. Ida Widihastuti, MT selaku Dosen Pembimbing I.
4. Bapak Bustanul Arifin, ST, MT selaku Dosen Pembimbing II.
5. Seluruh staf Dosen dan karyawan Fakultas Teknologi Industri UNISSULA untuk segala masukan dan bantuannya selama ini.
6. Bapak Syamian, SH, MH yang selalu menasehati dan memberi masukan.
7. Cumba Nugraha yang telah membantu dalam pembuatan laporan
8. Teman-temanku seperjuangan Mahasiswa Jurusan Teknik Elektro angkatan 2003 yang telah memberikan bantuan material maupun spiritual dan motivasi.
9. Semua pihak yang tidak bisa disebutkan satu persatu terima kasih, semoga Tuhan Yang Maha Esa memberikan yang terbaik.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan laporan ini, maka penulis mengharapkan saran dan kritik yang membangun untuk menyempurnakan laporan ini.

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi kita semua, amin.

Wassalamu'alaikum. Wr. Wb.

Semarang, September 2011

Penulis

DAFTARISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Kemajuan teknologi dibidang komputer memungkinkan ribuan orang dan komputer diseluruh dunia terhubung dalam satu dunia maya dikenal sebagai *cyberspace*atau internet. Kemajuan teknologi selalu diikuti dengan sisi buruk, salah satunya keamanan data. Kriptografi merupakan pengamanan data dengan cara data yang dikirimkan melalui jaringan akan disamarkan sedemikian rupa dengan bantuan kunci. Kriptografi *vigenere cipher* merupakan *cipher alfabet* majemuk terbaik, dimana plainteks akan dienkripsi dengan pergeseran karakter tetapi setiap karakter di dalam plainteks akan mengalami pergeseran yang berbed. Pergeseran karakter pada plainteks akan di tentukan oleh kunci yang mempunyai posisi sama dengan karakter pada plainteks. Hasil kriptografi harus menggunakan kunci yang sama untuk menghasilkan plainteks sebenarnya.

Pembuatan program simulasi kriptografi *vigenere cipher* menggunakan visual basic yang merupakan pemrograman berorientasi objek sehingga hasil akhir pemrograman dapat di compile menjadi **.exe*. Kriptografi *vigenere cipher* membutuhkan tabel vigenere berisi semua karakter huruf, angka dan simbol.

Kriptografi menggunakan *vigenere cipher* membutuhkan kunci untuk mengubah karakter didalam plainteks dan bekerja efektif untuk ukuran palinteks lebih besar dari kunci. Kriptografi menggunakan *vigenere cipher* dapat diterapkan untuk jumlah karakter lebih banyak, sehingga didapat variasi kunci yang lebih banyak.

Vignere Cipher dirancang dengan bahasa pemrograman Visual Studio. Proses pengujian dilakukan pada enkripsi (dengan menggeserkan huruf) dan dekripsi (mengembalikan sandi seperti semula). Pada pengujian aplikasi ini proses yang dilakukan dengan cara menggeser huruf, angka dan simbol sesuai yang diinginkan. Input data yang digunakan berupa tulisan teks dan tipe file txt. Presentase keamanan pada kriptografi ini sebesar 100% jadi bisa dikatakan bahwa metode ini sangat aman.

Kata kunci : plainteks, kunci, cipherteks dan tabel vigenere

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi di bidang komputer memungkinkan ribuan orang dan komputer di seluruh dunia terhubung dalam satu dunia maya yang dikenal sebagai *cyberspace* atau Internet. Begitu juga ratusan organisasi seperti perusahaan, lembaga negara, lembaga keuangan, militer dan sebagainya. Tetapi sayangnya, kemajuan teknologi selalu diikuti dengan sisi buruk dari teknologi itu sendiri. Salah satunya adalah rawannya keamanan data sehingga menimbulkan tantangan dan tuntutan akan tersedianya suatu sistem pengamanan data yang sama canggihnya dengan kemajuan teknologi komputer itu sendiri.

Ada beberapa cara melakukan pengamanan data yang melalui suatu saluran, salah satu diantaranya adalah kriptografi. Dalam kriptografi, data yang dikirimkan melalui jaringan akan disamarkan sedemikian rupa sehingga kalaupun data itu bisa dibaca maka tidak bisa dimengerti oleh pihak yang tidak berhak. Data yang akan dikirimkan dan belum mengalami penyandian dikenal dengan isitilah *plaintext*, dan setelah disamarkan dengan suatu cara penyandian, maka plaintext ini akan berubah menjadi *ciphertext*.

Secara umum algoritma kriptografi klasik dikelompokkan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma substitusi (*substitution cipher*). Cipher transposisi mengubah susunan huruf-huruf di dalam pesan, sedangkan cipher substitusi mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain.Salah satu kriptografi klasik adalah vignere, huruf-huruf dalam plainteks berbeda, karena sudah diganti. Dengan kata lain algoritma ini melakukan substitusi terhadap rangkaian karakter di dalam teks.

Berdasar latar belakang masalah diatas, maka pada Tugas Akhir ini akan diketahui bagaimana proses kriptografi vigenere cipher dapat mengkriptografi suatu pesan. algoritma vigenere chiper ini diaplikasikan dalam bahasa pemrograman visual basic 6.0.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang ada dapat dirumuskan beberapa permasalahan yang timbul, yakni:

1. Bagaimana proses kriptografi menggunakan vigenere cipher.
2. Mensimulasikan kriptografi vigenere chiper menggunakan pemrograman Visual basic 6.0.

1.3 Pembatasan Masalah

Pada penulisan tugas akhir " kriptografi menggunakan vigenere cipher" penulis membatasi masalah pada:

1. Proses kriptografi menggunakan vigenere cipher dengan pemrograman visual basic 6.0.
2. Analisis hasil kriptografi menggunakan Vigenere cipher.

1.4 Tujuan

Adapun tujuan dari penulisan tugas akhir ini adalah :

1. Mengetahui bagaimana proses kriptografi menggunakan vigenere cipher dapat menjaga keamanan suatu pesan.
2. Mengetahui bentuk pesan yang dapat diproses dengan kriptografi menggunakan vigenere chiper.

1.5 Manfaat

Manfaat yang dapat diperoleh dalam penulisan tugas akhir ini adalah :

1. Kriptografi vigenere cipher dapat menjaga kerahasiaan pesan.
2. Kunci pada kriptografi vigenere cipher menjamin integritas pesan.

1.6 Metodelogi Penelitian

Metodelogi yang digunakan untuk penulisan tugas akhir ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur untuk memperoleh informasi dengan cara membaca buku-buku, mempelajari jurnal dan juga dapat berupa cuplikan data dari internet untuk mendukung penulisan tugas akhir.

2. Perancangan sistem

Merupakan metode untuk mengimplementasikan ide yang ditemukan beserta teori yang melandasinya. Program yang digunakan dalam tugas akhir ini adalah visual basic 6.0, membuat simulasi tentang proses kriptografi menggunakan vigenere cipher.

3. Pengujian sistem

Metode untuk menguji hasil dari perancangan sistem yang dibuat agar sistem bekerja dengan baik.

4. Analisa sistem

Analisa pengujian sistem yang dilakukan dengan berbagai variasi parameter pesan.

1.7 Sistematika penulisan

Untuk mempermudah dalam penyusunan tugas akhir ini, maka penulis menyusun sistematika tugas akhir ini sebagai berikut :

BAB I PENDAHULUAN

Berisi tentang judul, latar belakang, perumusan masalah, pembatasan masalah, maksud dan tujuan, manfaat, metodologi penulisan dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini berisi penjelasan mengenai teori-teori yang menunjang kriptografi vigenere cipher.

BAB III PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai perancangan dari program simulasi sistem kriptografi vignere cipher beserta langkah-langkah dalam pembuatan program simulasi yang disertai dengan sedikit gambaran tentang bahasa pemrograman visual basic 6.0.

BABIV IMPLEMENTASI SISTEM

Pada bab ini berisikan tentang pengujian dan analisa hasil simulasi sistemkriptografi vigenere cipher.

BABV KESIMPULAN

Berisi kesimpulan dan saran sebagai hasil dari pembahasan tugas akhir.

BAB II

LANDASAN TEORI

2.1 Definisi Kriptografi

Transmisi pesan merupakan proses pertukaran pesan antara pengirim dan penerima, pengirim tentu ingin pesan yang dikirim sampai kepihak yang dituju dengan aman. Pengertian aman di sini sangat luas. Aman bisa berarti bahwa selama pengiriman pesan tentu pengirim berharap pesan tersebut tidak dibaca oleh orang yang tidak berhak. Sebab, mungkin saja pesan yang dikirim berisi sesuatu yang rahasia sehingga jika pesan rahasia dibaca oleh pihak lawan atau pihak yang tidak berkepentingan, maka bocorlah kerahasiaan pesan yang dikirim. Ini adalah masalah keamanan pesan yang dinamakan kerahasiaan (*confidentiality* atau *privacy*).

Aman bisa juga berarti bahwa pesan yang dikirim sampai dengan utuh ke tangan penerima, artinya isi pesan tidak diubah atau dimanipulasi selama pengiriman oleh pihak ketiga. Di sisi penerima pesan, ia tentu ingin memastikan bahwa pesan yang ia terima adalah pesan yang masih asli, bukan pesan yang sudah ditambah atau dikurangi. Ini adalah masalah keamanan pesan yang disebut integritas data (*data integrity*). Selain itu, penerima yakin bahwa pesan tersebut memang benar berasal dari pengirim, bukan dari orang lain yang menyamar seperti pengirim. Pengirim pun yakin bahwa orang yang dikirimi pesan adalah orang yang sesungguhnya. Ini adalah masalah keamanan pesan yang dinamakan otentikasi *(auhentication)*.

Penerima pesan pun tidak ingin kelak pengirim pesan membantah pernah mengirim pesan. Ini adalah masalah keamanan yang disebut penyangkalan *(repudiaton)*. Zaman sekarang banyak orang yang membantah telah mengirim atau menerima pesan. Padahal anda yakin bahwa penerima memang menerima pesan dari orang tersebut. Jika pengirim membantah telah mengirim pesan, maka penerima perlu membuktikan ketidakbenaran penyangkalan tersebut *(non-repudiation)*.

Keempat masalah keamanan yang disebutkan di atas, yaitu kerahasiaan, integritas data, otentikasi, dan penyangkalan dapat diselesaikan dengan menggunakan kriptografi. Kriptografi tidak hanya menyediakan alat untuk keamanan pesan, tetapi juga sekumpulan teknik yang berguna.

Kriptografi (*cryptography*) berasal dari Bahasa Yunani: "*cryptós*" artinya "*secret*" (rahasia), sedangkan "*graphein*" artinya "*writing*" (tulisan). Jadi, kriptografi berarti "*secret writing*" (tulisan rahasia). Definisi yang dipakai di dalam buku-buku lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Namun saat ini kriptografi lebih dari sekadar *privacy*, tetapi juga untuk tujuan *data integrity*, *authentication*, dan *non-repudiation*.

Kata ''seni'' di dalam definisi di atas berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Cara-cara unik tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga pesan mempunyai nilai estitika tersendiri. Kriptografi berkembang menjadi sebuah seni merahasiakan pesan. Pada perkembangan selanjutnya, kriptografi berkembang menjadi sebuah disiplin ilmu tersendiri karena teknik-teknik kriptografi dapat diformulasikan secara matematik sehingga menjadi sebuah metode yang formal (Rinaldi Munir, 2006).

2.2 Terminologi

Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminologi. Beberapa istilah penting untuk diketahui, yaitu :

2.2.1 Pesan, plainteks dan cipherteks

Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data,dsb) atau yang disimpan di dalam media perekaman (Dony Ariyus, 2006)

Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks (*plaintext*) atau teks-jelas (*cleartext*). Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman

(kertas, *storage*, dsb). Pesan yang tersimpan tidak hanya berupa teks, tetapi juga dapat berbentuk citra (*image*), suara bunyi (*audio*) dan video, atau berkas biner lainnya.

Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut cipherteks (*ciphertext*) atau kriptogram (*cryptogram*). Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca. Gambar 2.1 memperlihatkan contoh dua buah plainteks, masing-masing berupa teks dan gambar, serta cipherteks yang berkoresponden. Perhatikan bahwa plainteks dapat dibaca dengan jelas, tetapi cipherteks sudah tidak dapat lagi dimengerti maknanya. Melalui proses yang berkebalikan, cipherteks dapat ditransformasikan kembali menjadi plainteks semula.

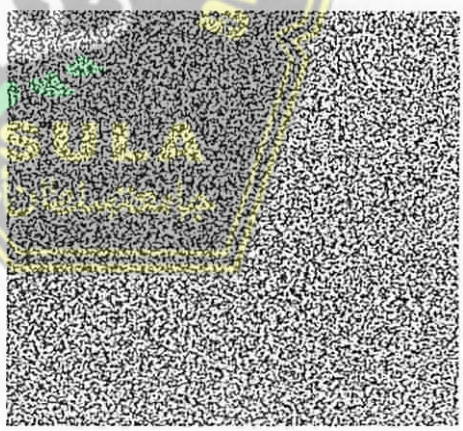
```
Ketika saya berjalan-jalan di
pantai, saya menemukan banyak
sekali kepiting yang merangkak
menuju laut. Mereka adalah
anak-anak kepiting yang baru
menetas dari dalam pasir.
Naluri mereka mengatakan bahwa
laut adalah tempat kehidupan
mereka.
```

(a) Plainteks (teks)

```
Ztâxzp/épêp/qtüyp{p)<yp{p}/sx/ p}â
px;
épêp/|t)t|äzp}/qp}êpz/étzp{x/zt
xâx
)v épêp}v/|tüp}vzpz/t}äyä/{pää=/\t
ütz
p {psp{pw/p}pz<p}pz/zt |xâx}v/êp}
v/qpüä t)tâpé/spüx/sp{p|/ péxü=/
]
p{äüxd ttüzp/|t}vpâpzp}/qpwâp/{pä
â
/psp{pw ât pâ/ztwxsä p}/|tützp=
```

(b) Cipherteks dari (a)

(c) Plainteks (citra)



(d) Cipherteks dari (c)

Gambar 2.1 Contoh plainteks dan cipherteks

2.2.2 Pengirim dan penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Entitas disini dapat berupa orang, mesin (komputer), kartu kredit, dan sebagainya. Dalam jaringan komputer, komputer berkomunikasi dengan mesin (contoh : mesin ATM berkomunikasi dengan komputer *server* di bank).

Pengirim tentu menginginkan pesan dapat dikirim secara aman, yaitu ia yakin bahwa pihak lain tidak dapat membaca isi pesan yang ia kirim. Solusinya adalah dengan cara menyandikan pesan menjadi cipherteks.

2.2.3 Enkripsi dan dekripsi

Proses menyandikan plainteks menjadi cipherteks disebut enkripsi *(encryption)* atau *enciphering* (standard nama menurut ISO 7498-2). Sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi *(decryption)* atau *deciphering* (standard nama menurut ISO 7498-2). Enkripsi dan dekripsi dapat diterapkan baik pada pesan yang dikirim maupun pada pesan tersimpan. Istilah *encryption of data in motion* mengacu pada enkripsi pesan yang ditransmisikan melalui saluran komunikasi, sedangkan istilah *encryption of data at-rest* mengacu pada enkripsi dokumen yang disimpan di dalam *storage*. Contoh *encryption of data in motion* adalah pengiriman nomor PIN dari mesin ATM ke komputer *server* di kantor bank pusat.Contoh *encryption of data at—rest* adalah enkripsi *file* basis data di dalam hard disk.

2.2.4 Cipher dan kunci

Algoritma kriptografi disebut juga cipher yaitu aturan untuk *enciphering* dan *deciphering*, 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 kedua himpunan tersebut. Misalkan P menyatakan plainteks dan C menyatakan cipherteks, maka fungsi enkripsi E memetakan P ke C.

$$E(P)=C \tag{2.1}$$

dan fungsi dekripsi D memetakan C ke P,

$$D(C) = P \tag{2.2}$$

karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka kesamaan berikut harus benar,

$$D(E(P))=P \tag{2.3}$$

Keamanan algoritma kriptografi sering diukur dari banyaknya kerja (*work*) yang dibutuhkan untuk memecahkan cipherteks menjadi plainteksnya tanpa mengetahui kunci yang digunakan. Kerja ini dapat diekivalenkan dengan waktu, memori, uang, dan lain-lain. Semakin banyak kerja yang diperlukan, yang berarti juga semakin lama waktu yang dibutuhkan, maka semakin kuat algoritma kriptografi tersebut, yang berarti semakin aman digunakan untuk menyandikan pesan.

Keamanan kriptografi ditentukan dengan menjaga kerahasiaan algoritmanya, maka algoritma kriptografinya dinamakan algoritma *restricted*. Algoritma *restricted* mempunyai sejarah tersendiri di dalam kriptografi. Algoritma *restricted* biasanya digunakan oleh sekelompok orang untuk bertukar pesan satu sama lain. Mereka membuat suatu algoritma enkripsi dan algoritma enkripsi tersebut hanya diketahui oleh anggota kelompok itu saja. Tetapi, algoritma *restricted* tidak cocok lagi saat ini, sebab setiap kali ada anggota kelompok keluar, maka algoritma kriptografi harus diganti lagi.

