KEAMANAN PESAN TEKS MENGGUNAKAN KUNCI DENGAN TEKNIK PERMUTASI PADA ALGORITMA *VIGENERE CIPHER*

Skripsi Disusun Dan Diajukan Untuk Memenuhi Persyaratan Ujian Akhir Memperoleh Gelar Sarjana Komputer Pada Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan

SKRIPSI

OLEH

NAMA	**: INDAH PUTRI SUWANDARI**
NPM	**: 1514370939**
PROGRAM STUDI	**: SISTEM KOMPUTER**

UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS DAN TEKNOLOGI
MEDAN
2019

ABSTRAK

Kriptografi berfungsi sebagai menjaga keamanan informasi seperti data rahasia, integritas data, dan autentikasi sebuah data meskipun pihak ketiga dapat membaca pesan tersebut akan tetapi ia sulit untuk dapat memahami isi pesan tersebut. Dalam kriptografi biasanya mempunyai enkripsi dan Deskripsi yang berfungsi untuk menyembunyikan tulisan atau teks dan biasanya kriptogarfi selalu beriringan dengan algoritma. Pada aplikasi ini penulis menggunakan sebuah metode yaitu Algoritma *Vigenere Cipher*, dan teknik permutasi sebagai aturan sebuah pergeseran. Sistem pengamanan tersebut masih mengalami kendala dalam mengamankan data. Salah satunya password mudah diretas karena mudah ditebak atau jumlah karakter yang minim. Dalam hal tersebut penulis berkeinginan Diharapkan dengan adanya aplikasi ini, mahasiswa serta dosen dapat melakukan uji coba enkripsi menggunakan algoritma Vigenere Cipher.

Kata Kunci: Kriptografi, Vigenere.

DAFTAR ISI

Halaman

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yang Maha Esa, yang telah memberikan rahmat-Nya kepada peneliti, sehingga Skripsi ini dapat diselesaikan oleh peneliti tepat pada waktunya dengan judul Keamanan Pesan Teks Menggunakan Kunci Dengan Teknik Permutasi Pada Algorita Vigenere Chiper. Skripsi ini dilakukan guna memenuhi salah satu syarat pemenuhan kurikulum dalam menyelesaikan pendidikan pada Program Studi S1 Sistem Komputer Fakultas Sains Dan Teknologi pada Universitas Pembangunan Panca Budi Medan. Pada kesempatan ini, penulis menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya kepada :

1. Teristimewa kepada Kedua Orang Tua dan Keluarga saya, yang telah banyak memberikan bimbingan dan bantuan baik moril maupun material selama penulis mengikuti pendidikan hingga selesainya Skripsi ini.

2. Rektor Universitas Pembangunan Panca Budi, Dr. H. Muhammad Isa Indrawan, SE., M.M

3. Ibu Sri Shindi Indira selaku Dekan Fakultas Sains Dan Teknologi Universitas Pembangunan Panca Budi Medan.

4. Bapak Dr. Muhammad Iqbal, S.Kom., M. Kom., selaku Ketua Program Studi sekaligus Dosen Pembimbing I.

5. Bapak Andysah Putera Utama Siahaan, S.Kom., M. Kom., ph.D, selaku Dosen Pembimbing II .

6. Kepada Ibu Roslaini Z dan Hirzi Lubis, penulis mengucapkan banyak terima kasih untuk suport dan dukungannya selama proses pembelajaran hingga selesainya Skripsi ini.

7. . Kepada Ratu Gibah *Squad* penulis ucapkan terima kasih atas dukungan dan semangatnya.

Medan, September 2019

Penulis,

INDAH PUTRI SUWANDARI

(1514370939)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Keamanan pada pengiriman pesan berupa teks terkadang sangat banyak memunculkan masalah ketika pengiriman, terkadang pesan yang dikirim tidak lagi berupa bentuk asli dengan adanya pihak ketiga yang mencoba membobol atau mengubah pesan asli tersebut. Oleh karena muncul lah sebuah ilmu yang mempelajari cara menjaga pesan atau data tetap aman dikirim dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga yang biasa disebut dengan kriptografi. Dimana kriptografi itu berfungsi sebagai menjaga keamanan informasi seperti data rahasia, integritas data, dan autentikasi sebuah data meskipun pihak ketiga dapat membaca pesan tersebut akan tetapi ia sulit untuk dapat memahami isi pesan tersebut.

Pada awalnya kriptografi pesan yang digunakan hanyalah bersifat umum, dimana kunci yang digunakan untuk proses enkripsi dan deskripsi sama. Namun ada suatu masalah dari metode ini yaitu pentingnya mendistribusikan kunci yang digunakan dalam keadaan aman. Sebuah cara pun ditemukan untuk mengatasi sebuah kelemahan tersebut dengan model enkripsi yang tidak memerlukan untuk didistribusikan metode ini dikenal dengan nama kunci publik *(public key)*.

Keamanan ini akan dibuat menggunakan metode *vigenere cipher,vigenere cipher* ialah suatu metode penyandian teks dengan menggunakan deretan sandi. Kode sandi pada *vigenere* termasuk *alfabet*. Masalah diatas dapat diatasi jika menggunakan

bantuan komputer yang merupakan sebuah alat yang canggih sehingga kita dapat membuat sebuah media keamanan pesan teks atau data menggunakan *software* atau aplikasi yang ada.

Sehubungan dengan uraian diatas, maka diangkatlah judul skripsi sebagai berikut "**Keamanan Pesan Teks Menggunakan Kunci Dengan Teknik Permutasi Pada Algoritma *Vigenere Cipher*"**. Dimana akan dibuat sebuah media aplikasi yang bertujuan sebagai keamanan pada sebuah pesan teks.

1.2 Rumusan Masalah

Adapun masalah yang akan dibahas dalam skripsi ini yaitu:

a. Seperti apakah bentuk komponen media aplikasi keamanan pesan teks menggunakan metode vigenere cipher ?
b. Bagaimana merancang dan membuat media aplikasi keamanan pesan teks agar keaslian pesan tetap terjaga ?

1.3 Batasan Masalah

Karena keterbatasan dan waktu maka penulis akan membatasi pokok permasalahan yang akan dibahas yaitu:

a. Media aplikasi keamanan pesan teks ini dibuat dengan menggunakan *Visual Basic versi 2010*
b. Keamanan pesan teks ini menggunakan metode *vigenere cipher*.

c. Keamanan ini hanya berupa pesan teks.

1.4 Tujuan Penulisan

Adapun tujuan dari penulisan ini sebagai berikut:

a. Menjelaskan komponen keamanan pada pesan teks.

b. Merancang dan menjelaskan komponen keamanan pesan teks, agar keaslian data tetap terjaga.

1.5 Metode Penelitian

Metode yang akan digunakan dalam perancangan dan pembuatan perangkat lunak ini terdiri dari tahap-tahap berikut:

a. Studi Literatur

Melakukan studi kepustakaan terhadap berbagai referensi yang berkaitan dengan bahasa program *visual basic versi 2010*

b. Perancangan

Pembuatan Hirarki, perancangan layar, dan pembuatan spesifikasi media aplikasi pembelajaran dengan perangkat komputer.

c. Uji Coba

Melakukan proses pengujian pada keamanan pesan teks.

1.6 Sistematika Penulisan

Dalam penyusunan ini, penulis membaginya menjadi 5 bab yang terdiri dari:

I. PENDAHULUAN

Pada pendahuluan akan diuraikan tentang latar belakang, perumusan masalah, batasan masalah, metode penelitian, sistematika penulisan, tujuan dan manfaat.

II. LANDASAN TEORI

Menerangkan tentang teori dasar yang berhubungan dengan judul yang diambil, serta menerangkan dasar yang berhubungan dengan aplikasi yang dirancang serta bahasa pemograman yang digunakan.

III. ANALISA DAN PERANCANGAN SYSTEM

Mengemukakan tentang analisa masalah aplikasi yang akan dirancang dan perancangan aplikasi yang digunakan dalam penulisa skripsi ini.

IV. IMPLEMENTASI DAN HASIL

Mengemukakan tentang implementasi dalam pengujian dari program atau aplikasi yang dibuat pada skripsi ini.

V. PENUTUP

Mengemukakan kesimpulan dari pokok bahasa judul skripsi yang diambil serta saran dari penulis tentang hasil skripsi tersebut.