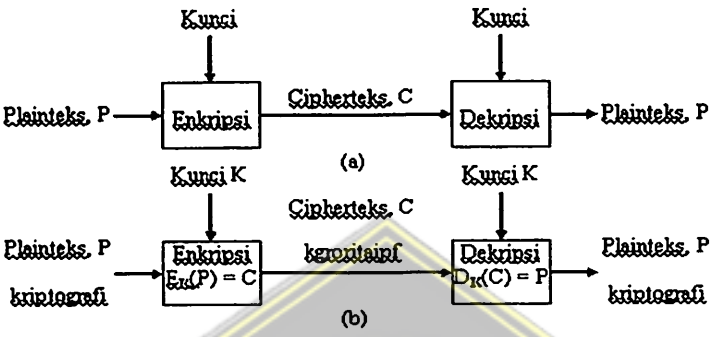
Kriptografi modern mengatasi masalah di atas dengan penggunaan kunci, yang dalam hal ini algoritma tidak lagi dirahasiakan, tetapi kunci harus dijaga kerahasiaannya. Kunci (key) adalah parameter yang digunakan untuk transformasi *enciphering* dan *deciphering*. Kunci biasanya berupa *string* atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai

$$E_k(P) = C \quad \text{dan} \quad D_k(C) = P \tag{2.4}$$

dan kedua fungsi ini memenuhi

$$D_k(E_k(P)) = P \tag{2.5}$$

Gambar 2.2(a) memperlihatkan skema enkripsi dan dekripsi dengan menggunakan kunci, sedangkan gambar 2.2(b) mengilustrasikan enkripsi dan dekripsi terhadap sebuah pesan.



Gambar 2.2 (a) Skema enkripsi dan dekripsi, (b) Contoh ilustrasi enkripsi dan dekripsi pesan

Istilah "cipher" sering disamakan dengan kode (*code*). Kode mempunyai sejarah tersendiri di dalam kriptografi. Sebenarnya kedua istilah ini tidak sama pengertiannya. Jika cipher adalah transformasi karakter-ke-karakter atau bit-ke-bit tanpa memperhatikan strukur bahasa pesan, maka kode sering diacu sebagai prosedur yang mengganti setiap plainteks dengan kata kode, misalnya

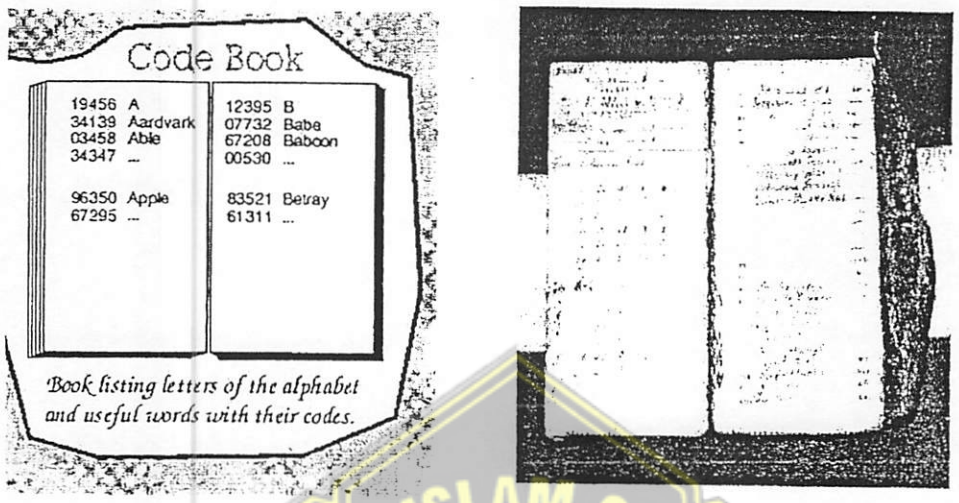
"kapal api datang dikodekan menjadi hutan bakau hancur"

Kode juga dapat berupa deretan angka dan huruf yang tidak bermakna, seperti

"kapal api datang dikodekan menjadi xyztvq bkugbf hjqpot"

Transformasi dari plainteks menjadi kode sering disebut *encoding*, sedangkan transfonrnasi kebalikannya sering disebut decoding. Di dalam kriptografi, buku kode (*codebook*) -contohnya seperti pada Gambar 2.3- adalah dokumen yang digunakan untuk mengimplementasikan suatu kode. Buku kode terdiri dari tabel *lookup* (*lookup table*) untuk *encoding* dan *decoding*. Untuk melakukan enkripsi dan dekripsi pesan, buku kode yang sama harus tersedia di sisi pengirim dan penerima pesan. Penyebaran buku kode menimbulkan masalah tersendiri menyangkut keamanannya, sehingga penggunaan kode di dalam

kriptografi tidak mempunyai umur yang panjang dan cipher menjadi teknik yang dominan. Pihak lawan yang mencoba mentransformasikan kode menjadi plainteks dinamakan pemecah kode (*codebreaker*).



Gambar 2.3 (a) Buku kode, (b) Sebuah buku kode yang digunakan untuk korespondensi telegraf

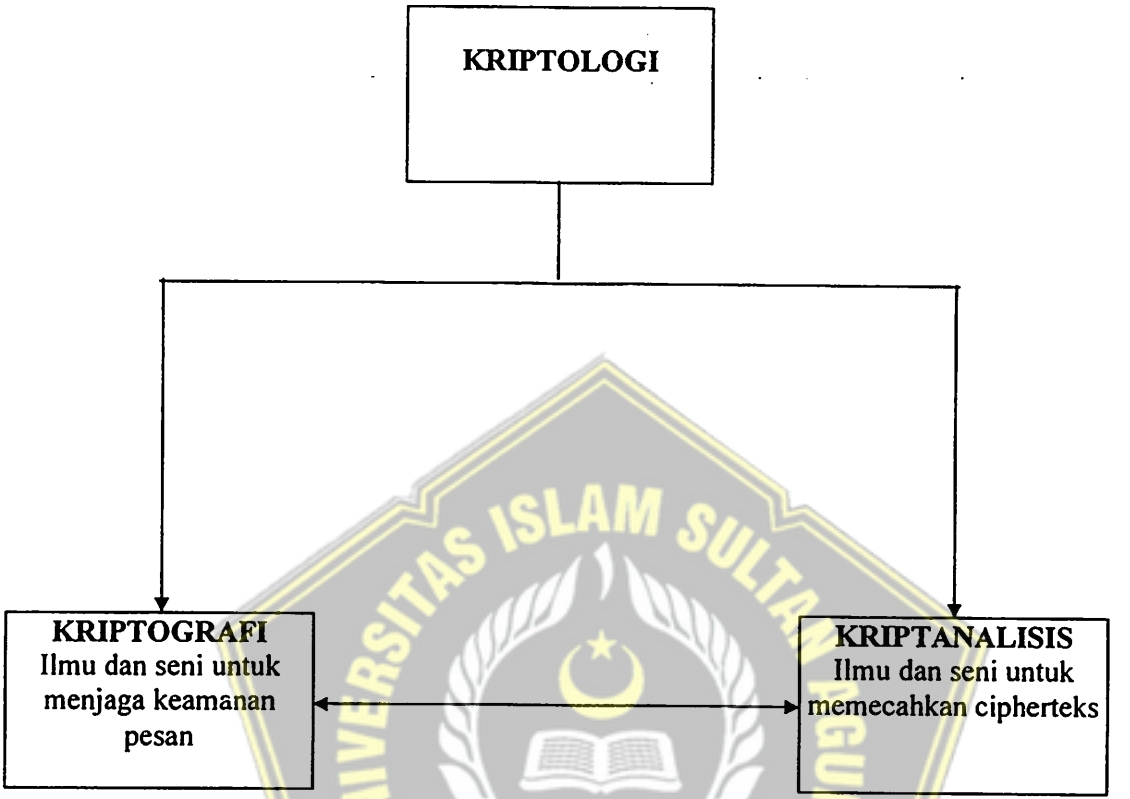
2.2.5 Sistem kriptografi

Kriptografi membentuk sebuah sistem yang dinamakan sistem kriptografi. Sistem kriptografi (*cryptosystem*) adalah kumpulan yang terdiri dari algoritma kriptografi, semua plainteks dan cipherteks yang mungkin, dan kunci. Di dalam sistem kriptografi, cipher hanyalah salah satu komponen saja.

2.2.6 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. Gambar 2.4 memperlihatkan pohon kriptologi.



Gambar 2.4 Kriptografi dan kriptanalisis adalah cabang bidang ilmu kriptografi

Sebagian para parktisi sering menggunakan istilah kriptografi dan kriptologi secara bergantian, sebagian lagi membedakan bahwa kriptografi mengacu pada penggunaan praktis teknik-teknik kriptografi, sedangkan kriptologi mengacu pada subjek.

2.3 Tujuan kriptografi

Penjelasan pada definisi kriptografi dapat disimpulkan tujuan dari kriptografi sebagai berikut :

1. Kerahasiaan (*confidentiality*)

Layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak. Di dalam kriptografi, layanan ini direalisasikan dengan menyandikan pesan menjadi cipherteks.

2. Integritas data (*data integrity*)

Layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman. Dengan kata lain "Apakah pesan yang diterima masih asli atau tidak mengalami perubahan (modifikasi)?". Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi pesan oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubstitusian data lain kedalam pesan yang sebenarnya. Di dalam kriptografi, layanan ini direalisasikan dengan menggunakan tanda-tangan digital (*digital signature*). Pesan yang telah ditandatangani menyiratkan bahwa pesan yang dikirim adalah asli.

3. Otentikasi (*authentication*)

Layanan yang berhubungan dengan identifikasi, baik mengidentifikasi 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*)

Layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

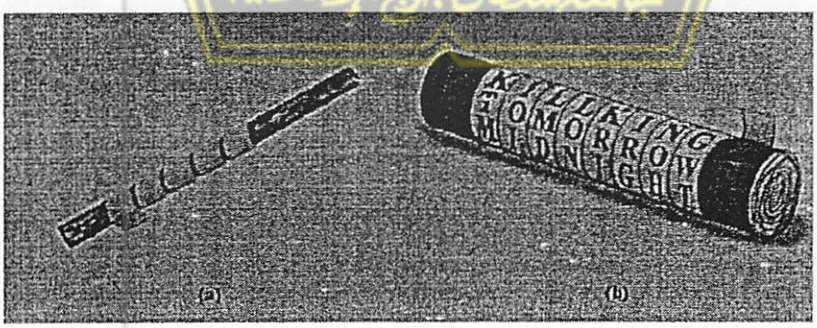
2.4 Sejarah kriptografi

Kriptografi mempunyai sejarah yang panjang. Informasi yang lengkap mengenai sejarah kriptografi dapat ditemukan di dalam buku David Kahn yang berjudul *The Codebreakers*. Buku ini menulis secara rinci sejarah kriptografi mulai dari penggunaan kriptografi oleh Bangsa Mesir 4000 tahun yang lalu (berupa *hieroglyph* yang tidak standard pada piramid) hingga penggunaan kriptografi pada abad ke-20. Secara historis ada empat kelompok orang yang berkontribusi terhadap perkembangan kriptografi, yaitu kalangan militer (termasuk intelijen dan mata-mata), kalangan diplomatik, penulis buku harian, dan

pencinta (*lovers*). Kalangan militer memberikan kontribusi paling penting, karena pengiriman pesan di dalam suasana perang membutuhkan teknik enkripsi dan dekripsi yang rumit.

Sejarah kriptografi sebagian besar merupakan sejarah kriptografi klasik, yaitu metode enkripsi yang menggunakan kertas dan pensil atau mungkin dengan bantuan alat mekanik sederhana. Secara umum algoritma kriptografi klasik dikelompokkan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma substitusi (*substitution cipher*). Cipher transposisi mengubah susunan huruf-huruf di dalam pesan, sedangkan cipher substitusi mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelornpok huruf lain. Sejarah kriptografi klasik mencatat penggunaan cipher transposisi oleh tentara Sparta di Yunani pada permulaan tahun 400 SM. Mereka menggunakan alat yang namanya *scytale* (Gambar 2.5(a)). *Scytale* terdiri dari sebuah kertas panjang dan daun *papyrus* yang dililitkan pada sebuah silinder dari diameter tertentu (diameter silinder menyatakan kunci penyandian). Pesan ditulis secara horizontal, baris per baris (lihat Gambar 2.5(b)). Bila pita dilepaskan, maka huruf-huruf di dalamnya telah tersusun secara acak membentuk pesan rahasia. Untuk membaca pesan, penerima pesan harus melilitkan kembali kertas tersebut ke silinder yang diameternya sama dengan diameter silinder pengirim.

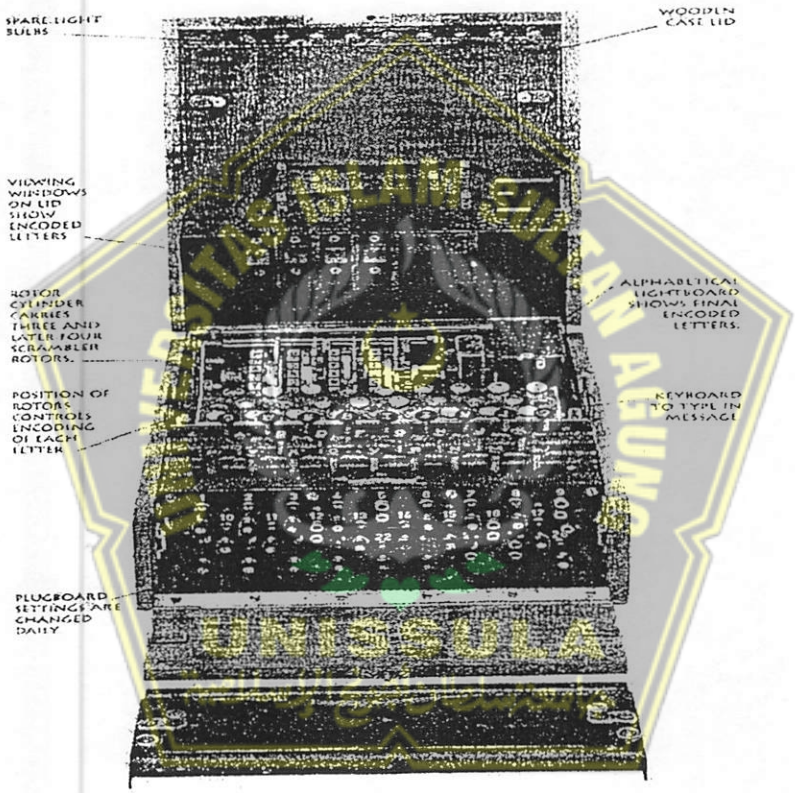
Algoritma substitusi paling awal dan paling sederhana adalah Caesar cipher, yang digunakan oleh raja Yunani kuno, Julius Caesar. Caranya adalah dengan mengganti setiap karakter di dalam alfabet dengan karakter yang terletak pada tiga posisi berikutnya di dalam susunan alfabet.



Gambar 2.5 (a) Sebuah *scytale* (b) Pesan ditulis secara horizontal baris per baris.

Kriptografi juga digunakan untuk tujuan keamanan. Kalangan gereja pada masa awal agama Kristen menggunakan kriptografi untuk menjaga tulisan relijius dari gangguan otoritas politik atau budaya yang dominan saat itu.

Kriptografi umumnya digunakan oleh kalangan militer. Pada perang dunia ke-2, pemerintah Nazi Jerman membuat mesin enkripsi yang dinamakan *enigma*. Gambar 2.6, mesin yang menggunakan beberapa buah rotor (roda berputar) ini melakukan enkripsi dengan cara sangat rumit. Namun *enigma* berhasil dipecahkan oleh pihak sekutu dan keberhasilan memecahkan *enigma* sering dikatakan sebagai faktor yang memperpendek perang dunia ke-2.



Gambar 4.9 Mesin Enigma

Gambar 2.6 Mesin enkripsi enigma

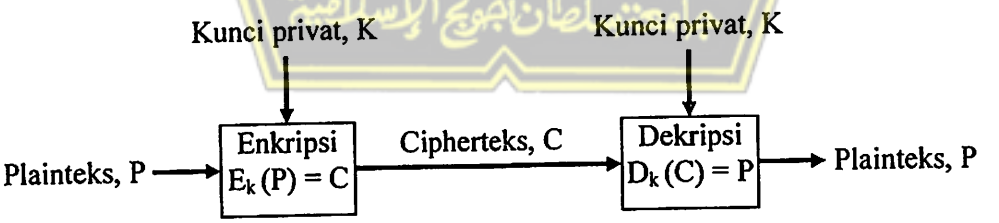
2.5 Kriptografi kunci simetri dan nirsimetri

Selain berdasarkan sejarah yang membagi kriptografi menjadi kriptografi klasik dan kriptografi modern, maka berdasarkan kunci yang digunakan untuk

enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi kriptografi kunci simetri (*symmetric key cryptography*) dan kriptografi kunci nirsimetri (*asymmetric key cryptography*).

Pada sistem kriptografi kunci simetri, kunci untuk enkripsi sama dengan kunci untuk dekripsi, oleh karena itu dinamakan kriptografi simetri (Gambar 2.7). Istilah lain untuk kriptografi kunci simetri adalah kriptografi kunci privat (*private key cryptography*), kriptografi kunci rahasia (*secret key cryptography*), atau kriptography konvensional (*conventional cryptography*). Sistem kriptografi kunci simetri mengasumsikan pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kuncinya. Kriptografi simetri merupakan satu-satunya jenis kriptografi yang dikenal dalam catatan sejarah hingga tahun 1976. Semua algoritma kriptografi klasik termasuk ke dalam sistem kriptografi simetri.

Secara umum, cipher yang termasuk ke dalam kriptografi simetri beroperasi dalam mode blok (*block cipher*), yaitu setiap kali enkripsi dan dekripsi dilakukan terhadap satu blok data yang berukuran tertentu, atau beroperasi dalam mode aliran (*stream mode*), yaitu setiap kali enkripsi atau dekripsi dilakukan terhadap satu bit atau satu *byte* data. Aplikasi kriptogarfi simetri yang utama adalah melindungi kerahasiaan data yang dikirim melalui saluran tidak aman dan melindungi kerahasiaan data yang disimpan pada media yang tidak aman. Kelemahan dari sistem ini adalah baik pengirim maupun penerima pesan harus memiliki kunci yang sama, sehingga pengirim pesan harus mencari cara yang aman untuk memberitahukan kunci kepada penerima pesan.

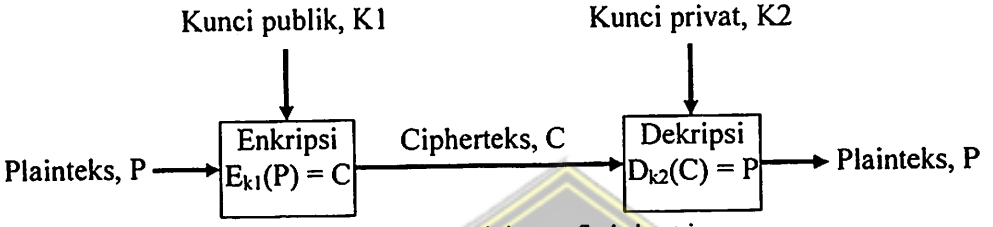


Gambar 2.7 Skema kriptografi simetri

Jika kunci untuk enkripsi tidak sama dengan kunci untuk dekripsi, maka kriptografinya dinamakan sistem kriptografi nirsimetri. Nama lainnya adalah

kriptografi kunci publik (*public key cryptography*), sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan. Pada kriptografi jenis ini, setiap orang yang berkomunikasi mempunyai sepasang kunci, yaitu kunci privat dan kunci publik.

Pengirim mengenkripsi pesan dengan menggunakan kunci publik, hanya penerima pesan yang mengetahui kunci privatnya yang dapat mendekripsikan pesan tersebut.



Gambar 2.8 Skema kriptografi nirsimetri

2.6 Algoritma Kriptografi Klasik

Kriptografi mempunyai sejarah yang panjang, mulai dari kriptografi Caesar yang berkembang pada zaman sebelum Masehi sampai kriptografi modern yang digunakan dalam komunikasi antar komputer di abad 20. Ada dua cara yang paling dasar pada kriptografi klasik., yaitu:

1. Transposisi.

   Transposisi adalah mengubah susunan huruf pada plaintext sehingga urutannya berubah. Contoh yang paling sederhana adalah mengubah suatu kalimat dengan menuliskan setiap kata secara terbalik.

   Contoh transposisi yang sedikit lebih sulit adalah plaintext yang disusun dalam kelompok huruf yangterdiri dari beberapa kolom huruf, misalnya 5 kolom huruf :

   IBUAK
   ANDAT
   ANGBE
   SOKPA
   GIAAA

Gambar 2.9 Plaintext disusun dalam 5 kolom huruf

kemudian dituliskan per kolom, dengan urutan kolom yang bisa berubah-ubah

IAASGBNNOIUDGKAAABPAKTEAA

Gambar 2.10 Kolom huruf dituliskan berurutan dari kolom 1,2,3,4,5

UDGKAKTEAAAABPAAIAASGBNNOI

Gambar 2.11 Kolom huruf dituliskan dengan urutan kolom 3,5,4,1,2

2. Substitusi

Substitusi yaitu setiap huruf pada plaintext akan digantikan dengan huruf lain berdasarkan suatu cara atau rumus tertentu, contoh:

Plaintext : IBU AKAN DATANG BESOK PAGI
Ciphertext: KDW CMCP FCVCPI DGUQM RCIK

Gambar 2.12 Substitusi dengan pergeseran 2 huruf

Algoritma kriptografi modern tidak lagi mengandalkan keamanannya pada kerahasiaan algoritma tetapi kerahasiaan kunci. Plainteks yang sama bila disandikan dengan kunci yang berbeda akan menghasilkan cipherteks yang berbeda pula. Dengan demikian algoritma kriptografi dapat bersifat umum dan boleh diketahui oleh siapa saja, akan tetapi tanpa pengetahuan tentang kunci, data tersandi tetap saja tidak dapat terpecahkan (Ilmu Komputer.Com, 2006).

2.7 Aritmetika Modulo

Misalkan a adalah bilangan bulat dan m adalah bilangan bulat > 0. Operasi a mod m (dibaca "a modulo m") memberikan sisa jika a dibagi dengan m. Bilangan m disebut modulus atau modulo, dan hasil aritmetika modulo m terletak di dalam himpunan {0, 1, 2, ….., m-1}

Notasi: a mod m = r sedemikian sehingga a= mq + r, dengan $0 \leq r < m$.

Contoh :

(i) 23 mod 5 = 3 (23 = 5. 4 + 3)

(ii) 27 mod 3 = 0 (27 =3. 9 + 0)

2.8 Vigenere cipher

Vigenere cipher adalah contoh terbaik dari cipher alfabet-majemuk 'manual'. Algoritma ini dipublikasikan oleh diplomat (sekaligus seorang

kriptologis) Perancis, Blaise de vignere pada abad 16, tahun 1586. Sebenarnya Giovan Batista Belaso telah menggambarkannya untuk pertama kali pada tahun 1955. Vigenere cipher di publikasikan pada tahun 1586, tetapi algoritma tersebut baru dikenal 200 tahun kemudian yang oleh penemunya cipher tersebut kemudian dinamakan vigenere cipher.

Pada kriptografi vigenere, plainteks akan dienkripsi dengan pergeseran huruf tetapi setiap huruf di dalam plainteks akan mengalami pergeseran yang berbeda. Kunci pada kriptografi vigenere dapat berupa angka atau huruf. Pergeseran huruf pada plainteks akan ditentukan oleh kunci yang mempunyai posisi sama dengan huruf pada plainteks. Kriptografi vigenere ini dikenal sebagai *polyalphabetic substitution* cipher, karena enkripsi terhadap satu huruf yang sama bisa menghasilkan huruf yang berbeda, sehingga lebih sukar untuk mengubah cipherteks menjadi plainteks asal jika tidak mengetahui kuncinya.

Pergeseran setiap huruf pada plainteks ditentukan oleh huruf pada posisi yang sama.

Tabel 2.1 Tabel pergeseran pada kriptografi vigenere

Kunci	A	B	C	D	E	F	G	H	I
Pergesaran *k*	0	1	2	3	4	5	6	7	8
Kunci	J	K	L	M	N	O	P	Q	R
Pergeseran *k*	9	10	11	12	13	14	15	16	17
Kunci	S	T	U	V	W	X	Y	Z	
Pergeseran *k*	18	19	20	21	22	23	24	25	

Rumus enkripsi dan deskripsi pada kriptografi vigenere dapat dinyatakan sebagai :

$$C = E(P) = (P + k) \bmod 26 \quad (2.6)$$

$$D = D(C) = (C - k) \bmod 26 \quad (2.7)$$

Dimana P adalah plaintext, C adalah ciphertext, k adalah pergeseran huruf sesuai dengan huruf pada posisi huruf pada plainteks.

2.8.1 Angka

Teknik vigenere dengan angka disebut teknik subtitusi kode geser dengan modulus 26 memberikan angka ke setiap alfabet seperti A⇔ 0, B⇔1 ... Z⇔ 25. Agar lebih jelas dapat dilihat pada tabel 2.1.

Contoh :

Plainteks : **UNISSULA**

Kunci : **KLASIK**

Untuk mendapatkan cipherteks, plainteks dan kunci diubah ke dalam bentuk angka kemudian dijumlahkan berulang sepanjang plainteks. Hasil penjumlahannya diubah kembali kebentuk huruf yang kemudian disusun sehingga menghasilkan cipherteks. Jika lebih dari 25, setelah ditambah dengan kunci maka dikurangi dengan 26.

Tabel 2.2 Contoh kriptografi vigenere dengan kunci angka

Plainteks	U	N	I	S	S	U	L	A
Pergeseran *k*	20	13	8	18	18	20	11	0
Kunci	K	L	A	S	I	K	K	L
Pergeseran *k*	10	11	0	18	8	10	10	11
Hasil	4	24	8	10	0	4	21	11
Cipherteks	E	Y	I	K	A	E	V	L

Cipherteks : **EYIKAEVL**

2.8.2 Huruf

Ide dasarnya adalah dengan menggunakan kode kaisar, tetapi jumlah pergeseran hurufnya berbeda-beda untuk setiap periode beberapa huruf tertentu. Untuk mengenkripsi pesan dengan vigenere digunekan *tabula recta* (disebut juga bujursangkar vigenere) seperti gambar dibawah ini :

Plainteks

Kunci	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	M
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 2.13 *Tabula Recta*

Tabula recta digunakan untuk memperoleh cipherteks dengan menggunakan kunci yang sudah ditentukan. Jika panjang kunci lebih pendek daripada panjang plainteks maka penggunaan kunci diulang. Secara matematis enkripsi dengan kode vignere bisa dinyatakan sebagai :

$$E(pi) = V(pi, kn \bmod m) \qquad (2.8)$$

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.*

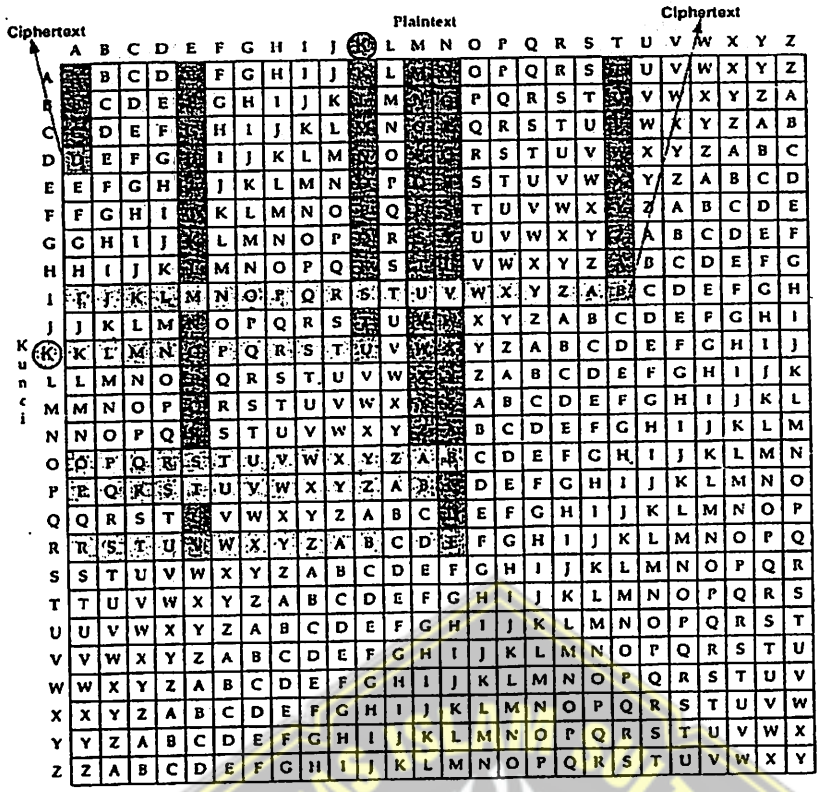
Contoh :

Plainteks : **KEAMANAN DATA MENGGUNAKAN CIPHER VIGENERE**

Kunci : **KRIPTOGRAFI**

Dengan menggunakan *tabula recta* akan didapat cipherteks sebagai berikut :

UVIBTBGE DFBK WVVVZCTRKFV FZOTGSXHE HQZYMP



Gambar 2.14 vignere Teknik tabula recta

Cara menentukan cipherteks pada sistem ini, pada tabula recta bisa dilihat bahwa posisi horizontal merupakan plainteks dan pada posisi vertikal adalah kunci. Jika plainteks huruf "K" maka dilihat posisi huruf K pada plainteks tabula recta dan posisi huruf "K" pada posisi kunci jika huruf pertama kunci juga kebetulan "K". Jika sudah menemukan, tarik garis lurus ke bawah dari plainteks dan garis lurus ke samping dari posisi kunci maka akan ditemukan huruf "U". huruf U inilah yang akan menjadi cipherteks, begitu seterusnya.

2.9. Mengenal Visual Basic (VB)

Visual Basic adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi

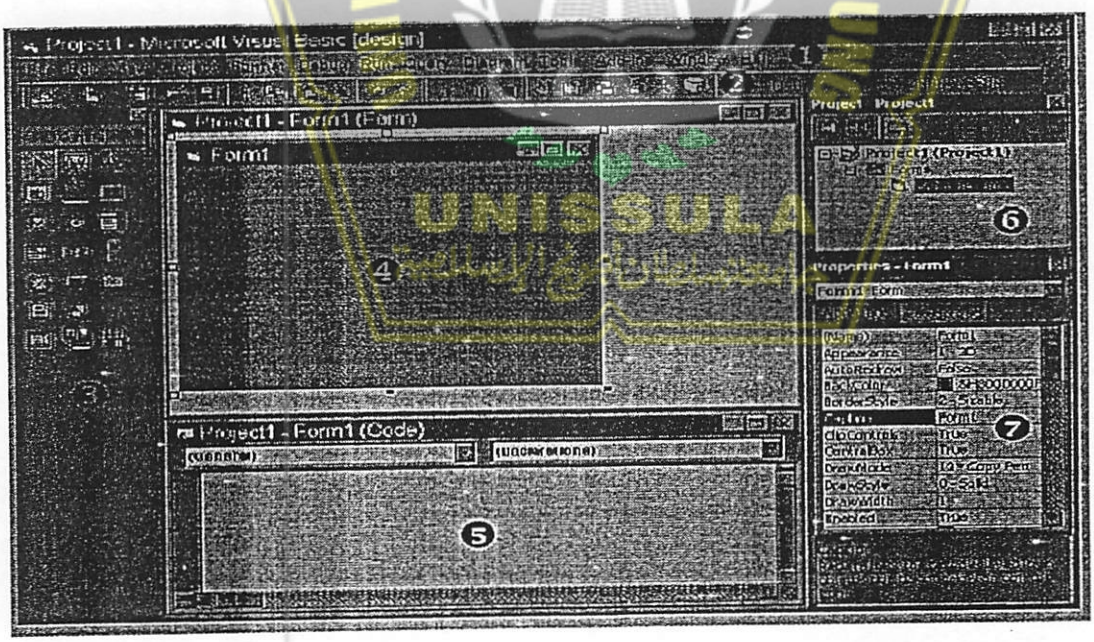
Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP).

2.9.1 Mengenal *Integrated Development Environment* (IDE) VB 6

Aktifkan VB 6 melalui tombol Start > Programs > Microsoft Visual Studio 6.0 > Microsoft Visual Basic 6.0. Tunggulah beberapa saat hingga muncul tampilan berikut :

Gambar 2.15 Tampilan utama ketika Microsoft Visual Studio 6.0 terbuka.

Pilih Standard EXE dan klik tombol Open. Anda akan melihat tampilan area kerja atau IDE VB 6. Kenali bagian-bagian utama di dalam IDE VB 6 berikut ini :



Gambar 2.14 Tampilan area kerja Microsoft Visual Studio 6.0

Keterangan :

1. Menubar
2. Toolbar
3. Toolbox

   Bila Toolbox tidak muncul klik tombol Toolbox  pada bagian Toolbar atau klik menu View > Toolbox.
4. Jendela Form

   Bila Jendela Form tidak muncul klik tombol View Object  pada bagian Project Explorer atau klik menu View > Object.
5. Jendela Code

   Bila Jendela Code tidak muncul klik tombol View Code  di pada bagian Project Explorer atau klik menu View > Code.
6. Project Explorer

   Bila Project Explorer tidak muncul klik tombol Project Explorer  pada bagian Toolbar atau klik menu View > Project Explorer.
7. Jendela Properties

   Bila Jendela Properties tidak muncul klik tombol Properties Window  pada bagian Toolbar atau klik menu View > Properties Window.

BAB III
PERANCANGAN SISTEM

Perancangan sebuah program merupakan langkah yang pertama dan utama dilakukan untuk membangun suatu sistem, hal itu adalah yang utama perlu dilakukan, dengan melakukan perancangan program yang telah diperhitungkan dengan matang, akan menghasilkan output sistem yang memiliki konstruksi yang kokoh dan baik maupun proses pengolahan data yang tepat dan akurat.

3.1 Penunjang Untuk Pembuatan Program

3.1.1 Kebutuhan Hardware

Kebutuhan hardware yang diperlukan untuk membuat program adalah seperangkat komputer dengan spesifikasi sebagai berikut :

1. Processor : AMD Duron 2200+
2. Hardisk : 40 GB
3. VGA : NVDIA Geforce MX 4000 64 MB
4. RAM : 256 MB

3.1.2 Kebutuhan Software

Untuk membuat program komputer otomatis dibutuhkan juga sebuah software pendukung baik itu software untuk sistem operasi komputer itu sendiri dan juga software yang digunakan untuk mengedit dan mendebug program, software yang digunakan untuk membuat program adalah :

1. Sistem operasi yang digunakan dengan menggunakan sistem operasi Microsoft Windows XP
2. Bahasa pemograman yang digunakan adalah Microsoft visual Basic 6.

3.2 Vigenere Cipher

Vigenere Cipher sangat dikenal karena mudah dipahami dan diimplementasikan. *Cipher* menggunakan bujursangkar *Vignere* untuk melakukan enkripsi seperti ditunjukkan pada Gambar 3.1. Kolom paling kiri dari bujursangkar menyatakan hutuf-huruf kunci, sedangkan baris paling atas menyatakan huruf-huruf plainteks. Setiap baris

di dalam bujursangkar menyatakan huruf-huruf cipherteks yang diperoleh dengan *Caesar cipher*, yang mana jumlah pergeseran huruf plainteks ditentukan nilai numerik huruf kunci tersebut (yaitu, a = 0, b = 1, c = 2, ...,z = 25). Sebagai contoh, huruf kunci c (= 2) menyatakan huruf-huruf plainteks digeser sejauh 2 huruf ke kanan (dari susunan alfabetnya),

sehingga huruf-huruf cipherteks pada baris c adalah :

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

Bujursangkar *Vigenere* digunakan untuk memperoleh cipherteks dengan menggunakan kunci yang ditentukan. Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci diulang penggunaannya (sistem periodik). Bila panjang kunci adalah *m*, maka periodenya dikatakan *m*. Sebagai contoh, jika plainteks adalah THIS PLAINTEXT dan kunci adalah sony, maka penggunaan kunci secara periodik adalah sebagai berikut:

Plainteks : THIS PLAINTEXT

Kunci : sony sonysonys

	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	M
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 3.1 Bujursangkar *Vigenere*

Setiap huruf plainteks akan dienkripsi dangan setiap huruf kunci di bawahya. Untuk mengerjakan enkripsi dengan *Vigenere Cipher*, lakukan pada bujursangkar *Vigenere* sebagai berikut : tarik garis vertikal dari huruf plainteks ke bawah, lalu tarik garis mendatar dari huruf kunci ke kanan. Perpotongan kedua garis tersebut menyatakan huruf cipherteksnya.