BAB II

LANDASAN TEORI

2.1 Kriptografi

Kriptografi merupakan seni dan ilmu menyembunyikan informasi dari penerima yang tidak berhak. Kata *cryptographi* berasal dari bahasa yunani yaitu *kryptos* (tersembunyi) dan *graphein* (menulis). *Criptanalysis* adalah aksi untuk memecahkan mekanisme kriptografi dengan cara mendapatkan *plaintext* atau kunci dari *ciphertext* yang digunakan untuk mendapatkan informasi berharga kemudian mengubah atau memalsukan pesan dengan tujuan untuk menipu penerima yang sesungguhnya. Enkripsi adalah mentransformasi data kedalam bentuk yang tidak dapat terbaca tanpa sebuah kunci tertentu. Tujuannya adalah untuk meyakinkan privasi dengan menyembunyikan informasi dari orang – orang yang tidak ditujukan, bahkan dari mereka yang memiliki akses ke data terenkripsi. Deskripsi merupakan kebalikan dari enkripsi yaitu transformasi data terenkripsi kembali ke bentuk semula.

Kriptografi adalah seni dan ilmu untuk menjaga agar pesan rahasia tetap aman (Schneier, 1996). Kriptografi juga salah satu cabang ilmu algoritma matematika. Ada dua tipe dasar dari teknologi kriptografi yaitu *symmetric key (secret / private key) cryptography dan asymmetric key (public key*) cryptography. Pada *symmetric key cryptography* baik pengirim maupun penerima memilki kunci rahasia yang umum. Pada *asymetric key cryptography* pengirim dan penerima masing – masing berbagi kunci *public* dan privat.. Pesan *plaintext* yang telah dienkripsi

(atau dikodekan) dikenal sebagai *ciphertext* (teks sandi). Dalam kriptografi kita akan sering menemukan berbagai istilah atau terminology. Beberapa istilah yang harus diketahui yaitu :

a. Pesan, *Plainteks,* dan *Cipherteks*

Pesan *(message)* ialah data atau informasi yang dapat dibaca dan dimengerti arti dan maknanya. Biasa disebut untuk pesan adalah *(plaintext)* atau teks jelas *(cleartext).*

b. Pengirim dan Penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim *(sender)* ialah entitas yang mengirim pesan kepada entitas lainnya. Penerima *(receiver)* ialah entitas yang menerima pesan.

c. Enkripsi dan dekripsi

Proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi *(encryption)* atau *enciphering* (standard nama menurut ISO 7498-2). Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* semula disebut dekripsi *(decryption)* atau *deciphering* (standard nama menurut ISO 7498-2).

d. *Cipher* dan kunci

Algoritma kriptogarfi disebut juga *cipher,* yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan

dekripsi. Beberapa *cipher* membutuhkan algoritma yang beda untuk enkripsi dan dekripsi. Konsep matematisnya yang didasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen *plaintext* dan himpunan yang berisi *ciphertext*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan P menyatakan plainteks dan C menyatakan cipherteks, maka :

E(P) = C fungsi enkripsi E memetakan P ke C

D(C) = P fungsi dekripsi D memetakan C ke P

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka persamaan D(E(P)) = P harus benar.Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini
algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci *(key)* adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K. Sejarah kriptografi sebagian besar merupakan sejarah kriptografi klasik, yaitu metode enkripsi yang menggunakan kertas dan pensil atau mungkin dengan bantuan alat mekanik sederhana. Adapun aspek- aspek keamanan yaitu :

a. Kerahasiaan *(confidentiality)* ialah fasilitas yang diarahkan untuk menjaga supaya pesan tidak mudah dibaca oleh pihak-pihak yang tidak berhak.

b. Integritas data *(data integrity)* ialah fasilitas yang mengamankan bahwa pesan masih asli atau belum pernah dipalsukan selama pengiriman.

c. Otentikasi *(authentication)* ialah fasilitas yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran bagian yang berkomunikasi *(user autehentication).*

d. *Non-repudiation* ialah fasilitas untuk menjaga entitas yang berkomunikasi melakukan penyangkalan *Advanced Encryption Standard (AES)*

2.2 Fungsi Kriptografi

Fungsi kriptografi dalam teknologi informasi, terus menerus dikembangkan cara untuk menangkal berbagai bentuk serangan seperti penyadapan dan pengubahan data yang dikirimkan. Salah satu cara yang ditempuh mengatasi masalah ini adalah dengan menggunakan kriptografi yang menggunakan transformasi data sehingga data yang dihasilkan tidak dapat dimengerti oleh pihak yang tidak berhak mengakses. Transformasi ini memberikansolusi pada dua macam masalah keamanan data, yaitu masalah privasi *(privacy)* dan keautentikan *(authenticatioan).* Privasi mengandung arti bahwa data yang dikirimkan hanya dapat dimengerti informasinya oleh penerima yang sah atau berhak. Sedangkan keotentikan mencegah pihak ketiga untuk mengirimkan data yang salah atau mengubah data yang dikirimkan.

2.3 Enkripsi

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak

terbaca). Enkripsi dapat diartikan sebagai kode atau *chiper*. Isu- isu yang terkait dengan keamanan dan kerahasiaan data adalah *privacy (*kerahasiaan), *integrity* (keutuhan), *authenticity* (keaslian), *non-repudiation* (pembuktian yang tak tersangkal). Di pertengahan tahun 1970-an, enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah digunakan pada sistem secara luas, seperti Internet *e-commerce,* jaringan Telepon bergerak dan ATM pada bank. Enkripsi dapat digunakan untuk tujuan keamanan. Ilmu yang mempelajari teknik enkripsi disebut kriptografi. Gambaran sederhana tentang enkripsi, misalnya mengganti huruf a dengan n, b dengan m dan seterusnya. Pembahasan enkripsi akan terfokus pada enkripsi password dan enkripsi komunikasi data. Terdapat tiga kategori enkripsi yaitu :

a. Kunci enkripsi rahasia, dalam hal ini terdapat sebuah kunci yang digunakan untuk mengikrisi dan juga sekaligus mendeskripsikan informasi.
b. Kunci enkripsi *public*, dalam hal ini terdapat dua kunci yang digunakan, satu untuk proses enkripsi, satu lagi untuk proses deskripsi.
c. Fungsi *one-way*, dimana informasi dienkripsi untuk menciptakan *"signature"* dari informasi asli yang bisa digunakan untuk keperluan *autentifikasi.*

2.4 Dekripsi

Deskripsi adalah satu kaedah upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti oleh orang yang tidak langsung mengalaminya sendiri. Dalam keilmuan, deskripsi diperlukan agar peneliti tidak melupakan pengalamannya dan agar pengalaman

tersebut dapat dibandingkan dengan pengalaman peneliti lain, sehingga mudah untuk dilakukan pemeriksaan dan kontrol terhadap deskripsi tersebut. Pada umumnya deskripsi menegaskan sesuatu, seperti apa sesuatu itu kelihatannya, bagaimana bunyinya, bagaimana rasanya, dan sebagainya. Deskripsi yang detail diciptakan dan dipakai dalam disiplin ilmu sebagai istilah teknik. Tulisan deskripsi adalah tulisan yang bertujuan untuk menjelaskan sebuag objek secara terperinci tanpa adanya pengaruh pendapat pendapat pengarang di dalam deskripsi tersebut *(andy the gunnerz).*

2.5 Kriteria Keamanan Kriptografi

Sebuah algoritma kriptografi dikatakan aman *(computationally secure)* bila memenuhi tiga kriteria berikut:

a. Persamaan matematis yang menggambarkan operasi algoritma kriptografi sangat kompleks sehingga algoritma tidak mungkin dipecahkan secara analitik.

b. Biaya untuk memecahkan chiperteks melampaui nilai informasi yang terkandung di dalam chiperteks tersebut.

c. Waktu yang diperlukan untuk memecahkan chiperteks melampaui lamanya waktu informasi tersebut harus dijaga kerahasiaannya.

2.6 Penyerangan terhadap Kriptografi

Selain ada pihak yang ingin menjaga agar pesan tetap aman, namun ada juga pihak-pihak yang ingin mengetahui pesan rahasia tersebut secara tidak sah. Bahkan

ada pihak-pihak yang ingin agar dapat mengubah isi pesan tersebut. Ilmu untuk mendapatkan pesan yang asli dari pesan yang telah disandikan tanpa memiliki kunci untuk membuka pesan rahasia tersebut disebut kriptoanalisis. Sedangkan usaha untuk membongkar suatu pesan sandi tanpa mendapatkan kunci dengan cara yang sah dikenal dengan istilah serangan *(attack)*.

2.7 Jenis-Jenis Serangan terhadap Kriptografi

Di bawah ini dijelaskan beberapa macam penyerangan terhadap pesan yang sudah dienkripsi, berdasarkan ketersediaan data yang ada, dan tingkat kesulitannya bagi penyerang, dimulai dari yang paling sulit adalah :

a. *Ciphertext only attack,* penyerang hanya mendapatkan *ciphertext* dari sejumlah pesan yang seluruhnya telah dienkripsi menggunakan algoritma yang sama. Sehingga, metode yang digunakan untuk memecahkannya adalah *exhaustive key search,* yaitu mencoba semua kemungkinan yang ada untuk menemukan kunci.
b. *Known plaintext attack,* dimana penyerang selain mendapatkan sandi, juga mendapatkan pesan asli. Terkadang disebut pula *clear-text attack.*
c. *Choosen plaintext attack,* sama dengan *known plaintext attack*, namun penyerang bahkan dapat memilih penggalan mana dari pesan asli yang akan disandikan. Serangan jenis ini lebih hebat daripada *known-plaintext attack*, karena kriptoanalis dapat memilih plainteks tertentu untuk dienkripsikan, yaitu plainteks-plainteks yang lebih mengarahkan penemuan kunci.
d. *Chosen-ciphertext attack* pada tipe ini, kriptoanalis dapat memilih cipherteks

yang berbeda untuk didekripsi dan memiliki akses atas *plaintext* yang didekripsi.

e. *Chosen-key attack.* Kriptoanalis pada tipe penyerangan ini memiliki pengetahuan tentang hubungan antara kunci-kunci yang berbeda dan memilih kunci yang tepat untuk mendekripsi pesan.

f. *Rubber-hose cryptanalysis* pada tipe penyerangan ini, kriptoanalis mengancam, menyiksa, memeras, memaksa, atau bahkan menyogok seseorang hingga mereka memberikan kuncinya. Ini adalah cara yang paling ampuh untuk mendapatkan kunci.

g. *Adaptive – chosen – plaintext attack* penyerangan tipe ini merupakan suatu kasus khusus *chosen-plaintext attack*. Kriptoanalis tidak hanya dapat memilih plainteks yang dienkripsi, ia pun memiliki kemampuan untuk memodifikasi pilihan berdasarkan hasil enkripsi sebelumnya. Dalam *chosen-plaintext attack*, kriptoanalis mungkin hanya dapat memiliki plainteks dalam suatu blok besar untuk dienkripsi; dalam *adaptive-chosen-plaintext attack* ini ia dapat memilih blok plainteks yang lebih kecil dan kemudian memilih yang lain berdasarkan hasil yang pertama, proses ini dapat dilakukannya terus menerus hingga ia dapat memperoleh seluruh informasi.

2.8 aplikasi

Aplikasi Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanankan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus *computer* eksekutif, aplikasi mempunyai arti yaitu

pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, "Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu"

2.9 jenis- jenis metode kriptografi

Adapun beberapa jenis dari kriptografi yaitu :

a. Permutasi adalah memindahkan atau merotasikan karakter dengan aturan tertentu. Sebagai contoh : huruf – huruf plaintext A T T A C K A T D A W N dapat dipermutasi jadi D C K A A W N A T A T T . Ciphertransposisi kolumnar adalah cipher dimana plaintext ditulis secara horizontal pada kertas dan dibaca secara vertikal. Cipher dapat diserang melalui analisis frekuensi, namun cipher menyembunyikan properti statistik dari pasangan huruf – huruf seperti IS dan TOO (Hartono, 2007).

b. Substitusi Caesar cipher adalah cipher substitusi sederhana yang mencakup pergeseran alfabet 3 posisi ke kanan. Caesar cipher merupakan subset dari cipher polialfabetik vigenere. Pada Caesar cipher karakter – karakter dan pengulangan kunci dijumlahkan bersama, modulo 26. Dalam penjumlahan modulo 26 huruf – huruf A – Z dari alfabet masing – masing memberikan nilai 0 sampai 25. Tipe cipher ini dapat diserang dengan menggunakan analisis frekuensi. Dalam frekuensi analisis digunakan karakteristik frekuensi yang tampak dalam penggunaan huruf – huruf

alfabet pada bahasa tertentu. Tipe *crytanalysis i*ni dimungkinkan karena Caesar cipher adalah monoalfabetik cipher atau cipher substitusi sederhana, dimana karakter ciphertext disubtitusi untuk setiap karakter plaintext. Serangan ini dapat diatasi dengan menggunakan substitusi polialfabetik. Substitusi polialfabetik dicapai melalui penggunaan beberapa cipher substitusi, namun substitusi ini dapat diserang dengan penemuan periode, saat substitusi berulang kembali (Hartono, 2007).

c. *Vernam Cipher (One Time Pad)* Cipher ini diimplementasikan melalui sebuah kunci yang terdiri dari sekumpulan random karakter – karakter yang tidak berulang. Setiap huruf kunci dijumlahkan modulo 26 dengan huruf plainext. Pada One Time Pad tiap huruf kunci digunakan satu kali untuk satu pesan dan tidak digunakan kembali. Panjang stream karakter kunci sama dengan panjang pesan (Hartono, 2007).

d. *Book Key Cipher / Running Key Cipher Cipher* ini menggunakan teks dari sebuah sumber (misalnya buku) untuk mengenkripsi *plaintext.* Kunci diketahui oleh pengirim dan penerima yang dimaksud dapat berupa halaman dan jumlah baris dari teks pada buku. Teks ini adalah karakter yang sesuai untuk karakter dengan *plaintext,* dan penjumlahan modulo 26 dijalankan untuk mempengaruhi enkripsi. *Running key* cipher mengeliminasi priodisitas, namun masih dapat diserang dengan memanfaatkan redudansi pada kunci (Hartono, 2007).

e. *Codes Codes* berkaitan dengan kata – kata dan frase dan menghubungkan kata – kata ini sebagai frase untuk sekelompok angka atau huruf. Sebagai contoh angka 526 dapat berarti "*Attack at dawn"* (Hartono, 2007).

f. *Steganography* Adalah seni menyembunyikan keberadaan pesan. *"Steganography"* berasal dari kata Yunani *"steganos*" yang berarti "terlindungi" dan *"graphein"* yamg berarti "menulis". Sebuah contohnya adalah microdot yang mengkompresi pesan kedalam ukuran period atau dot. *Steganography* dapat digunakan untuk membuat *"watermark"* digital untuk mendeteksi penyalinan *image* digital secara illegal (Hartono, 2007).

2.10 Visual Basic

Bahasa Pemrograman Visual Basic 6.*0 Microsoft Visual Basic* merupakan bahasa pemrograman yang berbasis *Ms-Windows*, sebagai bahasa pemrograman yang mutakhir*, Microsoft Visual Basic 6.0* didesain untuk memanfaatkan fasilitas yang tersedia dalam *Ms-Windows. Microsoft Visual Basic 6.0* juga merupakan bahasa pemrograman *Object Oriented Programming (OOP),* yaitu pemrograman yang berorientai objek. Visual Basic merupakan salah satu *software* untuk membuat program yang cukup sederhana tetapi banyak cakupan yang dapat dikerjakan, karena visual basic dapat mengakses banyak *software* seperti *Excel, Access* dan sebagainya. Visual Basic lebih sederhana dari pemrograman yang lain. Kesederhanaan visual basic terletak pada kemudahan membuat bahasa pemrograman dan bentuk tampilan yang dikehendaki. Visual Basic ini merupakan pengembangan bahasa baic yang diterapkan pada program yang berbasis *windows. Visual Basic 6.0* adalah salah satu *development tools* untuk membangun aplikasi dalam lingkungan *windows.* Dalam pengembangan aplikasi, visual basic menggunakan pendekatan visual untuk merancang *user interface* atau tampilan dalam bentuk *form,* sedangkan untuk

kodingnya menggunakan bahasa basic yang cenderung mudah dipelajari. Visual basic telah menjadi *tools* yang terkenal bagi para pemula maupun *developer.[6]*