Contoh 3.1.

Misalkan plainteks THIS PLAINTEXT dienkripsi dengan kunci sony. Karena panjang kunci tidak sama dengan panjang plainteks, maka kunci diulang secara periodik :

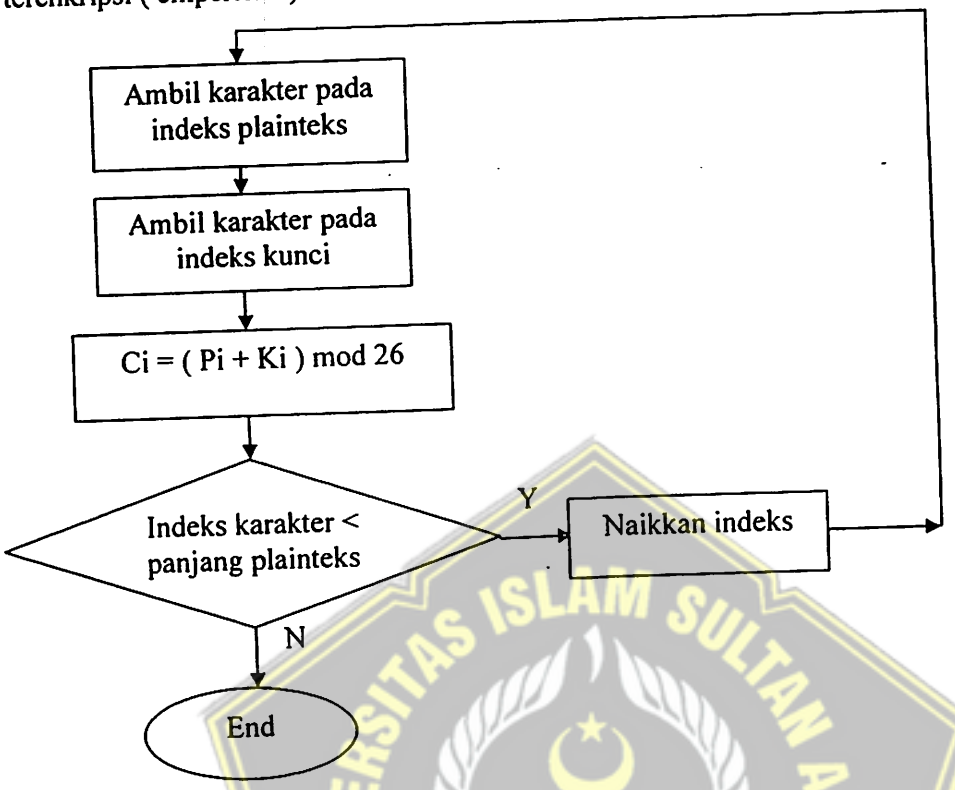
Plainteks : THIS PLAINTEXT
Kunci : sony sonysonys

Untuk huruf plainteks pertama T, tarik garis vertikal dari huruf T dan tarik garis mendatar dari huruf s, perpotongannya adalah pada kotak yang berisi huruf L. Dengan cara yang sama, tarik garis vertikal dari huruf H dan tarik garis mendatar dari huruf o, perpotongannya adalah pada kotak yang juga berisi huruf V. Hasil enkripsi seluruhnya adalah sebagai berikut :

Plainteks : THIS PLAINTEXT
Kunci : sony sonysonys
Cipherteks : **LVVQ HZNGFHRVL**

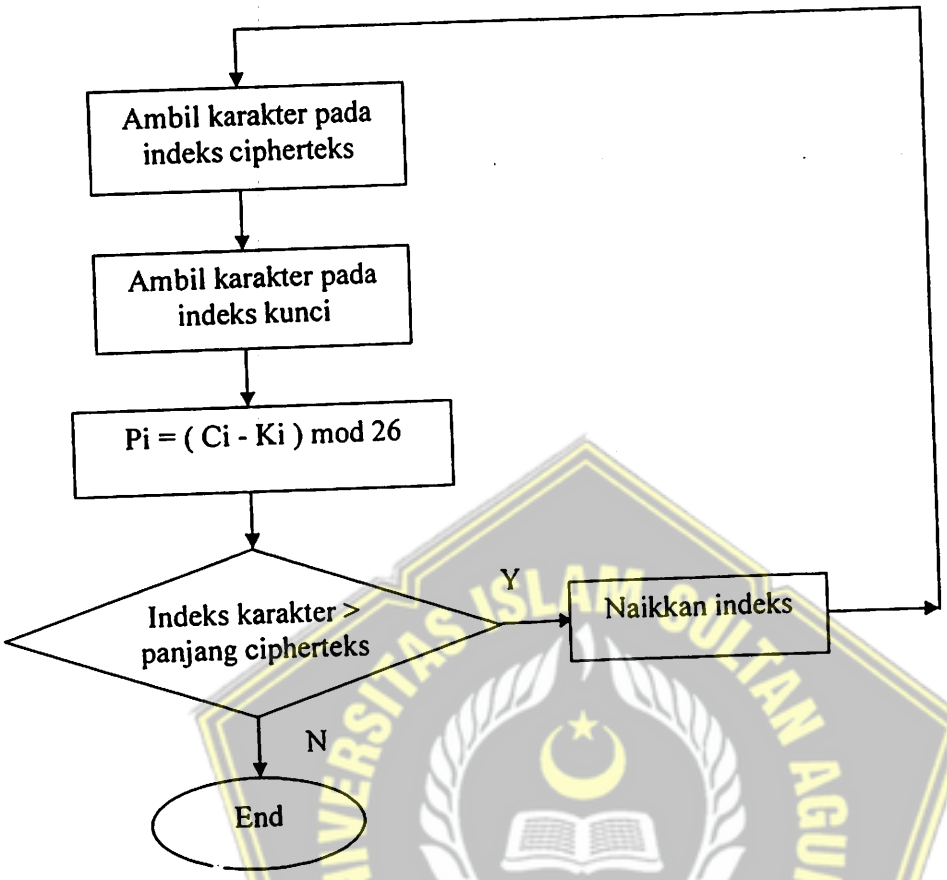
Amatilah bahwa huruf plainteks T dapat dienkripsi menjadi L atau H, dan huruf cipherteks V dapat merepresentasikan huruf plainteks H, I, dan X. Hal ini merupakan karakteristik dari *cipher* alfabet-majemuk. Pada *cipher* substitusi sederhana, setiap huruf cipherteks selalu menggantikan huruf plainteks tertentu, sedangkan pada *cipher* alfabet-majemuk setiap huruf cipherteks dapat memiliki kemungkinan banyak huruf plainteks. Jadi, dengan menggunakan *Vigenere Cipher*, kita dapat mencegah frekuensi huruf-huruf di dalam cipherteks yang mempunyai pola tertentu yang sama sebagaimana yang diperlihatkan pada *cipher* substitusi sederhana (*cipher* alfabet-tunggal). Berikut adalah

gambar diagram alir program dengan input dari plainteks dan kunci untuk mendapatkan data terenkripsi (chiperteks) :



Gambar 3.2 *Flowchart Enkripsi*

Dan gambar 3.3 adalah gambar diagram alir fungsi deskripsi



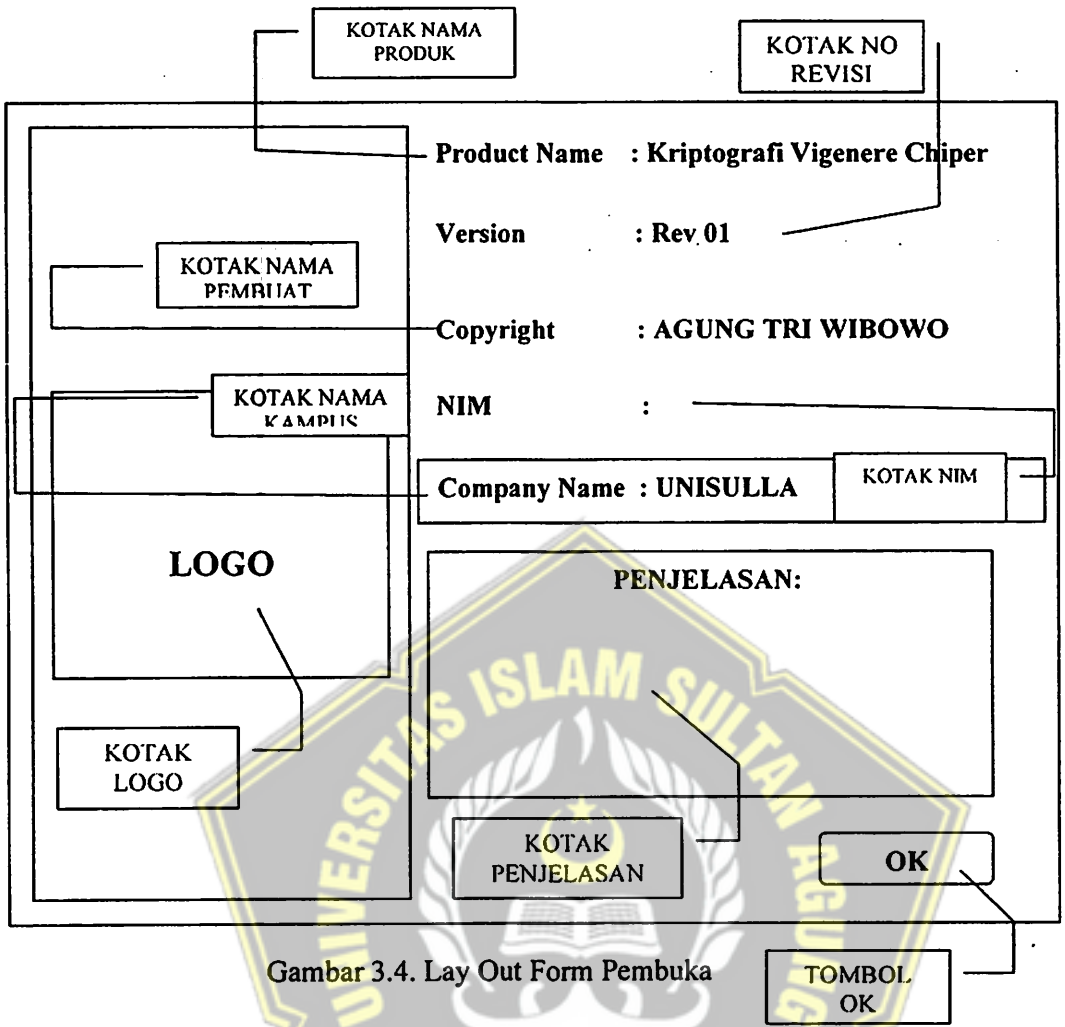
Gambar *3.3 Flowchart Deskripsi*

3.3 Desain Layout Tampilan Pada Visual Basic

Perancangan sebuah desain layout pada microsoft visual basic 2008 bertujuan untuk memperjelas display tampilan program dan juga bersifat mempermudah user dalam menjalankan program kriptografi algoritma vigenere chiper. Adapun tampilan layoutnya sebagai berikut :

3.3.1 Rancangan Form Pembuka

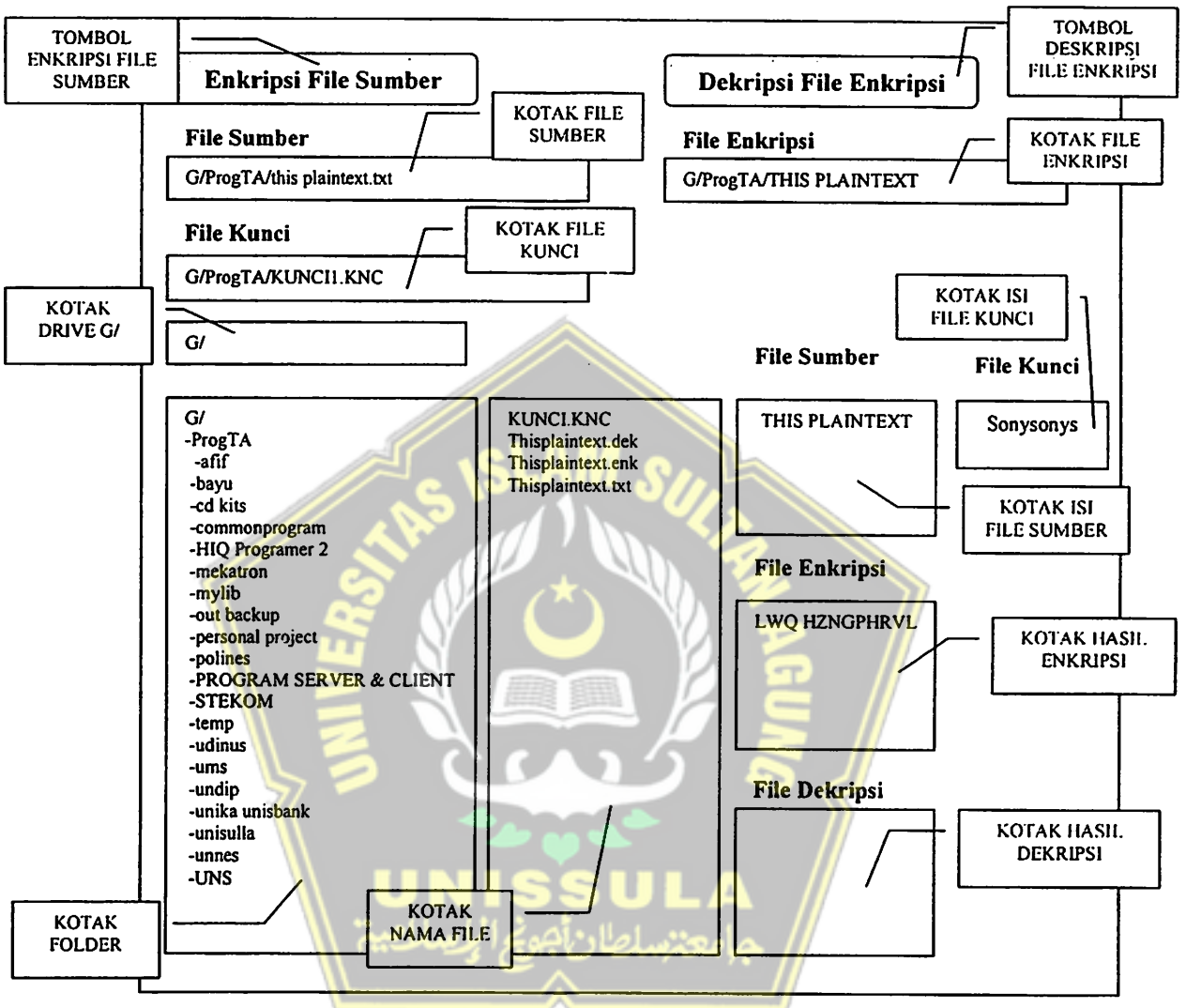
tampilan program dilakukan dengan menggunakan bahasa pemrograman microsoft visual basic 6, Tampilan awal program pada rancangan pembuka akan seperti berikut :



Gambar 3.4. Lay Out Form Pembuka

3.3.2 Rancangan Form Program Halaman Kriptografi Vigenere Chiper.

Setelah itu rancangan form program Halaman Kriptografi Vigenere chiper akan di berbentuk seperti berikut



Gambar *3.5* Form Rancangan Halaman Kriptografi Vignere

BAB IV
IMPLEMENTASI DAN SIMULASI PROGRAM

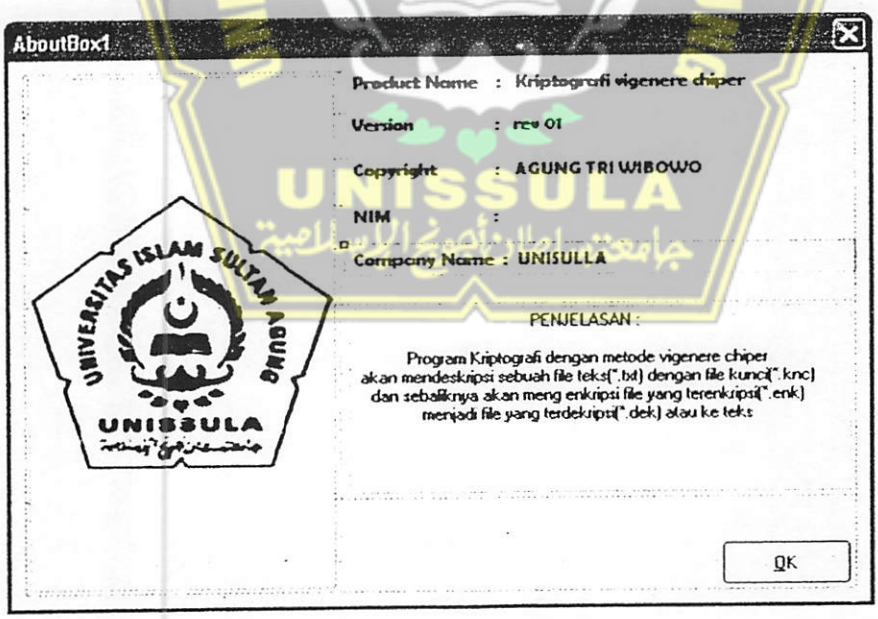
Pengujian ini dilakukan untuk mengetahui apakah program simulasi kriptografi vigenere cipher dapat berjalan dengan baik atau tidak dan kesalahan-kesalahan yang mungkin terjadi. Proses pengujian dilakukan pada enkripsi (menguji jumlah karakter dan kunci yang dapat digunakan untuk enkripsi) dan dekripsi (menguji kunci yang dapat digunakan untuk dekripsi).