Sejarah Perkembangan Visual Basic 6.0 Perkembangan visual basic sangat pesat dikarenakan pemakaiannya yang mudah dan juga dikarenakan banyaknya fasilitas-fasilitas yang disediakan visual basic. Perkembangan yang pesat dapat dilihat dari sejarah perkembangan visual basic tersebut. Berikut ini akan menjelaskan point-point penting sejarah perkembangan bahasa pemrograman visual basic, yaitu :
a. Visual Basic pertama kali dikeluarkan pada tahun1991 yaitu program visual basic untuk DOS dan unutk *Windows.*

b. Pada tahun 1993 visual basic 3.0 diliris. Pada akhir tahun 1994 visual basic 4.0 dengan tambahan untuk mendukung aplikasi 32 bit.

c. Pada akhir tahun 1998 visual basic 6.0 diliris.

d. Pada tahun 2002, versi terbaru dari visual basic diliris yaitu versi visual basic.Net.
Adapun obyek-obyek yang dipergunakan dalam program ini adalah :

a. *Project*

Project adalah sekumpulan modul. Jadi project merupakan aplikasi itu sendiri. *Project* disimpan dalam file yang berakhiran VBP. Jika kita akan melaksanakan pembuatan program aplikasi, akan terdapat jendela *project* yang berisi semua *file* yang dibutuhkan menjalankan program aplikasi *Visual Basic.net* pada saat pembuatan program aplikasi baru maka jendela *project* otomatis akan berisi object form1. Pada jendela *project* terdapat tiga *icon* yaitu *View Code, View Object,* dan *Toggle Folders*. Icon

View Code dipakai untuk menampilkan jendela editor kode program. *Icon View Object* dipakai untuk menampilkan bentuk formulir *(form)* dan *icon Toggle Folders* digunakan untuk menampilkan folder

b. *Form*

Form adalah jendela yang dipakai untuk membuat *user interface/*tampilan. Secara otomatis akan tersedia *form* yang baru jika membuat suatu program aplikasi yang baru, dengan nama *Form1*. pada umumnya dalam suatu form terdapat garis titik-titik yang disebut dengan *Grid*. Untuk lebih memahami form ini maka di bawah ini terdapat gambar jendela form.

c. *Toolbox*

Toolbox adalah kumpulan dari obyek yang digunakan untuk membuat *user interface* (tampilan) serta *control* bagi program aplikasi. Untuk menempatkan *control* pada suatu *form* dapat dilakukan dengan klik ganda *control* dalam toolbox, kemudian mengubah besar dan ukurannya serta memindahkannya dengan metode *Drag* and *Drop* atau dengan cara mengklik kontrol *toolbox*, kemudian pindahkan pointer *mouse* jendela *form.* Kursor berubah menjadi Crosshair lalu tempatkan pada sudut kiri atas dimana kita inginkan kontrol tersebut diletakkan, tekan tombol *mouse* kiri dan tahan ketika menyeret kursor ke arah sudut kanan bawah.

d. *Properties*

Properties berisikan daftar struktur *setting* properti yang digunakan pada sebuah object terpilih. Kotak *drop-down* pada bagian atas jendela

berisi daftar semua object pada *form* yang aktif. Ada tab tampilan, yaitu *alphabetic* (urut abjad) dan categorized (urut berdasarkan kelompok).

e. Kode Program

Kode program adalah serangkaian tulisan perintah yang akan dilaksanakan jika suatu obyek dijalankan. Kode program ini mengkontrol dan menentukanjalannya suatu obyek.

f. *Event*

Event adalah peristiwa atau kejadian yang diterima suatu obyek, misalnya klik, seret, tunjuk, dan lain sebagainya.

g. Metode *(Methods)*

Metode adalah serangkaian perintah yang sudah tersedia pada suatu obyek yang dapat diminta untuk mengerjakan tugas khusus.

h. *Module*

Module dapat disejajarkan dengan *form,* tetapi *module* tidak mengandung obyek. *Module* berisikan prosedur umum, deklarasi variabel dan definisi konstanta yang digunakan oleh aplikasi.

2.11 . Vigenere Chipper

a. Konsep Dasar Vigenere

Cipher merupakan algoritma kriptografi klasik. Operasi pada algoritma kriptografi klasik berbasis pada operasi karakter, sedangkan operasi pada algoritma kriptografi *modern* berbasis pada operasi bit. Dalam kriptografi klasik, Vigenere Cipher termasuk ke dalam cipher substitusi abjad majemuk, yang terbuat dari sejumlah cipher abjad tunggal, masing-

masing dengan kunci yang berbeda. Vigenere Cipher telah berkali-kali diciptakan ulang dengan cukup bervariasi. Namun, metode aslinya digambarkan oleh Giovan Batista Belaso pada tahun 1553 seperti tertulis di dalam bukunya *La Cifra del Sig. Giovan Batista Belaso*. Meskipun demikian, Vigenere Cipher dipopulerkan oleh Blaise de Vigenere pada tahun 1586.

2.12 Keamanan Informasi

Keamanan informasi adalah mengamankan suatu aset yang berharga bagi kelangsungan hidup organisai baik Pemerintah maupun non Pemerintah. Aset tersebut adalah sebuah informasi. Keamanan informasi adalah hal yang harus diutamakan dan diperhatikan oleh organisasi dari tindakan kriminal yang ilegal oleh pihak yang tidak berwenang. Pertama kebocoran informasi, kedua merubah dan memanipulasi aset atau informasi, merusak sistem yang dapat menyebabkan kerugian baik dari sisi finansial maupun produktifitas organisasi. Melindungi keamanan informasi sebaik mungkin selalu ada upaya yang harus dilakukan. Keamanan informasi dalam suatu organisasi memiliki beberapa aspek-aspek, diantaranya adalah :

a. *Confidentiality* (Kerahasiaan) Merupakan keamanan informasi yang menjamin, memastikan dan menjaga kerahasiaan aset, bahwa hanya dapat diakses oleh mereka yang memiliki wewenang.
b. *Integrity* (Integritas) Merupakan keamanan informasi yang menjamin kelengkapan aset, menjamin asset tersebut tidak berubah

dan dimodifikasi maupun dihilangkan tanpa otorisasi yang tidak jelas. Menjaga keakuratan dan ancaman dari pihak luar yang tidak berkepentingan.

c. *Availibility* (Ketersediaan) Merupakan keamanan informasi yang menjamin bahwa aet tetap tersedia, dapat diakses ketika dibutuhkan tanpa adanya gangguan dari pihak lain. Proses berasal dari kebutuhan organisasi, implementasi yang baik untuk menyelesaikan dan menyediakan kebutuhan.

d. Teknologi Teknologi membantu organisasi untuk meningkatkan keamanan yang berasal dari perangkat keras dan lunak agar membuat proses lebih efisien.

2.13 Pengertian UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET (Prastuti Sulistyorini, 2012).

Unified Modeling Language (UML) adalah kumpulan notasi grafis yang didukung oleh sebuah model tunggal, yang membantu dalam menjelaskan dan merancang sistem perangkat lunak, khususnya sistem perangkat lunak dibangun menggunakan gaya berorientasi objek. UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis kedalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang kita sebut sebuah model (Andy Prasetyo Utomo, 2013).

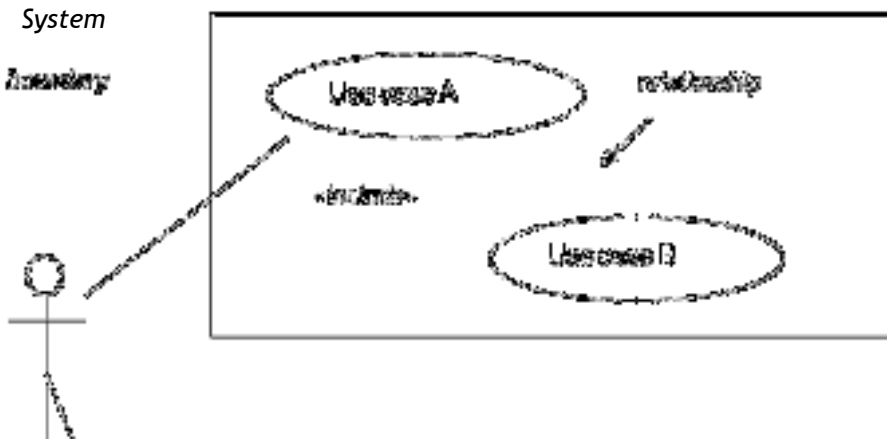
Dengan menggunakan model ini diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti *scalability*, *robustnees*, *security*, dan sebagainya. Untuk melakukan pemodelan sistem perangkat lunak secara visual digunakan UML (*Unified Modelling Language*) yang digambarkan secara elektronik lewat sarana perangkat lunak *Rational Rose*. Sebagai mana telah diterapkan oleh Gufran (2012) di mana UML diterapkan untuk mengukur kinerja mahasiswa menggunakan pendekatan berorientasi objek. Kemudian UML diterapkan juga oleh Sunguk (2012) untuk menerapkan sistem *database* dan aplikasi komputer. Selanjutnya Jakimi dan Koutbi (2009) menerapkan pendekatan UML untuk sekenario rekayasa dan kode generasi.

2.14 Use Case Diagram

Use case merupakan teknik menangkap kebutuhan-kebutuhan fungsional dari sistem baru atau sistem yang diubah. Setiap *use case* terdiri dari satu atau lebih skenario yang menerangkan bagaimana sistem berinteraksi dengan pengguna atau sistem yang lain untuk mencapai suatu sasaran bisnis tertentu. Dalam tehnik ini tidak diterangkan cara kerja sistem secara internal maupun implementasinya. Yang ditunjukkan adalah langkah-langkah yang dilakukan pengguna dalam menggunakan perangkat lunak (Nyimas Artina, 2006).

Diagram *Use Case* merupakan diagram yang menggambarkan fungsi berupa komponen, kelas, atau kejadian yang ada dalam *system* (Ade Sutedi *et al*, 2015). *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa A.S dan M. Shalahuddin, 2014).

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.



Gambar 2.1 ***Use Case Diagram***

Terdapat 2 bagian utama dalam *use case modeling* sebagaimana dijelaskan sebagai berikut:

a. Aktor

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

Gambar 2.2 Aktor

b. Use Case

Use case merupakan fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*.

Gambar 2.3 *Use Case*

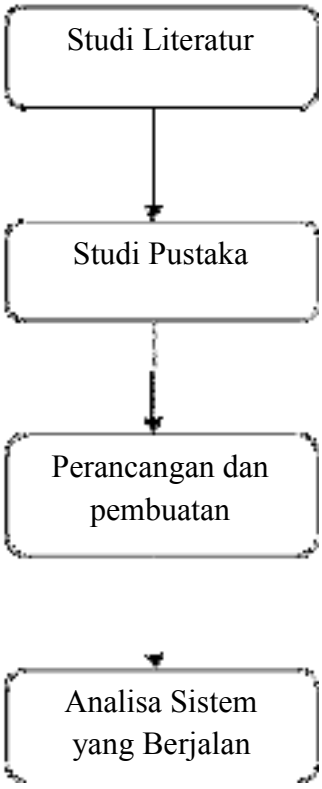
2.15 Activity Diagram

Activity diagrams menggambarkan *workflow* (aliran kerja) atau aktivitas sari sebuah sistem atau proses bisnis. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa A.S dan M. Shalahuddin, 2014).

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

3.2 Metode Pengumpulan Data

1. Studi Literatur

Pada pengumpulan data penulis melakukan beberapa cara yaitu *literatur,* jurnal, dan berbagai bacaan- bacaan yang berkaitan dengan judul penelitian tersebut. Dalam penelitian di lakukan beberapa metode yaitu :

a. Mempelajari studi *literatur* tentang proses enkripsi dan dekripsi.

b. Mempelajari studi *literatur* tentang pemrograman.

c. Mempelajari studi *literatur* tentang *vigernere chiper*.

2. Studi Pustaka

Pengumpulan data ini yang dilakukan menggunakan atau mengumpulkan sumber-sumber yang tertulis, dengan cara membaca, mempelajari dan mencatat hal yang penting sehubungan dengan penelitian tersebut. Untuk mendapatkan data yang diperlukan untuk melengkapi kesempurnaan tugas akhir ini adalah sebagai berikut :

a. Observasi

Pada pengumpulan data atau informasi dilakukan dengan pengamatan pada objek kajian yang bertujuan untuk mendapatkan data tentang suatu masalah, sehingga mendapatkan hasil pemahaman atau sebagai pembuktian pada informasi atau keterangan yang di dapat dari sebelumnya.

b. Teknik Analisis Data

Analisis data ialah kegiatan yang dilakukan untuk mengubah data hasil dari suatu penelitian menjadi data atau informasi yang nantinya bisa dipergunakan untuk menjadikan sebuah kesimpulan.

3.3 Analisa Sistem yang Berjalan

Analisa ialah suatu kegiatan penguraian dan penyelidikan pada sebuah inti masalah agar mendapatkan suatu pemahaman, pengertian dan arti sebenarnya dari sebuah inti permasalahan tersebut. Pada sebuah keamanan komputer mempunyai sebuah istilah enkripsi, yang mana enkripsi ialah termasuk salah satu jenis yang menggunakan metode *ciphertext*. Agar memperoleh hasil teks yang sudah diubah *(ciphertext),* menggunakan angka dan table untuk konversi. Algoritma *Vigener Chiper* ialah sebuah metode keamanan informasi dengan menambah *plaintext* dengan kunci hingga menghasilkan *ciphertext* yang memiliki sifat *kongruen*.

3.4 Kelemahan Sistem Yang Berjalan

Pada system ini mempunyai kelemahan, kelemahan algoritma vigenere cipher muncul jika panjang kunci lebih pendek dari panjang plainteksnya sehingga terdapat perulangan kunci yang digunakan untuk mengenkripsi plainteks tersebut. Kunci yang berulang tersebut menimbulkan celah berupa jumlah pergeseran yang sama untuk setiap plainteks yang disubstitusi oleh huruf pada kunci yang sama sehingga huruf-huruf pesan atau plainteks dapat dikelompokkan berdasarkan kunci yang digunakan. Karena terdapat kelompok huruf-huruf plainteks yang disubstitusi dengan huruf kunci yang sama karena perulangan kunci, maka tiap kelompok huruf-huruf tersebut dapat dikenakan metode analisis frekuensi terhadapnya. Sistem pengamanan tersebut masih mengalami kendala dalam mengamankan data. Salah satunya password mudah diretas karena mudah ditebak atau jumlah karakter yang minim.

3.5 Sistem Yang Diusulkan

Adapun system yang diusulkan berupa sebuah system keamanan pesan teks menggunakan kunci dengan teknik permutasi pada algoritma *vigenere chipper* yang berguna untuk sebuah pesan teks dimana hanya ada satu kunci dan menggunakan fungsi perulangan agar sulit untuk ditebaknya sebuah kunci.

3.6 Metode Yang Pernah Ada

Affine Cipher Affine cipher pada metode affine adalah perluasan dari metode Caesar Cipher, yang mengalikan plainteks (P) dengan sebuah nilai (a) dan menambahkannya dengan sebuah pergeseran (k). P menghasilkan cipherteks C dinyatakan dengan fungsi kongruen:

$C=((a \times P) +k) \bmod 26$……(1)

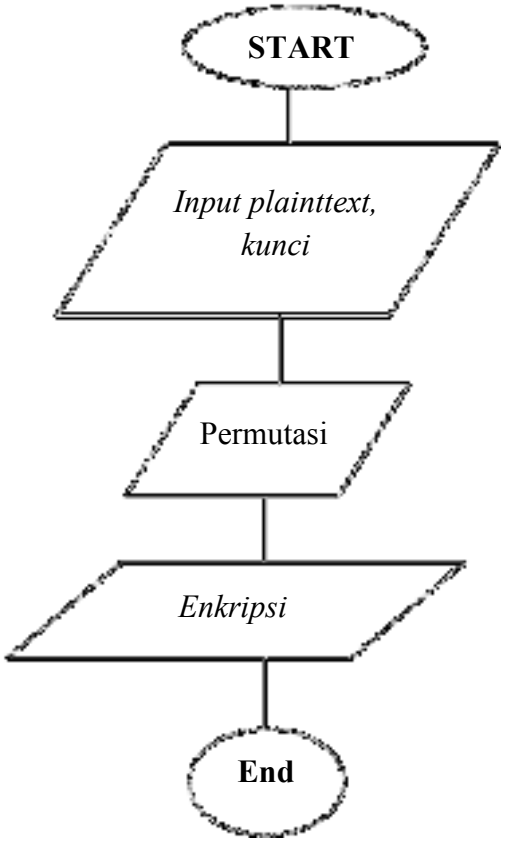
Dimana 26 adalah jumlah alphabet, persamaan 1 digunakan pada proses enkripsi. Proses dekripsi menggunakan persamaan 2 di bawah ini : $P=a\text{-}1\ (Ci\text{-}k) \bmod 26$ ……..…(2) a adalah

bilangan bulat yang harus relatif prima dengan 26. Dengan kata lain great common divisior gcd(a,26) harus sama dengan 1.