4.1 Langkah-Langkah Menjalankan Program

Agar program lebih *user friendly*, dibuatlah program menjadi *.exe, sehingga bisa langsung menjalankan program dari *.exe tersebut.

4.1.1 Tampilan Awal Program

Display ataupun tampilan program dilakukan dengan menggunakan bahasa pemrograman microsoft visual studi 2008, adapun tampilan tersebut dilakukan agar bisa memberikan informasi tentang nama applikasi program yang dijalankan, nama Tampilan awal program ketika dijalankan seperti berikut :

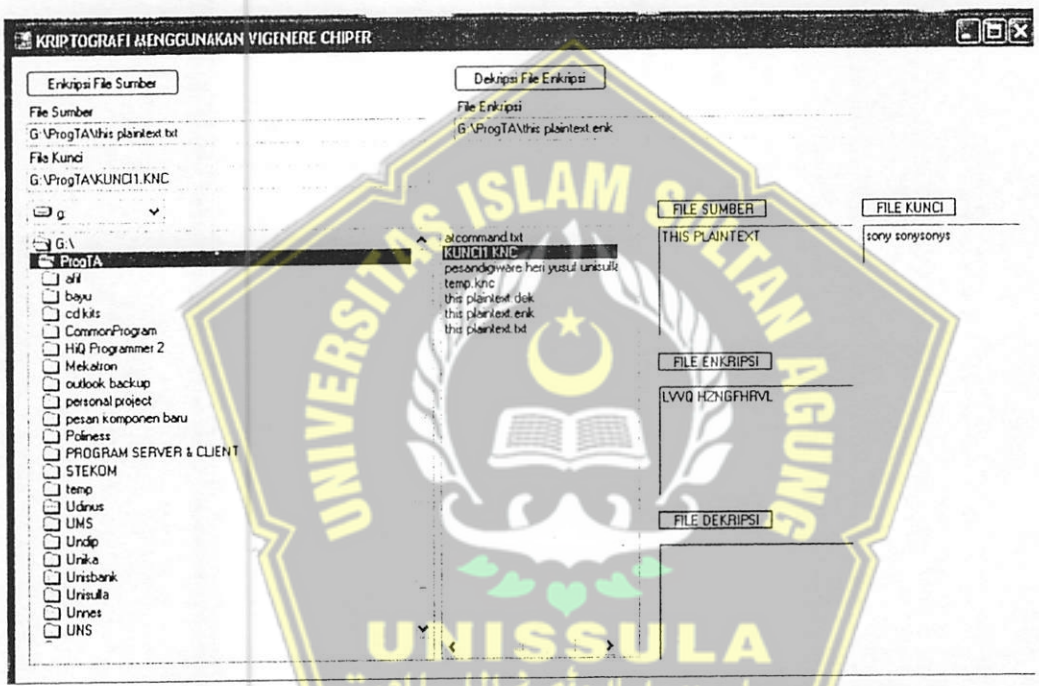


Gambar 4.1 Tampilan awal program ketika dijalankan

Tampilan awal program itu dilakukan dengan menggunakan form aboutbox yang ada pada microsoft visual studio 2008. Aboutbox dialog adalah tampilan form yang berfungsi sebagai tampilan yang memberikan informasi segala sesuatu tentang aplikasi program yang dibuat.

4.1.2 Tampilan Program Pembangkitan kunci, Enkripsi dan Deskripsi

Tampilan program pembangkitan kunci, enkripsi dan deskripsi akan muncul setelah tombol OK pada form about dialog di klik. Adapun tampilannya sebagai berikut :



Gambar 4.2 Tampilan program pembangkitan kunci, enkripsi dan deskripsi

Pada tampilan program utama enkripsi dan deskripsi terdapat beberapa tombol dan beberapa kotak input. Tombol-tombol itu adalah :

- Tombol Enkripsi File Sumber
- Tombol Dekripsi File Enkripsi

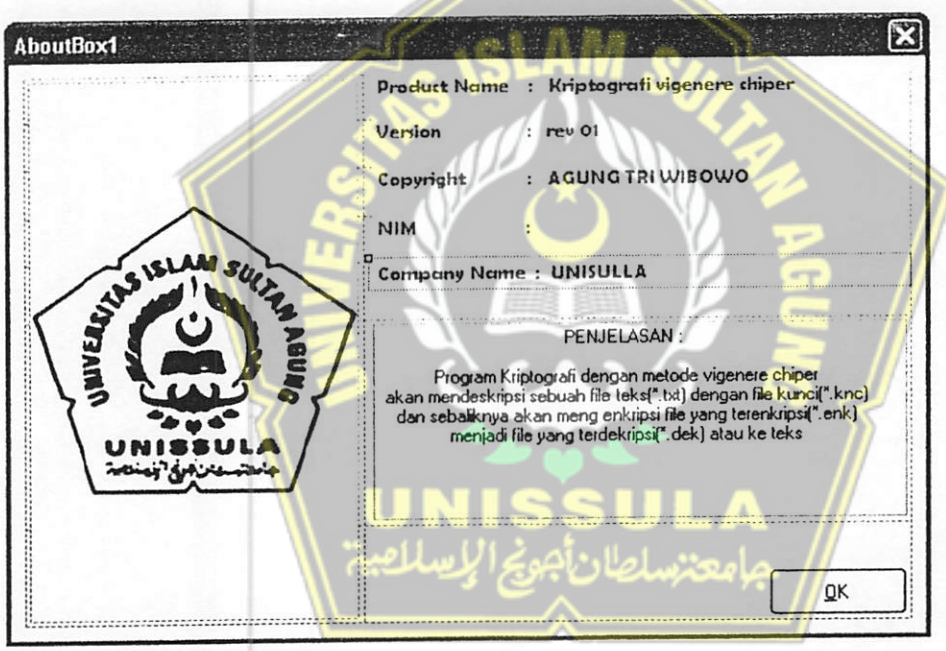
Dan beberapa kotak input yaitu

- Kotak input File Sumber
- Kotak input File Kunci
- Kotak input File Enkripsi
- Kotak input lokasi drive

Dan kotak output adalah:

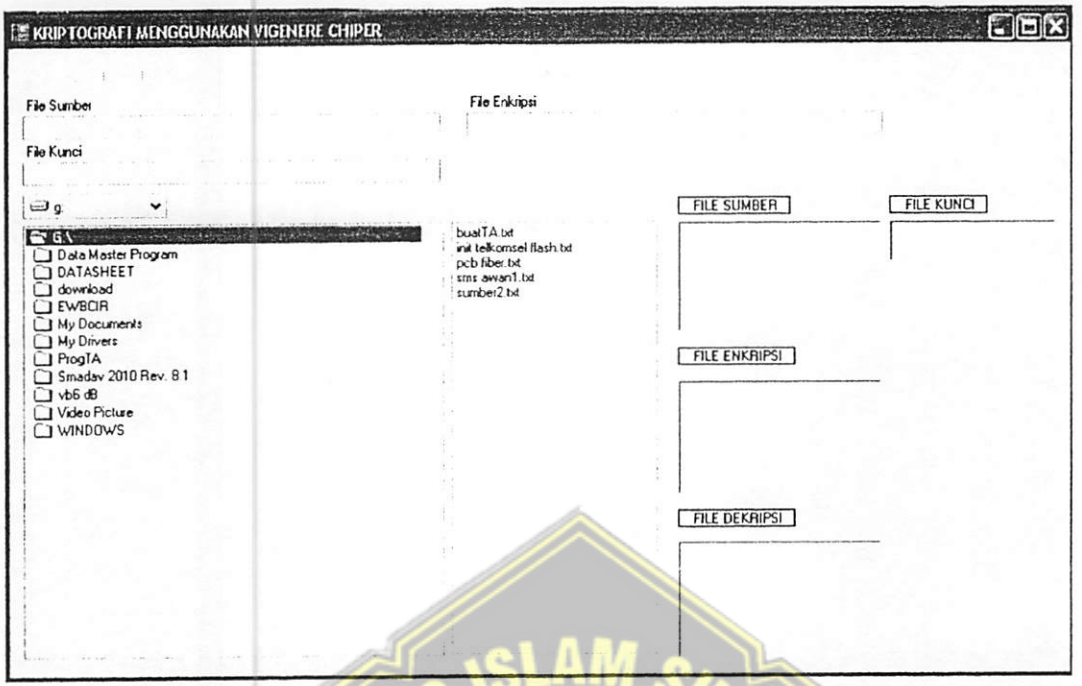
- Kotak tampilan lokasi folder
- Kotak tampilan file
- Kotak tampilan file sumber
- Kotak tampilan file kunci
- Kotak tampilan file enkripsi
- Kotak tampilan file dekripsi

Ketika pertama kali di jalankan program akan menampilkan kotak about box yang akan menampilkan informasi-informasi tentang applikasi yang ini dan nama pembuat.



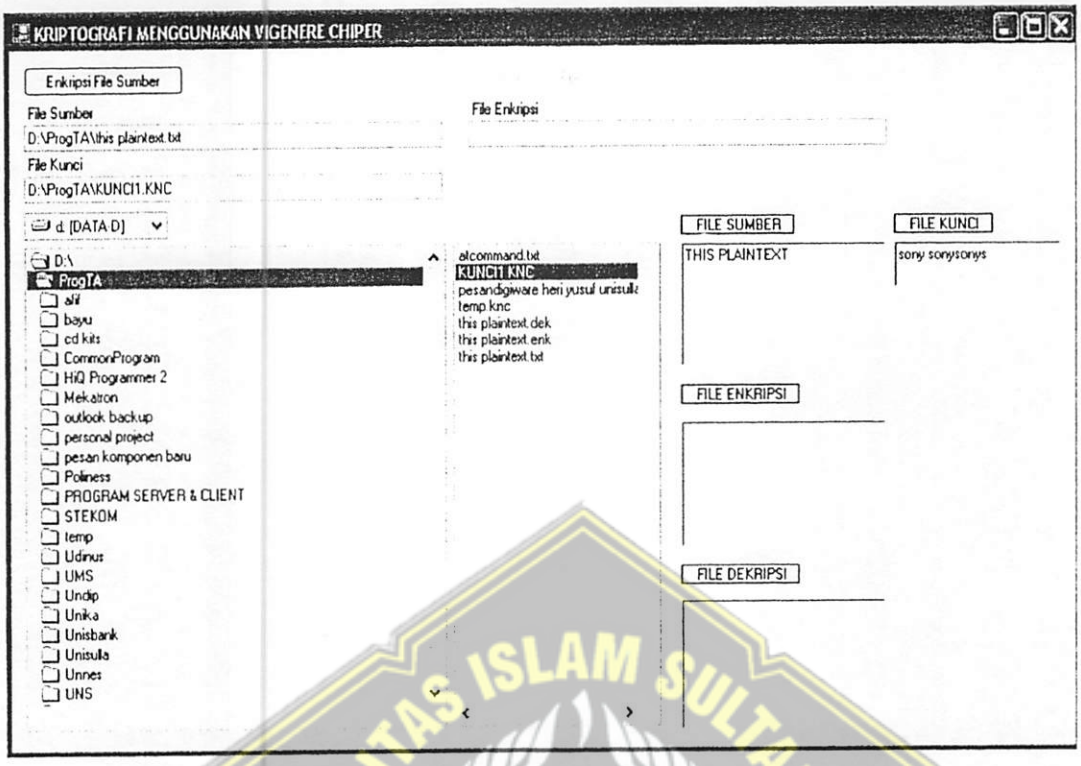
Gambar 4.3 Tampilan awal program ketika dijalankan

Setelah ditekan tombol OK pada lembar atau form about box maka program akan menampilkan lembar program utama fungsi enkripsi dan dekripsi dengan metode vigenere chipper.



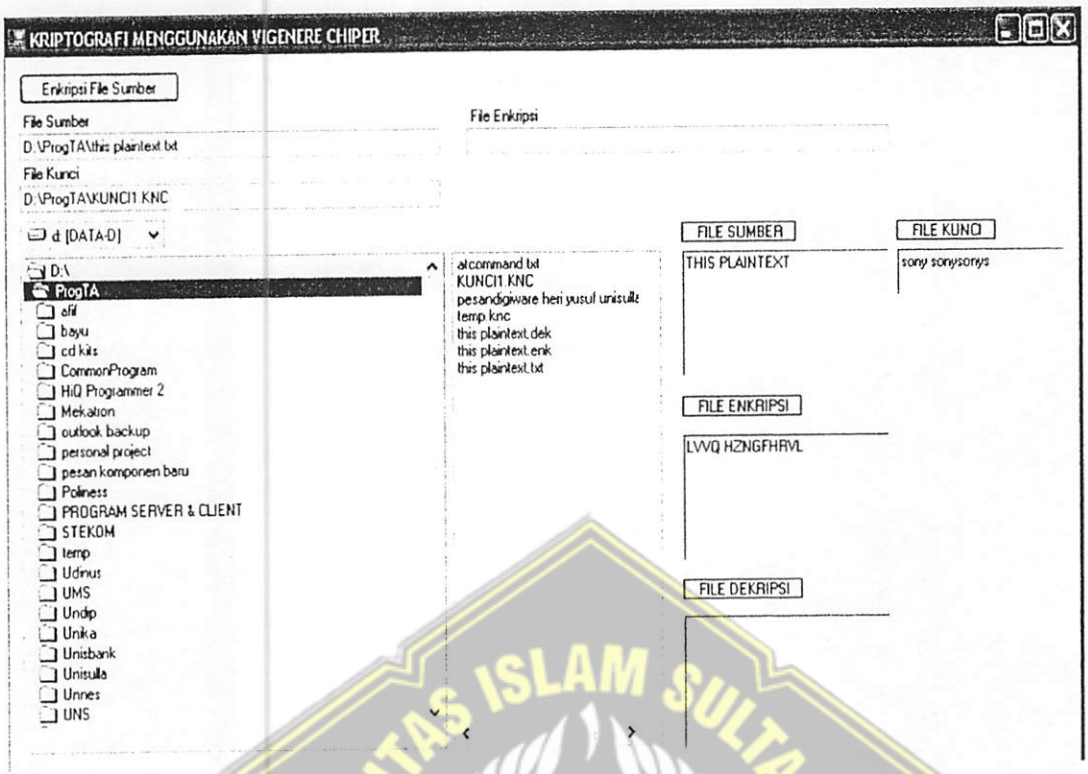
Gambar 4.4 Tampilan ketika tombol "Tampilkan kunci" diklik

Dalam gambar tampak Tombol Enkripsi File Sumber tampak tidak aktif ketika salah satu atau kedua kotak input File Sumber dan File Kunci kosong tidak ada file yang ditunjuk. Untuk mengaktifkan tombol Enkripsi File Sumber maka kotak input File SUmber dan file kunci harus berisi nama file. Untuk mengisi nama file secara otomatis maka user bisa mencari dan memilih file yang akan di enkripsi dengan mengganti drive maka kotak folder akan menampilkan semua folder pada drive dan kotak nama file akan menampilkan semua file, maka pengguna tinggal memilih file di kotak file dengan menekan dua kali tombol mouse maka file yang bersangkutan yang berextensi .txt akan masuk ke kotak File Sumber dan yang berekstensi *.knc (Kunci) akan masuk ke kotak File Kunci. Maka setelah ini tombol Enkripsi akan aktif. Tampak dalam gambar berikut



Gambar 4.5 Tampilan ketika memasukkan file sumber

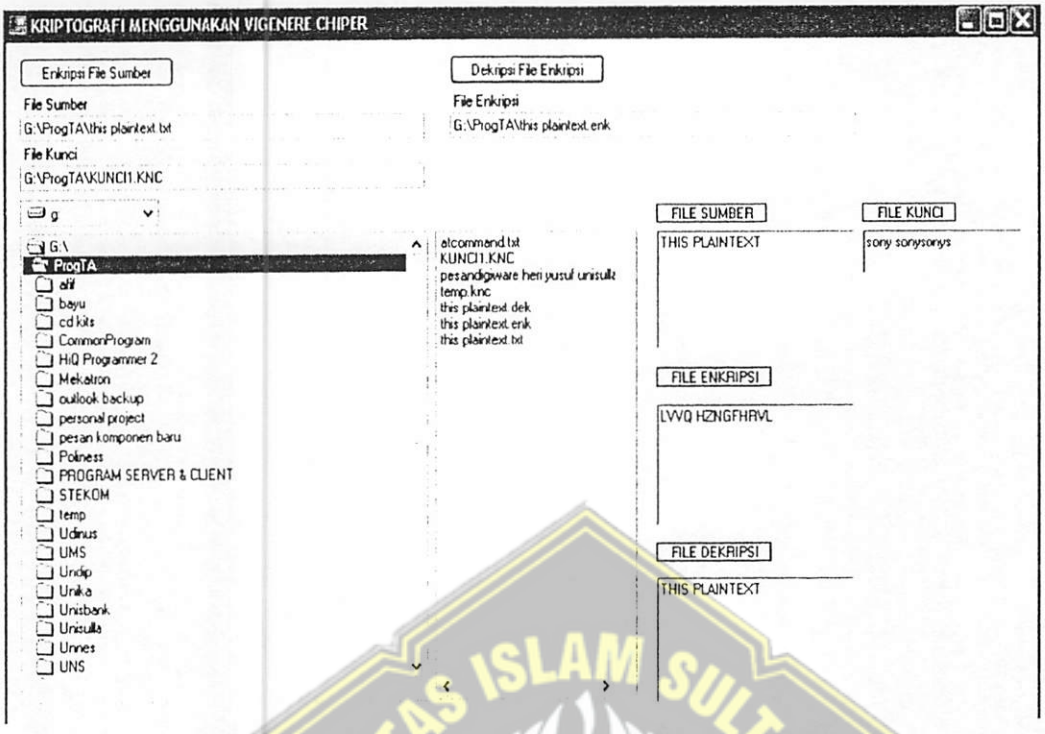
Dalam gambar diatas ketika salah satu file apakah itu file sumber atau file kunci terpilih maka isi file akan ditampilkan di kotak tampilan file sumber dan kotak tampilan file kunci. Maka setelah semua data untuk proses enkripsi sudah siap dan tombol enkripsi menjadi aktif maka ketika tombol enkripsi ditekan kemudian proses enkripsi file sumber dengan file kunci akan dikerjakan. Sebagaimana yang telah dijelaskan tentang proses enkripsi menggunakan metode vigenere chipper ini maka program visual basic akan membaca satu demi satu karakter dari file sumber yang telah ditunjuk tadi dan juga program visual basic akan membaca satu demi satu karakter pada file kunci. Kemudian hasil pembacaan karakter pada file sumber dengan hasil pembacaan karakter pada file kunci diproses dengan metode vigenere chipper sehingga didapatkan teks ter enkripsi dan hasil dari chipper teks file dekripsi akan ditampilkan ke kotak tampilan file dekripsi. Berikut adalah gambar setelah proses enkripsi



Gambar 4.6 Tampilan ketika muncul hasil enkripsi

Tampak pada gambar setelah proses enkripsi hasil teks terenkripsi di kotak File Enkripsi. Kemudian teks terenkripsi ini akan disimpan ke dalam file yang nama nya sama dengan nama File Sumber cuma berbeda ekstensi nya. File Sumber ber ekstensi .txt sedangkan File enkripsi ber ekstensi .dek.