3.7 Rancangan Penelitian

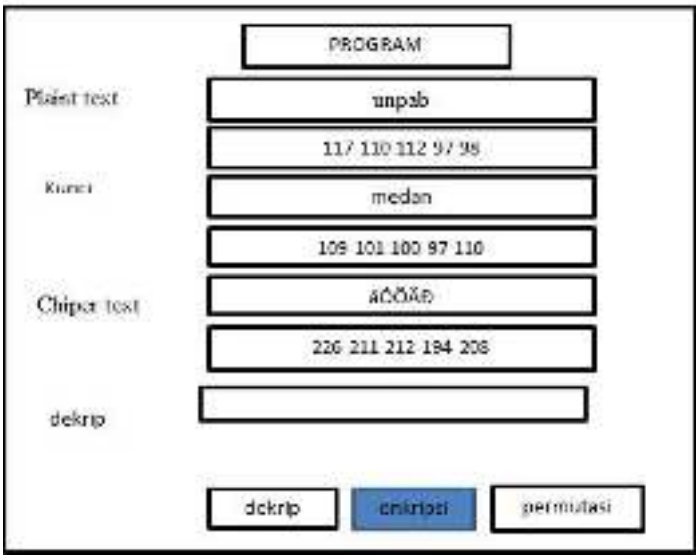
Mengenkripsi sebuah pesan teks atau informasi pada teori diatas dapat dilakukan menggunakan *Vigenere Cipher* dan membutuhkan sebuah permutasi dimana permutasi tersebut menentukan pergeseran, *a* adalah nilai pada *plaintext* yang akan ditambahkan dengan sebuah nilai *k* yaitu kunci. Program ini menggunakan *mod* 256 , kunci dan permutasi dapat mejaga dan membuat informasi lebih aman dan terjaga kerahasiaannya serta keasliannya sehingga sulit terdeteksi oleh pihak- pihak yang tidak berwenang dikarenakan penyandian tidak hanya bisa menyandikan huruf akan tetapi bisa juga digunakan untuk angka, symbol, tanda baca dan lain-lain. Dalam algoritma ini pemilihan kunci dilakukan secara acak dengan beberapa peluang agar dapat menemukan kunci yang sesuai dengan sifat algoritma *Vigenere Cipher* .

Proses Enkripsi



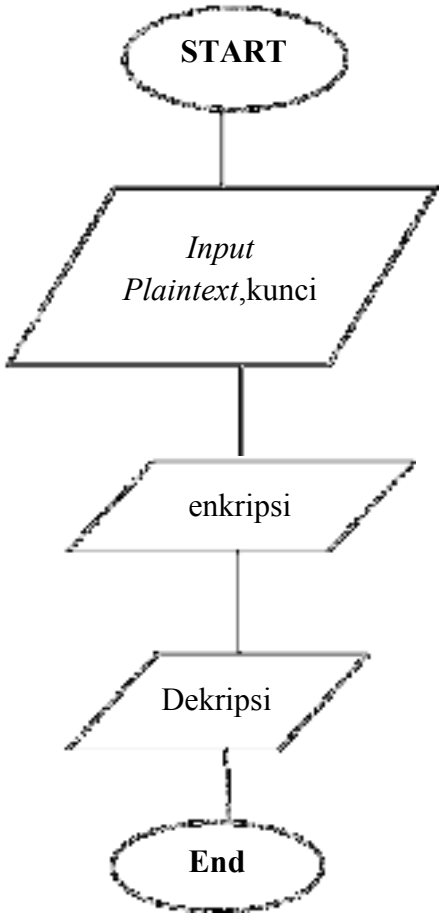
Gambar 3.2 Flowchart Proses Enkripsi pesan

Proses enkripsi dilakukan dengan memasukan kata atau sandi pada plaintext yang nanti nya nilai pada plaintext akan ditambah pada nilai yang terpada pada kunci. Nilai yang dimaksud pada teori ini adalah nilai yang telah dikonversi pada bilangan ASCII, setelah di enkripsi ditambahkan dengan pergeseran dengan *mod* 256. Lalu akan menampilkan hasil yang berupa huruf, angka atau symbol acak yang tidak dapat diketahui maknanya oleh pihak ketiga. Agar lebih jelas dapat dilihat gambar dibawah ini :



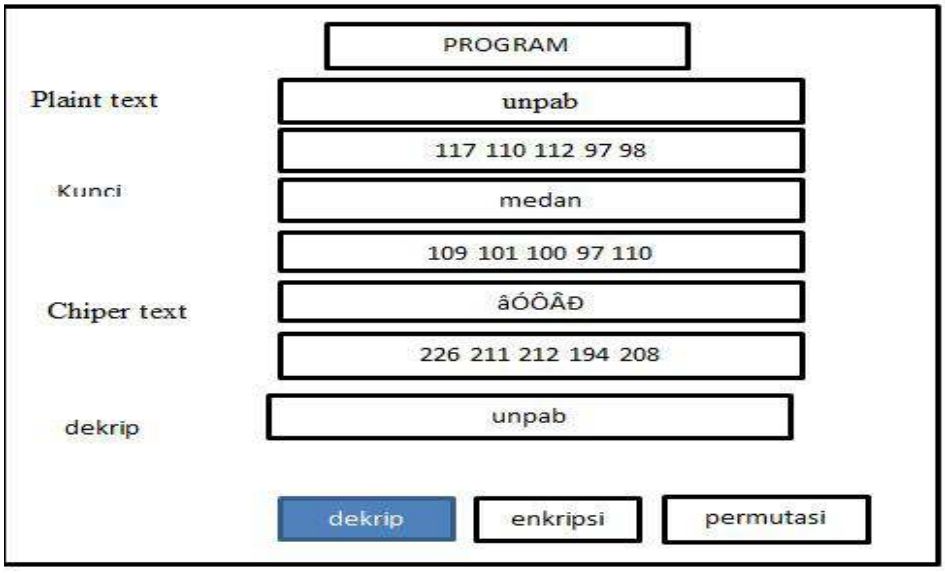
Gambar 3.3 Tampilan Proses Enkripsi.

3.4.1 Proses Dekripsi



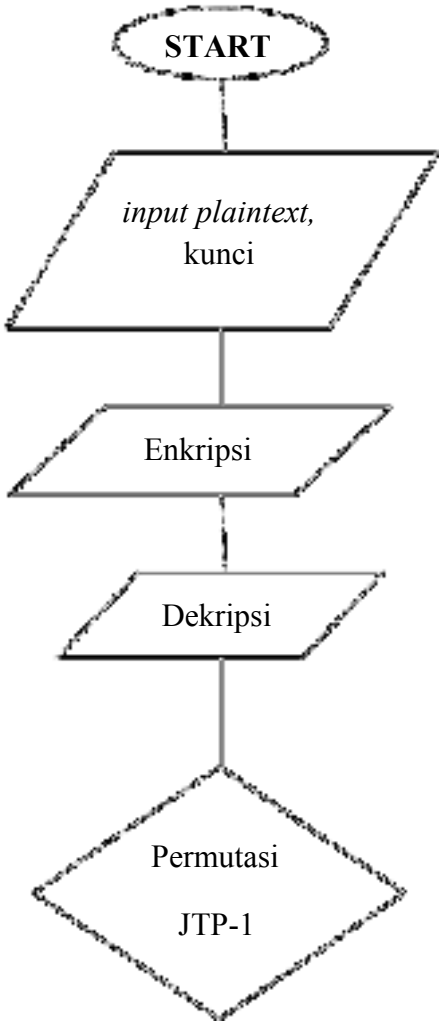
Gambar 3.4 Flowchart Proses Dekripsi

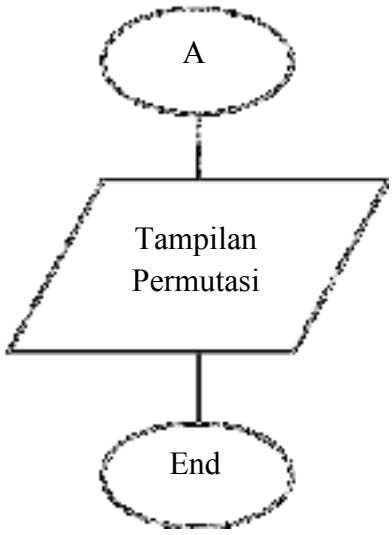
Dalam proses dekripsi awalnya memasukan kata atau pesan dan memasukan kunci agar setelah di enkripsi muncullah nilai pada bilangan ASCII yang terdapat pada plainttext, kunci, dan chipertext kemudian hasil *plaintext* yang telah dienkripsi akan ditambahkan dengan nilai kunci lalu *mod* 256 sehingga akan menghasilkan pesan atau teks seperti semula. Agar lebih jelas perhatikan gambar berikut ini :



Gambar 3.5 Tampilan Hasil Dekripsi..

3.8 Pembangkit Kunci





Gambar 3.6 Flowchart Pembangkitan Kunci

Pada algoritma *Vigenere Cipher* hanya memiliki satu kunci rahasia saja untuk mengubah *plaintext* menjadi *ciphertext* pembangkitan kunci pergeseran kemudian enkripsi. Kunci tersebut dapat dibangkitkan apabila kunci yang akan digunakan memenuhi syarat tukar posisi yaitu :

$$XY = 0sp\ (jtp - 1)$$

Dimana pertukaran posisi dilakukan secara acak, pada jumlah tukar posisi jumlah pertukaran posisi akan sepanjang dengan kunci tersebut. Pemilihan kunci tidak hanya berupa huruf, namun bisa juga dengan angka maupun simbol dengan demikian semakin sulit kuncinya maka akan semakin sulit seorang kriptanalis menebak kunci tersebut.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada tahap implementasi sistem ini merupakan sebuah tahap aplikasi yang sudah dirancang dan dijalankan. Tahap tersebut menunjukkan setiap proses yang sedang berjalan dan mampu bekerja sesuai yang diharapkan. Proses rancangan ini menggunakan *visual basic net 2010* yang ditampilkan dalam sebuah form – form agar menjadi sarana bagi penggunanya dalam melakukan proses implementasi.

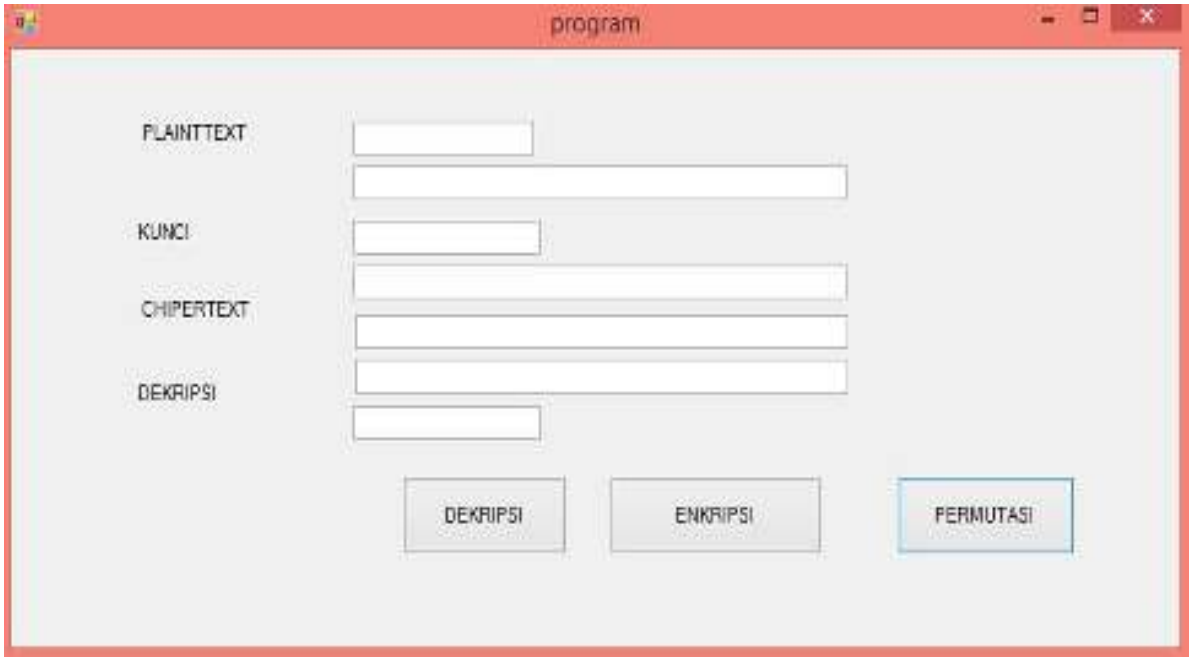
4.2 Pengujian Sistem

Dalam pengujian sebuah sistem memiliki tujuan agar dapat menemukan kesalahan fungsi pada aplikasi yang dibangun dan memperbaikinya, selain itu pengujian sistem dilakukan untuk mengetahui apakah sistem dapat berjalan sesuai yang diharakan.

Pengujian ini dapat dilakukan dengan sebuah teks selanjutnya diproses oleh aplikasi apakah aplikasi tersebut dapat memberikan hasil yang sesuai. Proses yang akan dilakukan pengujian dalam aplikasi ini adalah simulasi pengiriman pesan berupa teks dengan menggunakan metode algoritma vigenere cipher dengan menggunakan satu kunci hingga pada akhirnya keaslian pesan tetap terjaga.

4.2.1 Tampilan awal/ Home

Pada tampilan gambar dibawah merupakan tampilan awal ketika aplikasi dijalankan. Pada form ini terdapat beberapa form dengan fungsi masing-masing, selain itu terdapat beberapa tombol yaitu : dekripsi, enkripsi, permutasi yang masing-masing juga memiliki fungsi yang berbeda.



Gambar 4.1 Tampilan Awal

4.2.2 Tampilan Enkripsi

Tampilan enkripsi ini dilakukan dengan memasukan teks pada *plainttext* dan kunci, lalu tekan tombol enkripsi. Pada tahap ini *plainttext* akan dirubah menjadi *chiperttext* menggunakan algoritma vigenere chiper. Seperti contoh gambar dibawah ini memiliki hitungan yaitu :

1. *Plaintext* = unpab
2. kunci = medan

Artinya *plaintext* akan tambahkan pada kunci, berikut hitungan :