Untuk mendekripsi kan ulang file yang telah di enkripsi maka pilih file dengan ekstensi *.enk dengan menggunakan pointer mouse dan dobel klik maka secara otomatis file yang terpilih tadi akan masuk ke kotak File Enkripsi. Dan setelah ditekan tombol Dekripsi File Enkripsi maka program akan melakukan proses dekripsi file dan hasilnya akan ditampilkan di kotak File Dekripsi seperti pada gambar berikut:



Gambar 4.7 Tampilan ketika muncul hasil Deskripsi

4.2 Pengujian Aplikasi Program

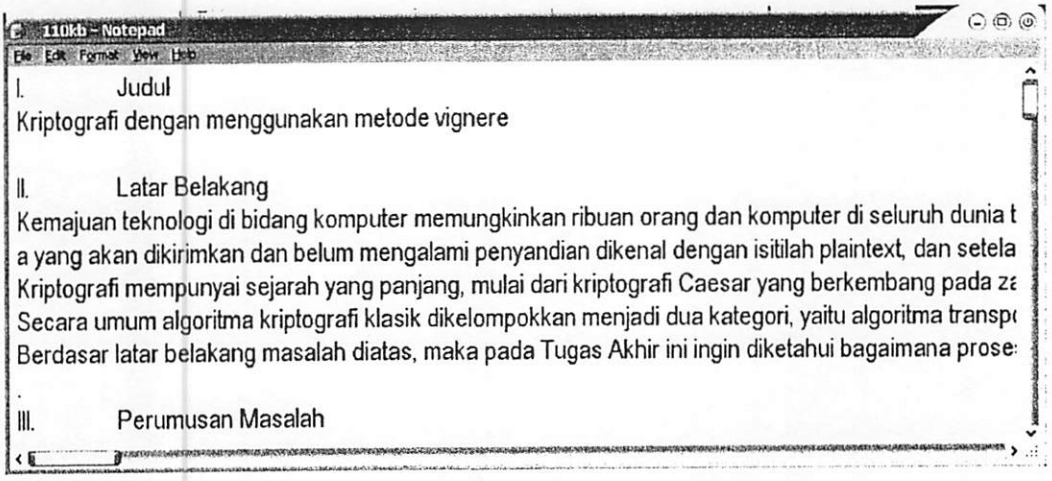
Pada pengujian aplikasi program dilakukan pada file berekstensi *.txt*, *.rtf* dan gambar.

4.2.1 Pengujian File Berekstensi *.txt*

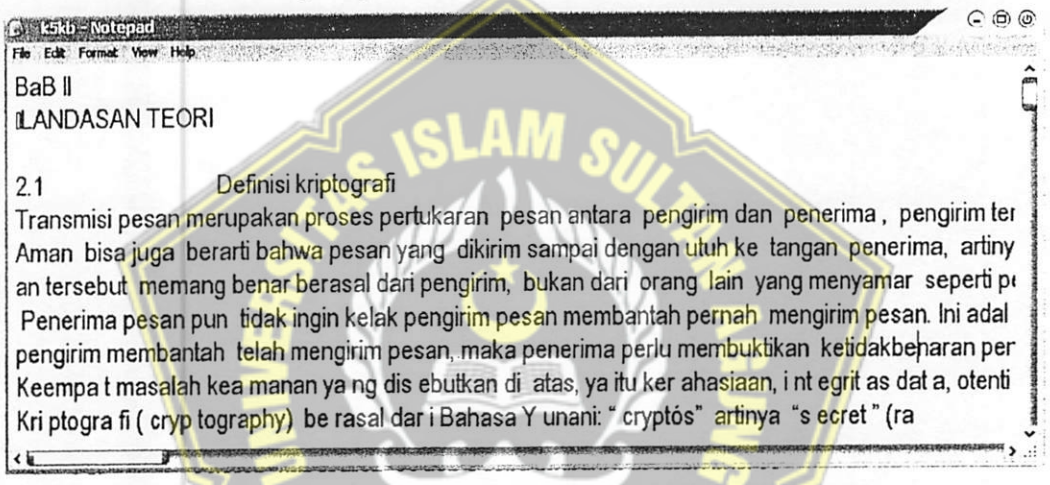
Pengujian file berekstensi *.txt* dibagi menjadi tiga yaitu pengujian ukuran file plainteks lebih besar dari ukuran file kunci, pengujian ukuran file plainteks samadengan ukuran file kunci dan pengujian ukuran file plainteks lebih kecil dari ukuran file kunci.

4.2.1.1 Pengujian Ukuran File Plainteks Lebih Besar Dari Ukuran File kunci

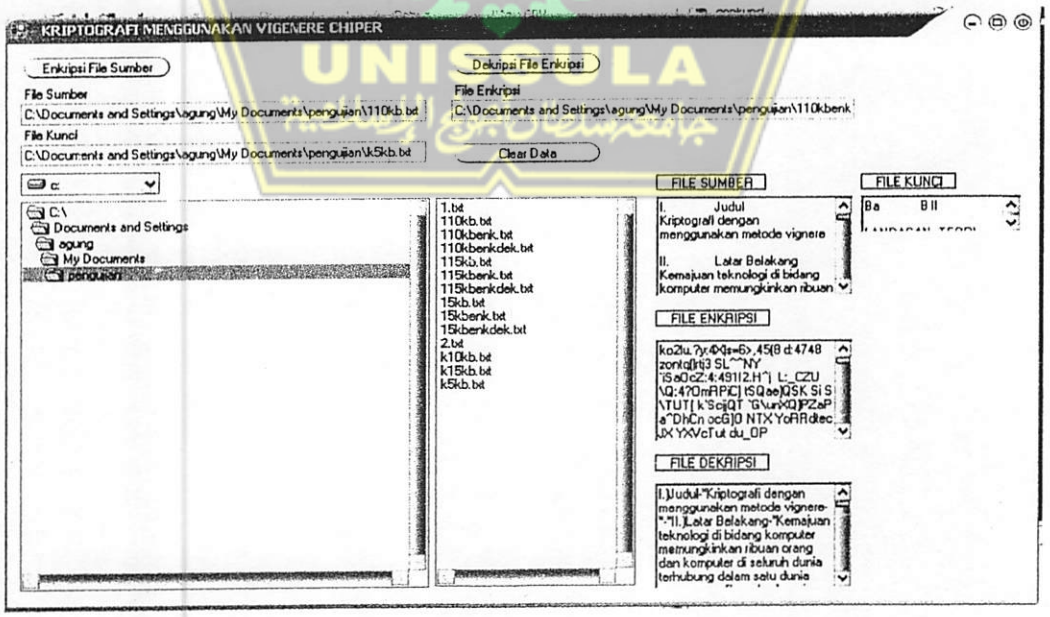
Pengujian ukuran file plainteks lebih besar dari ukuran file kunci dilakukan dengan ukuran file plainteks 10 Kb dan ukuran file kunci 5 Kb serta di dapatkan ukuran file hasil pengujian yang sama dengan ukuran file plainteks.



Gambar 4.8 Tampilan plainteks berupa file txt kapasitas 10 kb.



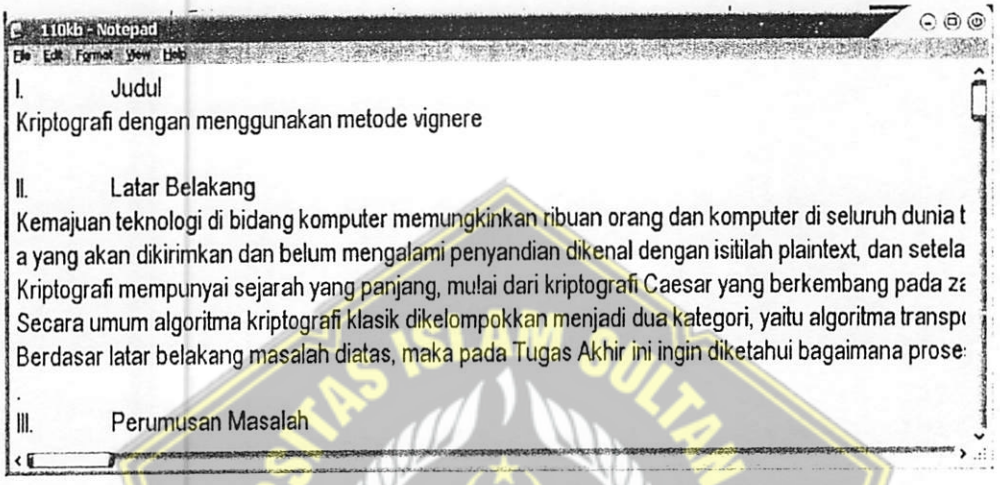
Gambar 4.9 Tampilan kunci berupa file txt dengan kapasitas 5 kb



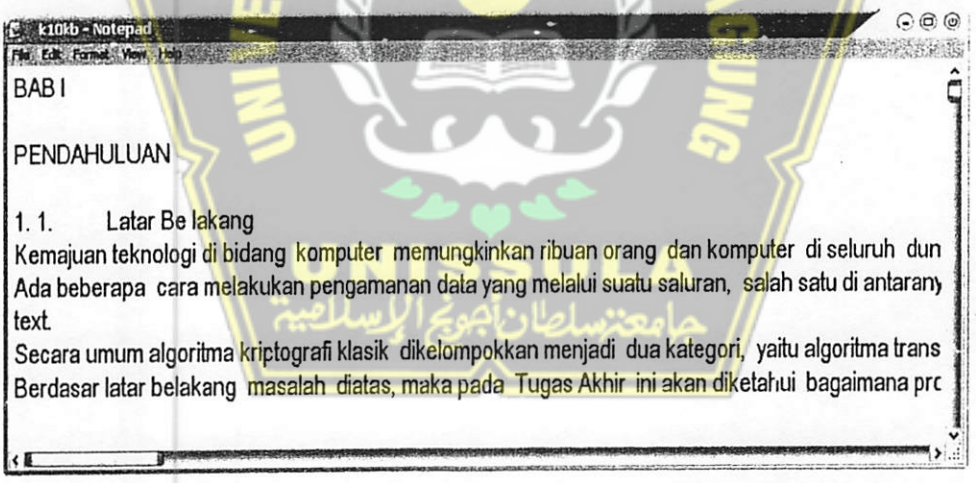
Gambar 4.10 Tampilan hasil enkripsi

4.2.1.2 Pengujian Ukuran File Plainteks Samadengan Ukuran File Kunci

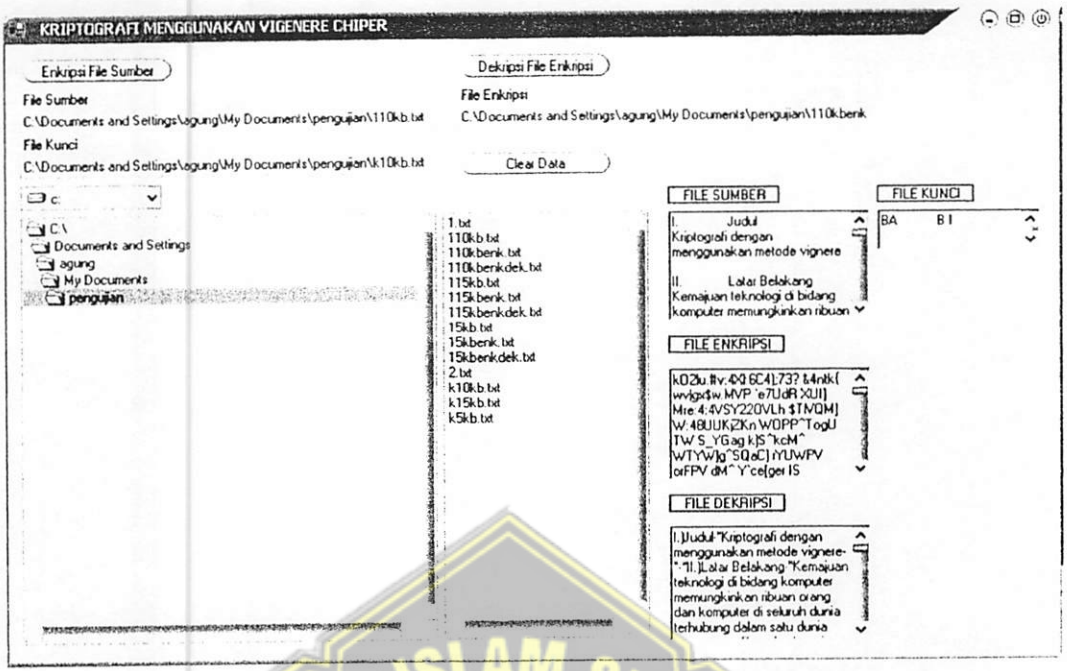
Pengujian ukuran file plainteks samadengan ukuran file kunci dilakukan dengan ukuran file plainteks 10 Kb dan ukuran file kunci 10 Kb serta di dapatkan ukuran file hasil pengujian yang sama dengan ukuran file plainteks.



Gambar 4.11 Tampilan plainteks berupa file txt kapasitas 10 kb.



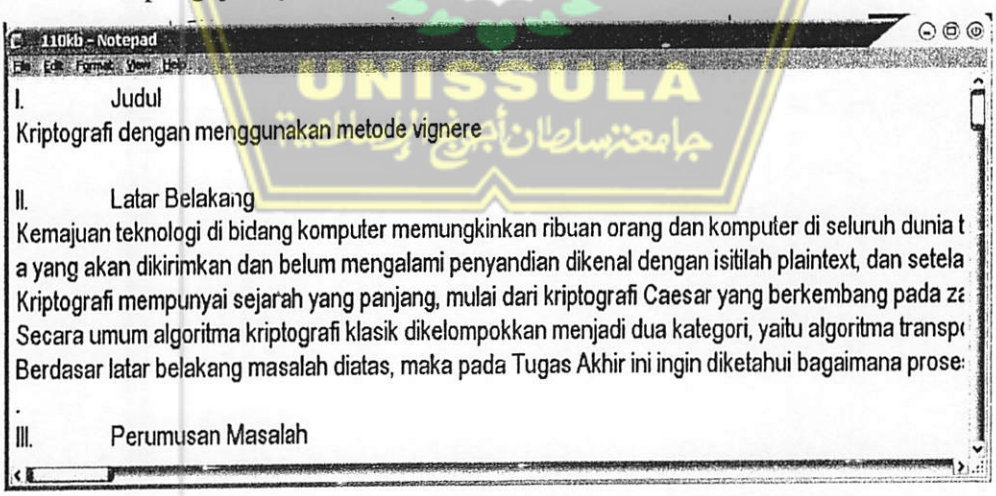
Gambar 4.12 Tampilan kunci berupa file txt kapasitas 10 kb.



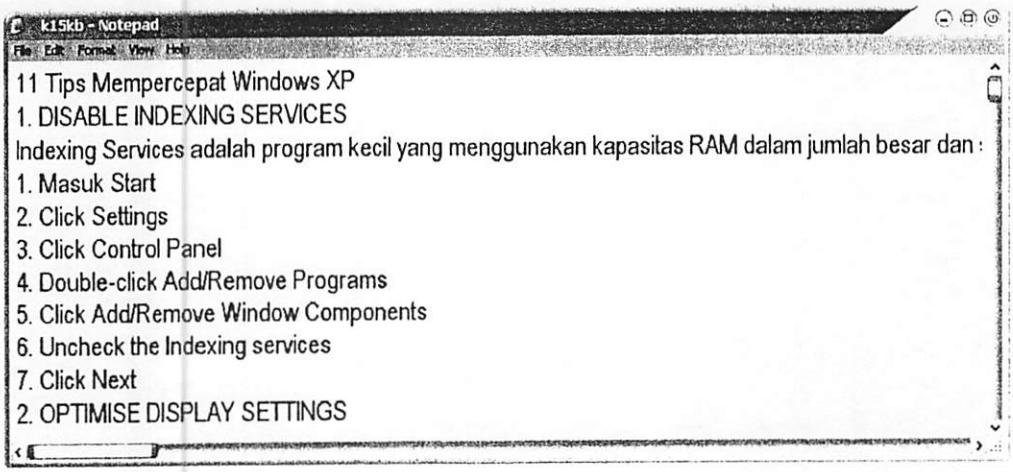
Gambar 4.13 Tampilan hasil enkripsi dan deskripsi

4.2.1.3 Pengujian Ukuran File Plainteks Lebih Kecil Dari Ukuran File Kunci

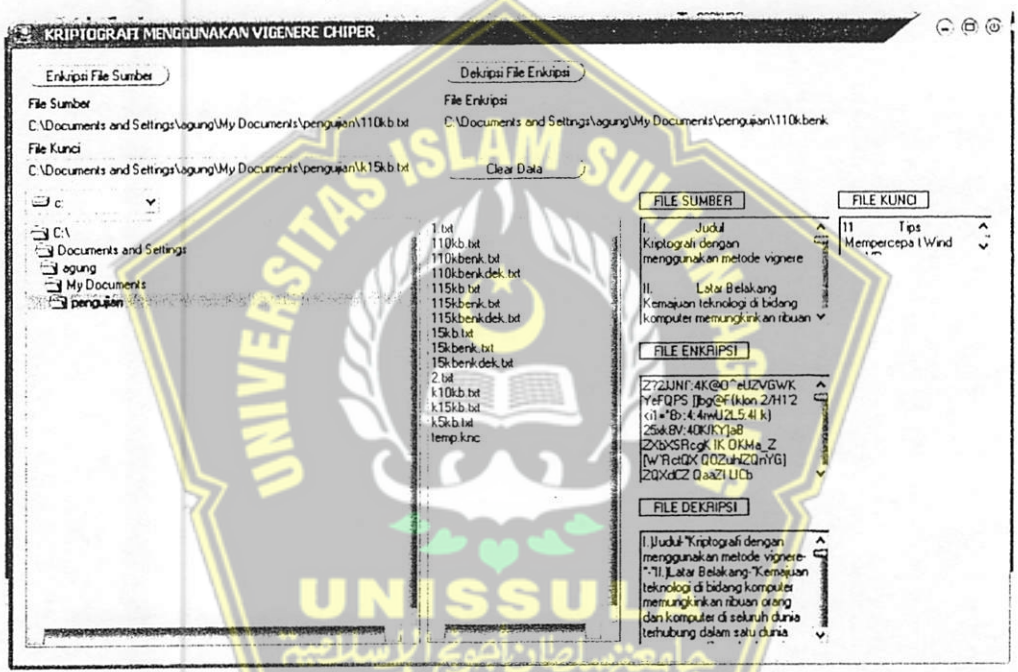
Pengujian ukuran file plainteks lebih kecil dari ukuran file kunci dilakukan dengan ukuran file plainteks 10 Kb dan ukuran file kunci 15 Kb serta di dapatkan ukuran file hasil pengujian yang sama dengan ukuran file plainteks.



Gambar 4.14 Tampilan plainteks berupa file txt kapasitas 10 kb.



Gambar 4.15 Tampilan kunci berupa file txt kapasitas 15 kb.



Gambar 4.16 Tampilan enkripsi dan deskripsi.

4.2.2 Pengujian File Berekstensi *.rtf*

Pengujian file berekstensi *.rtf* dilakukan dengan membuat file plainteks pada wordpad. Pada waktu melakukan pengujian ternyata file plainteks yang di buat dengan wordpad tidak dapat diidentifikasi oleh program sehingga tidak dapat diinputkan.

4.2.3 Pengujian File Gambar

Pengujian file gambar dilakukan dengan memasukan file berupa gambar. Pada waktu melakukan pengujian file tersebut tidak dapat diidentifikasi oleh program sehingga tidak dapat diinputkan.

4.3 Analisa

Berdasarkan pengujian yang telah dilakukan dapat dianalisa bahwa program yang di buat hanya dapat diimplementasikan pada file berekstensi *.*txt* dengan variasi ukuran kunci dan ukuran plainteks yang sama yaitu 10 Kb dan menghasilkan file yang berukuran sama dengan plainteks serta untuk file gambar dan file berekstensi *.*rtf* didapatkan hasil program berisi tulisan yang sulit dibaca.

Keamanan kriptografi vigenere dalam program tugas akhir dapat dipersentasekan. Pengujian dilakukan untuk ukuran file plainteks yang sama yaitu 10Kb dan variasi file kunci 5Kb, 10Kb dan 15Kb. Untuk pengujian file plainteks 10Kb dan file kunci 5 Kb kemungkinan kriptanalis mendapatkan key yang sama dapat dihitung dengan ${}_{10}P_5$ = 30240 kemungkinan, maka persentase pemecahan program untuk pengujian file plainteks 10Kb dan file kunci 5 Kb sebesar 0,833%, didapat dari $\frac{{}_{10}P_5}{{}_{10}P_9} \times 100\%$. Untuk pengujian file plainteks 10Kb dan file kunci 10 Kb kemungkinan kriptanalis mendapatkan key yang sama dapat dihitung dengan ${}_{10}P_{10}$ = 3628800 kemungkinan, maka persentase pemecahan program untuk pengujian file plainteks 10Kb dan file kunci 10 Kb sebesar 100% didapat dari $\frac{{}_{10}P_{10}}{{}_{10}P_9} \times 100\%$. Untuk pengujian file plainteks 10Kb dan file kunci 15 Kb kemungkinan kriptanalis mendapatkan key yang sama dapat dihitung dengan ${}_{10}P_{10}$ = 3628800 kemungkinan, maka persentase pemecahan program untuk pengujian file plainteks 10Kb dan file kunci 15 Kb sebesar 100% didapat dari $\frac{{}_{10}P_{10}}{{}_{10}P_9} \times 100\%$

.Dari analisa diatas dapat diambil kesimpulan bahwa program kriptografi vigenere dapat diproses dengan sangat baik pada file berekstensi *.*txt* dan

kriptanalisis sangat sulit mendapatkan plainteks asli untuk ukuran plainteks lebihbesar dari kunci.

BAB V
PENUTUP

5.1 Kesimpulan

Dari analisa system simulasi, dapat disimpulkan sebagai berikut :

1. Kriptografi vigenere menggunakan bantuan kunci untuk mengubah karakter didalam plainteks.
2. Program akan merubah karakter dari plainteks sesuai dengan kunci yang digunakan.
3. Program hanya dapat memproses dengan baik file berekstensi *.*txt*.
4. Kunci dekripsi harus sama dengan kunci enkripsi, untuk mendapatkan plainteks yang asli.
5. Tingkat keamanan kriptografi vigenere lebih terjamin dengan menggunakan ukuran kunci lebih kecil dari ukuran plainteks.

5.2 Saran

Berdasarkan kesimpulan diatas, maka saran yang dapat disampaikan adalah :

1. Kriptografi vigenere dapat diterapkan untuk jumlah karakter lebih banyak, sehingga didapat variasi kunci yang lebih banyak.
2. Untuk pengembangan, kunci enkripsi dan deskripsi harus berbeda, sehingga kerahasiaan pesan lebih terjamin.

DAFTAR PUSTAKA

Ariyus, Dony. 2006. *Computer Security*. Andi : Yogyakarta.

Abdia away, gunaidi. 2006. *The Shortcut of Matlab Programming.* Informatika : Bandung.

Arhami, Muhammad & Anita Desiani. 2005. *Pemrograman Matlab*. Andi : Yogyakarta.

Douglas R, Stinson. 2000. *Cryptography theory dan practice*. Boca raton : London.

Munir, Rinaldi. 2006. *Kriptografi*. Informatika : Bandung.

LAMPIRAN

Listing Program Tampilan Awal

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()
> _
Partial Class AboutBox1
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub


    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form
Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Dim resources As
System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(AboutBox1))
        Me.OKButton = New System.Windows.Forms.Button
        Me.TextBoxDescription = New System.Windows.Forms.TextBox
        Me.LogoPictureBox = New System.Windows.Forms.PictureBox
        Me.Label1 = New System.Windows.Forms.Label
        Me.TableLayoutPanel = New
System.Windows.Forms.TableLayoutPanel
        Me.LabelProductName = New System.Windows.Forms.Label
        Me.LabelVersion = New System.Windows.Forms.Label
        Me.LabelCopyright = New System.Windows.Forms.Label
        Me.LabelCompanyName = New System.Windows.Forms.Label
        CType(Me.LogoPictureBox,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.TableLayoutPanel.SuspendLayout()
        Me.SuspendLayout()
        '
        'OKButton
        '
        Me.OKButton.Anchor =
CType((System.Windows.Forms.AnchorStyles.Bottom Or
System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)
        Me.OKButton.DialogResult =
System.Windows.Forms.DialogResult.Cancel
        Me.OKButton.Location = New System.Drawing.Point(467, 334)
```

```
        Me.OKButton.Name = "OKButton"
        Me.OKButton.Size = New System.Drawing.Size(87, 31)
        Me.OKButton.TabIndex = 0
        Me.OKButton.Text = "&OK"
        '
        'TextBoxDescription
        '
        Me.TextBoxDescription.Dock =
System.Windows.Forms.DockStyle.Fill
        Me.TextBoxDescription.HideSelection = False
        Me.TextBoxDescription.Location = New
System.Drawing.Point(218, 168)
        Me.TextBoxDescription.Margin = New
System.Windows.Forms.Padding(6, 3, 3, 3)
        Me.TextBoxDescription.Multiline = True
        Me.TextBoxDescription.Name = "TextBoxDescription"
        Me.TextBoxDescription.ReadOnly = True
        Me.TextBoxDescription.ScrollBars =
System.Windows.Forms.ScrollBars.Both
        Me.TextBoxDescription.Size = New System.Drawing.Size(336,
131)
        Me.TextBoxDescription.TabIndex = 0
        Me.TextBoxDescription.TabStop = False
        Me.TextBoxDescription.Text =
resources.GetString("TextBoxDescription.Text")
        Me.TextBoxDescription.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center
        '
        'LogoPictureBox
        '
        Me.LogoPictureBox.Dock =
System.Windows.Forms.DockStyle.Fill
        Me.LogoPictureBox.Image =
CType(resources.GetObject("LogoPictureBox.Image"),
System.Drawing.Image)
        Me.LogoPictureBox.Location = New System.Drawing.Point(3,
3)
        Me.LogoPictureBox.Name = "LogoPictureBox"
        Me.TableLayoutPanel.SetRowSpan(Me.LogoPictureBox, 7)
        Me.LogoPictureBox.Size = New System.Drawing.Size(206, 362)
        Me.LogoPictureBox.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.Zoom
        Me.LogoPictureBox.TabIndex = 0
        Me.LogoPictureBox.TabStop = False
        '
        'Label1
        '
        Me.Label1.Dock = System.Windows.Forms.DockStyle.Fill
        Me.Label1.Font = New System.Drawing.Font("Berlin Sans FB
Demi", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.Label1.ForeColor = System.Drawing.Color.Red
        Me.Label1.Location = New System.Drawing.Point(218, 95)
        Me.Label1.Margin = New System.Windows.Forms.Padding(6, 0,
3, 0)
        Me.Label1.MaximumSize = New System.Drawing.Size(0, 17)
```

```
        Me.Label1.Name = "Label1"
        Me.Label1.Size = New System.Drawing.Size(336, 17)
        Me.Label1.TabIndex = 1
        Me.Label1.Text = "NIM                                :    "
        Me.Label1.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'TableLayoutPanel
        '
        Me.TableLayoutPanel.ColumnCount = 2
        Me.TableLayoutPanel.ColumnStyles.Add(New
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Per
cent, 38.08354!))
        Me.TableLayoutPanel.ColumnStyles.Add(New
System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Per
cent, 61.91646!))
        Me.TableLayoutPanel.Controls.Add(Me.Label1, 0, 3)
        Me.TableLayoutPanel.Controls.Add(Me.LogoPictureBox, 0, 0)
        Me.TableLayoutPanel.Controls.Add(Me.LabelProductName, 1,
0)
        Me.TableLayoutPanel.Controls.Add(Me.LabelVersion, 1, 1)
        Me.TableLayoutPanel.Controls.Add(Me.LabelCopyright, 1, 2)
        Me.TableLayoutPanel.Controls.Add(Me.LabelCompanyName, 1,
4)
        Me.TableLayoutPanel.Controls.Add(Me.TextBoxDescription, 1,
5)
        Me.TableLayoutPanel.Controls.Add(Me.OKButton, 0, 6)
        Me.TableLayoutPanel.Dock =
System.Windows.Forms.DockStyle.Fill
        Me.TableLayoutPanel.Location = New System.Drawing.Point(9,
9)
        Me.TableLayoutPanel.Name = "TableLayoutPanel"
        Me.TableLayoutPanel.RowCount = 7
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percen
t, 8.227848!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percen
t, 8.86076!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percen
t, 9.177216!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percen
t, 8.544304!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percen
t, 10.75949!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percen
t, 37.77174!))
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Percen
t, 17.3913!))
```

```
        Me.TableLayoutPanel.RowStyles.Add(New
System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolu
te, 20.0!))
        Me.TableLayoutPanel.Size = New System.Drawing.Size(557,
368)
        Me.TableLayoutPanel.TabIndex = 0
        '
        'LabelProductName
        '
        Me.LabelProductName.Dock =
System.Windows.Forms.DockStyle.Fill
        Me.LabelProductName.FlatStyle =
System.Windows.Forms.FlatStyle.Popup
        Me.LabelProductName.Font = New System.Drawing.Font("Berlin
Sans FB Demi", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.LabelProductName.ForeColor = System.Drawing.Color.Red
        Me.LabelProductName.Location = New
System.Drawing.Point(218, 0)
        Me.LabelProductName.Margin = New
System.Windows.Forms.Padding(6, 0, 3, 0)
        Me.LabelProductName.MaximumSize = New
System.Drawing.Size(0, 17)
        Me.LabelProductName.Name = "LabelProductName"
        Me.LabelProductName.Size = New System.Drawing.Size(336,
17)
        Me.LabelProductName.TabIndex = 0
        Me.LabelProductName.Text = "Product Name      :
Kriptografi vigenere chiper"
        '
        'LabelVersion
        '
        Me.LabelVersion.Dock = System.Windows.Forms.DockStyle.Fill
        Me.LabelVersion.Font = New System.Drawing.Font("Berlin
Sans FB Demi", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.LabelVersion.ForeColor = System.Drawing.Color.Red
        Me.LabelVersion.Location = New System.Drawing.Point(218,
30)
        Me.LabelVersion.Margin = New
System.Windows.Forms.Padding(6, 0, 3, 0)
        Me.LabelVersion.MaximumSize = New System.Drawing.Size(0,
17)
        Me.LabelVersion.Name = "LabelVersion"
        Me.LabelVersion.Size = New System.Drawing.Size(336, 17)
        Me.LabelVersion.TabIndex = 0
        Me.LabelVersion.Text = "Version                    :   rev
01"
        Me.LabelVersion.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'LabelCopyright
        '
        Me.LabelCopyright.Dock =
System.Windows.Forms.DockStyle.Fill
```

```
        Me.LabelCopyright.Font = New System.Drawing.Font("Berlin
Sans FB Demi", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.LabelCopyright.ForeColor = System.Drawing.Color.Red
        Me.LabelCopyright.Location = New System.Drawing.Point(218,
62)
        Me.LabelCopyright.Margin = New
System.Windows.Forms.Padding(6, 0, 3, 0)
        Me.LabelCopyright.MaximumSize = New System.Drawing.Size(0,
17)
        Me.LabelCopyright.Name = "LabelCopyright"
        Me.LabelCopyright.Size = New System.Drawing.Size(336, 17)
        Me.LabelCopyright.TabIndex = 0
        Me.LabelCopyright.Text = "Copyright                :    AGUNG
TRI WIBOWO"
        Me.LabelCopyright.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'LabelCompanyName
        '
        Me.LabelCompanyName.Dock =
System.Windows.Forms.DockStyle.Fill
        Me.LabelCompanyName.Font = New System.Drawing.Font("Berlin
Sans FB Demi", 9.75!, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.LabelCompanyName.ForeColor = System.Drawing.Color.Red
        Me.LabelCompanyName.Location = New
System.Drawing.Point(218, 126)
        Me.LabelCompanyName.Margin = New
System.Windows.Forms.Padding(6, 0, 3, 0)
        Me.LabelCompanyName.MaximumSize = New
System.Drawing.Size(0, 17)
        Me.LabelCompanyName.Name = "LabelCompanyName"
        Me.LabelCompanyName.Size = New System.Drawing.Size(336,
17)
        Me.LabelCompanyName.TabIndex = 0
        Me.LabelCompanyName.Text = "Company Name   :   UNISULLA"
        Me.LabelCompanyName.TextAlign =
System.Drawing.ContentAlignment.MiddleLeft
        '
        'AboutBox1
        '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!,
13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.CancelButton = Me.OKButton
        Me.ClientSize = New System.Drawing.Size(575, 386)
        Me.Controls.Add(Me.TableLayoutPanel)
        Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog
        Me.MaximizeBox = False
        Me.MinimizeBox = False
        Me.Name = "AboutBox1"
        Me.Padding = New System.Windows.Forms.Padding(9)
        Me.ShowInTaskbar = False
```

```
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterParent
        Me.Text = "AboutBox1"
        CType(Me.LogoPictureBox,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.TableLayoutPanel.ResumeLayout(False)
        Me.TableLayoutPanel.PerformLayout()
        Me.ResumeLayout(False)

    End Sub
    Friend WithEvents OKButton As System.Windows.Forms.Button
    Friend WithEvents TextBoxDescription As
System.Windows.Forms.TextBox
    Friend WithEvents LogoPictureBox As
System.Windows.Forms.PictureBox
    Friend WithEvents TableLayoutPanel As
System.Windows.Forms.TableLayoutPanel
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents LabelProductName As
System.Windows.Forms.Label
    Friend WithEvents LabelVersion As System.Windows.Forms.Label
    Friend WithEvents LabelCopyright As System.Windows.Forms.Label
    Friend WithEvents LabelCompanyName As
System.Windows.Forms.Label

End Class



    Private Sub AboutBox1_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        ' Set the title of the form.
        ' Dim ApplicationTitle As String
        'If My.Application.Info.Title <> "" Then
        'ApplicationTitle = My.Application.Info.Title
        ' Else
        ' ApplicationTitle =
System.IO.Path.GetFileNameWithoutExtension(My.Application.Info.Ass
emblyName)
        ' End If
        ' Me.Text = String.Format("About {0}", ApplicationTitle)
        ' Initialize all of the text displayed on the About Box.
        ' TODO: Customize the application's assembly information
in the "Application" pane of the project
        '    properties dialog (under the "Project" menu).
        'Me.LabelProductName.Text =
My.Application.Info.ProductName
        'Me.LabelVersion.Text = String.Format("Version {0}",
My.Application.Info.Version.ToString)
        'Me.LabelCopyright.Text = My.Application.Info.Copyright
        'Me.LabelCompanyName.Text =
My.Application.Info.CompanyName
        ' Me.TextBoxDescription.Text =
My.Application.Info.Description
    End Sub
```

```
    Private Sub OKButton_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles OKButton.Click

        Form1.Show()

    End Sub

    Private Sub LogoPictureBox_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
LogoPictureBox.Click

    End Sub
End Class
```

Listing Program Enkripsi

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BEnkripsi.Click
        Dim plaindt As Byte, kripdt As Integer, cipdt As Integer
        Dim plaindek As Byte, dttmp As Integer, idx As Long


        If Not (File.Exists(TBFSumber.Text)) Then
            MsgBox("File Sumber Tidak Ada !!",
MsgBoxStyle.Information, Nothing)
            Return
        End If

        If Not (File.Exists(TBFKunci.Text)) Then
            MsgBox("File Kunci Tidak Ada !!",
MsgBoxStyle.Information, Nothing)
            Return
        End If


        Dim FSumber1 As FileStream = New
FileStream(TBFSumber.Text, FileMode.Open)
        Dim FKunci1 As FileStream = New FileStream(TBFKunci.Text,
FileMode.Open)

        If FSumber1.Length <= 0 Or FKunci1.Length <= 0 Then
            MsgBox("File Sumber atau Kunci tidak boleh  Kosong
!!", MsgBoxStyle.Information, Nothing)
            Return
        End If

        FSumber1.Close()
        FKunci1.Close()

        genkunci()

        Dim FSumber As FileStream = New FileStream(TBFSumber.Text,
FileMode.Open)
        FSumber.Seek(0, SeekOrigin.Begin)
```

```
        Dim FKunci As FileStream = New FileStream(TBFKunci.Text,
FileMode.Open)
        FKunci.Seek(0, SeekOrigin.Begin)

        nmfile =
System.IO.Path.GetFileNameWithoutExtension(Me.TBFSumber.Text)
        path = System.IO.Path.GetDirectoryName(Me.TBFSumber.Text)

        Dim fs As FileStream

        fs = File.Create(path + ".\" + nmfile + "enk" + ".txt")
        fs.Close()
        ' Open the stream and write to it.
        fs = File.OpenWrite(path + ".\" + nmfile + "enk" + ".txt")


        For idx = 0 To FSumber.Length - 1
            dtfilebyte = FSumber.ReadByte   'baca /byte file
sumber
            If dtfilebyte <= 126 Then       'ubah byte ascii ke
urutan kode ascii
                If dtfilebyte >= 32 Then
                    plaindt = dtfilebyte - 32 'dikurangi 32
                Else
                    plaindt = dtfilebyte
                End If
            Else
                plaindt = dtfilebyte - 134
            End If                          'output di var plaindt

            If FKunci.Position >= FKunci.Length - 1 Then
                FKunci.Seek(0, SeekOrigin.Begin)
            End If                  'jika sudah akhir file kunci
ulangi dari awal lagi

            dtkuncibyte = FKunci.ReadByte
            If dtkuncibyte <= 126 Then
                If dtkuncibyte >= 32 Then
                    kripdt = dtkuncibyte - 32
                Else
                    kripdt = dtkuncibyte
                End If
            Else
                kripdt = dtkuncibyte - 134
            End If                          'output di var kripdt

            If plaindt > 94 Then            'ubah plain data ke ciper
 data dng menggunakan kunci kripto data
                cipdt = plaindt
            Else
                cipdt = (plaindt + kripdt) Mod 95
                cipdt = cipdt + 32          'ouput ciper data di var
 cipdt
            End If

            'dtvigchip(i) = CByte(cipdt)
```

```
            fs.WriteByte(CByte(cipdt))

            cipdt = cipdt - 32      'ubah lagi dari ciper data ke
plain karakter asli
            If cipdt >= 0 Then
                dttmp = (cipdt - kripdt) - (95 * Int((cipdt -
kripdt) / 95))
                dttmp = dttmp + 32
                plaindek = dttmp    'hasil output di plaindek
            Else
                plaindek = 32
            End If

            'Me.ListBox2.Items.Add(Chr(dtfilebyte(i)) + "-" +
Chr(dtkuncibyte(i)) + "-" + Chr(dtvigchip(i)) + "----" +
Chr(plaindek(i)))
        Next

        fs.Close()
        FSumber.Close()
        FKunci.Close()

        Me.FileListBox1.Refresh()

        Me.RBFEnk.LoadFile(path + ".\" + nmfile + "enk" + ".txt",
RichTextBoxStreamType.PlainText)

    End Sub

    Private Sub DirListBox1_Change(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DirListBox1.Change

    End Sub

    Private Sub DirListBox1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DirListBox1.Click

    End Sub

    Private Sub DirListBox1_DoubleClick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DirListBox1.DoubleClick
        FileListBox1.Path() = DirListBox1.Path
```

Listing Program Deskripsi

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BDekiprsi.Click
        Dim keydt As Integer, cipdt As Integer, nmfile As String,
path As String
        Dim plaindek As Byte, dttmp As Integer
```

```
        If Not (File.Exists(TBFEnk.Text)) Then
            MsgBox("File Enkripsi Tidak Ada !!",
MsgBoxStyle.Information, Nothing)
            Return
        End If

        If Not (File.Exists(TBFKunci.Text)) Then
            MsgBox("File Kunci Tidak Ada !!",
MsgBoxStyle.Information, Nothing)
            Return
        End If


        Dim FSumber1 As FileStream = New FileStream(TBFEnk.Text,
FileMode.Open)
        Dim FKunci1 As FileStream = New FileStream(TBFKunci.Text,
FileMode.Open)

        If FSumber1.Length <= 0 Or FKunci1.Length <= 0 Then
            MsgBox("File Sumber atau Kunci tidak boleh  Kosong
!!", MsgBoxStyle.Information, Nothing)
            Return
        End If

        FSumber1.Close()
        FKunci1.Close()

        Dim FEnkripsi As FileStream = New FileStream(TBFEnk.Text,
FileMode.Open)
        FEnkripsi.Seek(0, SeekOrigin.Begin)

        nmfile =
System.IO.Path.GetFileNameWithoutExtension(Me.TBFEnk.Text)
        path = System.IO.Path.GetDirectoryName(Me.TBFEnk.Text)

        Dim FKunci As FileStream = New FileStream(TBFKunci.Text,
FileMode.Open)
        FKunci.Seek(0, SeekOrigin.Begin)

        Dim fs As FileStream

        ' Create the file if it exists.
        If File.Exists(path + ".\" + nmfile + "dek" + ".txt") =
False Then
            ' Create the file.
            fs = File.Create(path + ".\" + nmfile + "dek" +
".txt")
            fs.Close()
        End If

        ' Open the stream and write to it.
        fs = File.OpenWrite(path + ".\" + nmfile + "dek" + ".txt")


        For i = 0 To FEnkripsi.Length - 1
            dtfilebyte = FEnkripsi.ReadByte
```

```
            If dtfilebyte <= 126 Then
                If dtfilebyte >= 32 Then
                    cipdt = dtfilebyte - 32
                Else
                    cipdt = dtfilebyte
                End If
            Else
                cipdt = dtfilebyte - 134
            End If

            If FKunci.Position >= FKunci.Length - 1 Then
                FKunci.Seek(0, SeekOrigin.Begin)
            End If
            dtkuncibyte = FKunci.ReadByte

            If dtkuncibyte <= 126 Then
                If dtkuncibyte >= 32 Then
                    keydt = dtkuncibyte - 32
                Else
                    keydt = dtkuncibyte
                End If
            Else
                keydt = dtkuncibyte - 134
            End If

            'cipdt = cipdt - 65
            If cipdt >= 0 And cipdt <= 94 Then
                dttmp = (cipdt - keydt) - (95 * Int((cipdt -
keydt) / 95))
                dttmp = dttmp + 32
                plaindek = dttmp
            Else
                plaindek = 32
            End If

            fs.WriteByte(plaindek)

            'Me.ListBox2.Items.Add(Chr(dtfilebyte(i)) + "-" +
Chr(dtkuncibyte(i)) + "-" + Chr(dtvigchip(i)) + "----" +
Chr(plaindek(i)))
        Next
        fs.Close()
        'My.Computer.FileSystem.WriteAllBytes(path + ".\" + nmfile
+ ".dek", plaindek, False)
        Me.RBFDek.LoadFile(path + ".\" + nmfile + "dek" + ".txt",
RichTextBoxStreamType.PlainText)

        Me.FileListBox1.Refresh()

    End Sub
```

Listing Program Kunci

```
Private Sub genkunci()
        Dim dir As String

        Dim FSumber As FileStream = New FileStream(TBFSumber.Text,
FileMode.Open)
        Dim FKunci As FileStream = New FileStream(TBFKunci.Text,
FileMode.Open)

        nmfile =
System.IO.Path.GetFileNameWithoutExtension(TBFKunci.Text)
        nmfile = "temp.knc"
        dir = System.IO.Path.GetDirectoryName(TBFKunci.Text)
        nmfile = dir + "\" + nmfile
        Dim FKunciGen As FileStream = New FileStream(nmfile,
FileMode.Create)


        pjgfsumber = FSumber.Length
        pjgfkunci = FKunci.Length
        FKunciGen.Seek(0, SeekOrigin.Begin)
        FKunci.Seek(0, SeekOrigin.Begin)
        FSumber.Seek(0, SeekOrigin.Begin)

        If pjgfkunci <> pjgfsumber Then

            If pjgfkunci > 0 Then
                For i = 0 To pjgfsumber - 1
                    dtfilebyte = FSumber.ReadByte

                    If FKunci.Position >= pjgfkunci Then
                        FKunci.Seek(0, SeekOrigin.Begin)
                    End If

                    If dtfilebyte >= 33 And dtfilebyte <= 126 Then
                        dtkuncibyte = FKunci.ReadByte
                        FKunciGen.WriteByte(dtkuncibyte)
                    Else
                        FKunciGen.WriteByte(dtfilebyte)
                    End If
                Next

                FKunciGen.Flush()
                FKunciGen.Close()
                FSumber.Close()
                FKunci.Close()

                System.IO.File.Replace(nmfile, TBFKunci.Text, dir
 + "\kunci.bak")
                RbFKunci.LoadFile(TBFKunci.Text,
 RichTextBoxStreamType.PlainText)
            Else
                'MessageBox.Show("Isi File KUNCI tidak Boleh
 Kosong", "Perhatian", , MessageBoxButtons.OK,
 MessageBoxIcon.Error, MessageBoxDefaultButton.Button1,
 MessageBoxOptions.ServiceNotification, False)
```

```
            End If
        Else
            FKunciGen.Close()
            FSumber.Close()
            FKunci.Close()

            RbFKunci.LoadFile(TBFKunci.Text,
RichTextBoxStreamType.PlainText)
        End If


    End Sub
```

Listing Program pembangkitan enkripsi dan deskripsi

```
Imports System
Imports System.IO
Imports System.Text

Public Class Form1


    Dim filerdr As String
    Dim dtfilebyte As Byte, dtkuncibyte As Byte
    Dim nmfile As String, path As String
    Dim pjgfsumber As Long, pjgfkunci As Long
    Dim focidx As Byte

    Public Sub New()

        ' This call is required by the Windows Form Designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent()
call.

    End Sub

    Protected Overrides Sub Finalize()
        MyBase.Finalize()
    End Sub

    Private Sub DirListBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DirListBox1.SelectedIndexChanged

    End Sub

    Private Sub DriveListBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DriveListBox1.SelectedIndexChanged
        Me.DirListBox1.Path() = Me.DriveListBox1.Drive
```

```
    End Sub

    Private Sub FileListBox1_MouseDoubleClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles FileListBox1.MouseDoubleClick
        Dim tmpext As String

        tmpext = System.IO.Path.GetExtension(Me.DirListBox1.Path()
+ "\" + Me.FileListBox1.FileName)
        tmpext = UCase(tmpext)

        If tmpext = ".TXT" Or tmpext = ".JPG" Or tmpext = ".RTF"
Then
            If focidx = 1 Then
                Me.TBFSumber.Text = Me.DirListBox1.Path() + "\" +
Me.FileListBox1.FileName()
                RbFSumber.LoadFile(Me.TBFSumber.Text,
RichTextBoxStreamType.PlainText)
                BEnkripsi.Enabled = False
                If TBFKunci.Text <> "" Then
                    BEnkripsi.Enabled = True
                End If
                focidx = 0
            End If

            If focidx = 2 Then
                Me.TBFKunci.Text = Me.DirListBox1.Path() + "\" +
Me.FileListBox1.FileName()
                Me.RbFKunci.LoadFile(Me.TBFKunci.Text,
RichTextBoxStreamType.PlainText)
                BEnkripsi.Enabled = False
                If TBFSumber.Text <> "" Then
                    BEnkripsi.Enabled = True
                End If
                focidx = 0
            End If

            If focidx = 3 Then
                Me.TBFEnk.Text = Me.DirListBox1.Path() + "\" +
Me.FileListBox1.FileName()
                Me.RBFEnk.LoadFile(Me.TBFEnk.Text,
RichTextBoxStreamType.PlainText)
                If TBFEnk.Text <> "" Then
                    BDekiprsi.Enabled = True
                End If
                focidx = 0
            End If
        End If
    End Sub

Private Sub DirListBox1_Change(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DirListBox1.Change

    End Sub
```

```
    Private Sub DirListBox1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DirListBox1.Click

    End Sub

    Private Sub DirListBox1_DoubleClick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DirListBox1.DoubleClick
        FileListBox1.Path() = DirListBox1.Path
    End Sub

    Private Sub FileListBox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
FileListBox1.SelectedIndexChanged

    End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub

    Private Sub Form1_Shown(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Shown
        AboutBox1.Hide()

    End Sub

    Private Sub Form1_FormClosing(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles
MyBase.FormClosing
        AboutBox1.Close()


    End Sub

    Private Sub TBFSumber_MouseClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.MouseEventArgs)
Handles TBFSumber.MouseClick
        focidx = 1
    End Sub

    Private Sub TBFKunci_MouseClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
TBFKunci.MouseClick
        focidx = 2
    End Sub

    Private Sub TBFEnk_MouseClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
TBFEnk.MouseClick
        focidx = 3
    End Sub

    Private Sub Button1_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
```

```
        RbFSumber.Clear()
        RBFDek.Clear()
        RBFEnk.Clear()
        RbFKunci.Clear()

    End Sub
    Private Sub Plaintochipér()
        Dim plaindt As Byte, kripdt As Integer, cipdt As Integer
        Dim plaindek As Byte, dttmp As Integer, idx As Long

        genkunciman()
        RBFManEnk.Clear()

        For idx = 0 To RbFSumber.TextLength - 1
            dtfilebyte = Asc(RbFSumber.Text(idx))     'baca /byte
file sumber
            If dtfilebyte <= 126 Then            'ubah byte ascii
ke urutan kode ascii
                If dtfilebyte >= 32 Then
                    plaindt = dtfilebyte - 32 'dikurangi 32
                Else
                    plaindt = dtfilebyte
                End If
            Else
                plaindt = dtfilebyte - 134
            End If                          'output di var plaindt

            dtkuncibyte = Asc(RbFKunci.Text(idx))
            If dtkuncibyte <= 126 Then
                If dtkuncibyte >= 32 Then
                    kripdt = dtkuncibyte - 32
                Else
                    kripdt = dtkuncibyte
                End If
            Else
                kripdt = dtkuncibyte - 134
            End If                          'output di var kripdt

            If plaindt > 94 Then            'ubah plain data ke ciper
data dng menggunakan kunci kripto data
                cipdt = plaindt
            Else
                cipdt = (plaindt + kripdt) Mod 95
                cipdt = cipdt + 32       'ouput ciper data di var
cipdt
            End If

            RBFManEnk.AppendText(Chr(cipdt))

            'fs.WriteByte(CByte(cipdt))

            cipdt = cipdt - 32       'ubah lagi dari ciper data ke
plain karakter asli
            If cipdt >= 0 Then
```

```
                dttmp = (cipdt - kripdt) - (95 * Int((cipdt -
kripdt) / 95))
                dttmp = dttmp + 32
                plaindek = dttmp     'hasil output di plaindek
            Else
                plaindek = 32
            End If

        Next idx
    End Sub

    Private Sub BManEnkripsi_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles BManEnkripsi.Click
        Plaintochiper()
    End Sub

    Private Sub genkunciman()
        Dim dtkuncigen As String = "", idxkey As Integer = 0

        If RbFKunci.TextLength > 0 And RbFSumber.TextLength > 0
Then

            For i = 0 To RbFSumber.TextLength - 1

                dtfilebyte = Asc(RbFSumber.Text(i))


                If dtfilebyte >= 33 And dtfilebyte <= 126 Then
                    dtkuncigen = dtkuncigen +
RbFKunci.Text(idxkey)

                    If idxkey >= RbFKunci.TextLength - 1 Then
                        idxkey = -1
                    End If

                    idxkey = idxkey + 1
                Else
                    dtkuncigen = dtkuncigen + RbFSumber.Text(i)
                End If


            Next

            RbFKunci.Clear()
            RbFKunci.AppendText(dtkuncigen)


        Else
            'MessageBox.Show("Isi File KUNCI tidak Boleh Kosong",
"Perhatian", , MessageBoxButtons.OK, MessageBoxIcon.Error,
MessageBoxDefaultButton.Button1,
MessageBoxOptions.ServiceNotification, False)

        End If

    End Sub
```

```
    Private Sub TBFSumber_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TBFSumber.TextChanged

    End Sub

    Private Sub RbFSumber_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RbFSumber.TextChanged

    End Sub

    Private Sub TBFKunci_TextChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TBFKunci.TextChanged

    End Sub

    Private Sub Label2_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Label2.Click

    End Sub
End Class
```