1. u + m mod 256

   117 + 109

   = 226 mod 256

2. n + e mod 256

   110 + 101

   = 211 mod 256

3. p + d mod

   256 112 + 100

   = 212 mod 256

4. a + a mod 256

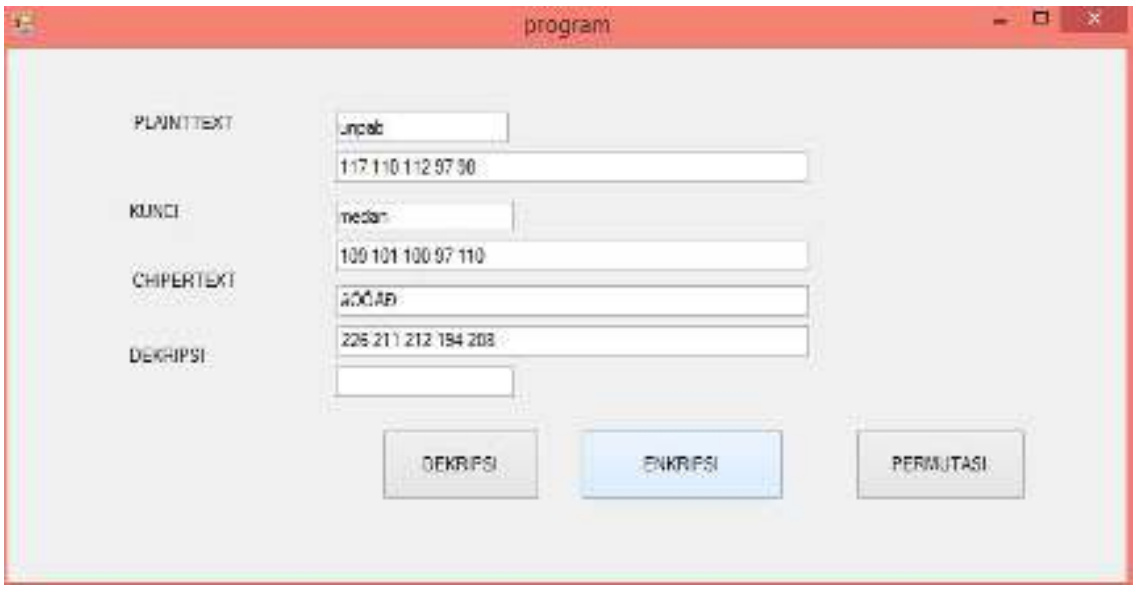
   97+97

   = 194 mod256

5. b + n mod

   256 98 + 110

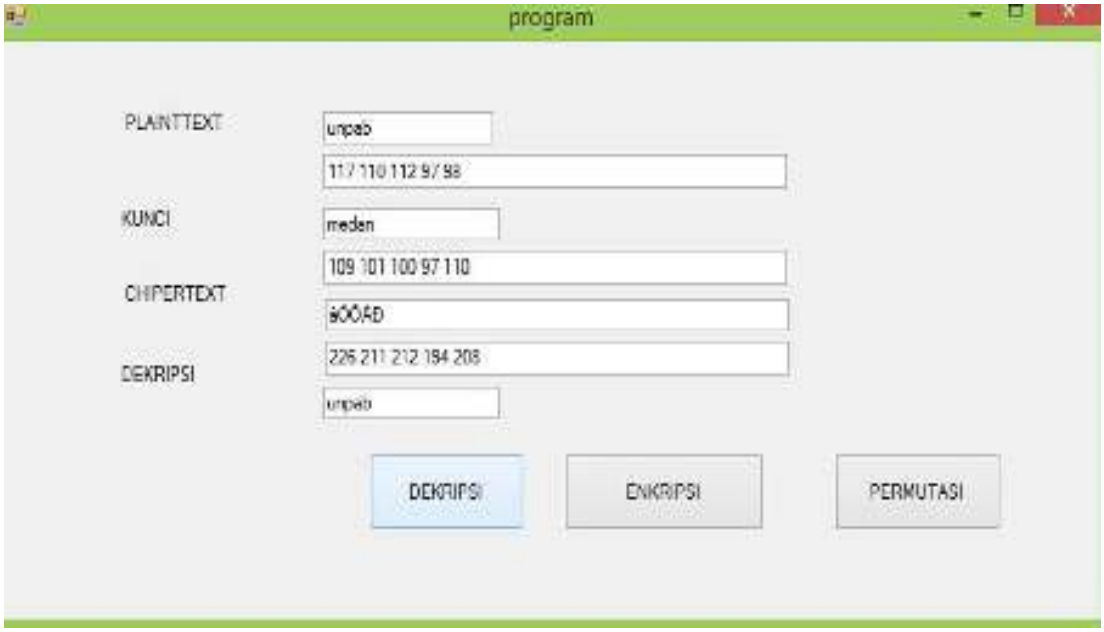
   = 208 mod 256



Gambar 4.2 Tampilan Enkripsi

4.2.3 Tampilan Dekripsi

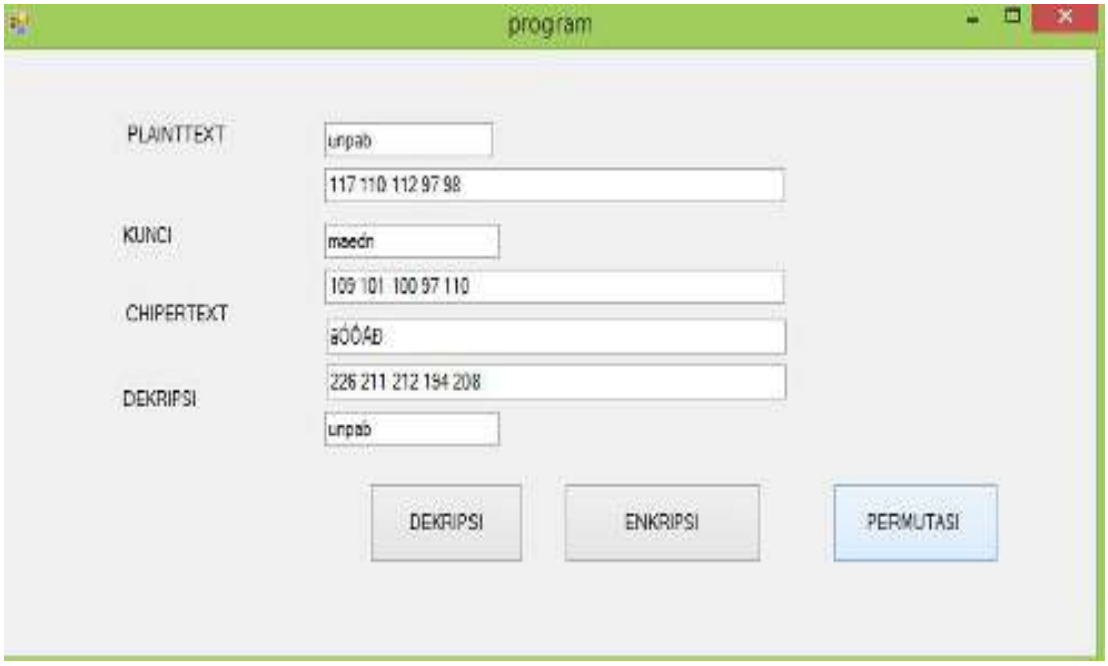
Tampilan dekripsi ini dilakukan agar mengetahui maksud dari pesan teks yang telah dirubah yang sulit diartikan, pada tahap ini proses dilakukan setelah melakukan enkripsi.



Gambar 4.3 Tampilan Dekripsi

4.2.4 Tampilan Permutasi

Tampilan permutasi ini dilakukan setelah melakukan tahap enkripsi dan dekripsi, yang mana permutasi tersebut berfungsi untuk mengacak sebuah kunci sepanjang isi yang dimiliki kunci tersebut.



Gambar 4.4 Tampilan Permutasi

4.3 Validasi Sistem

4.3.1 Hasil Perhitungan Manual Proses Enkripsi.

Pada tabel dibawah ini adalah tabel konversi yang berfungsi untuk menghitung manual proses enkripsi dan permutasi.

Tabel 4.1 Tabel ASCII I *(American Standard Code for Information Interchange)*

DEC	BIN	Symbol	Description
0	00000000	NUL	*Null char*
1	00000001	SOH	*Start of Heading*
2	00000010	STX	*Start of Text*
3	00000011	ETX	*End of Text*

4	00000100	EOT	*End of Transmission*
5	00000101	ENQ	*Enquiry*
6	00000110	ACK	*Acknowledgment*
7	00000111	BEL	*Bell*
8	00001000	BS	*Back Space*
9	00001001	HT	*Horizontal Tab*
10	00001010	LF	*Line Feed*
11	00001011	VT	*Vertical Tab*
12	00001100	FF	*Form Feed*
13	00001101	CR	*Carriage Return*
14	00001110	SO	*Shift Out / X-On*
15	00001111	SI	*Shift In / X-Off*
16	00010000	DLE	*Data Line Escape*
17	00010001	DC1	*Device Control 1 (oft. XON)*
18	00010010	DC2	*Device Control 2*
19	00010011	DC3	*Device Control 3 (oft. XOFF)*
20	00010100	DC4	*Device Control 4*
21	00010101	NAK	*Negative Acknowledgement*
22	00010110	SYN	*Synchronous Idle*
23	00010111	ETB	*End of Transmit Block*
24	00011000	CAN	*Cancel*

25	00011001	EM	*End of Medium*
26	00011010	SUB	*Substitute*
27	00011011	ESC	*Escape*
28	00011100	FS	*File Separator*
29	00011101	GS	*Group Separator*
30	00011110	RS	*Record Separator*
31	00011111	US	*Unit Separator*

32	00100000		*Space*
33	00100001	!	*Exclamation mark*
34	00100010	"	*Double quotes (or speech marks)*
35	00100011	#	*Number*
36	00100100	$	*Dollar*
37	00100101	%	*Per cent sign*
38	00100110	&	*Ampersand*
39	00100111	'	*Single quote*
40	00101000	(	*Open parenthesis (or open bracket)*
41	00101001	)	*Close parenthesis (or close bracket)*
42	00101010	*	*Asterisk*
43	00101011	+	*Plus*

44	00101100	,	*Comma*
45	00101101	-	*Hyphen*
46	00101110	.	*Period, dot or full stop*
47	00101111	/	*Slash or divide*
48	00110000	0	*Zero*
49	00110001	1	*One*
50	00110010	2	*Two*
51	00110011	3	*Three*
52	00110100	4	*Four*
53	00110101	5	*Five*
54	00110110	6	*Six*
55	00110111	7	*Seven*
56	00111000	8	*Eight*
57	00111001	9	*Nine*
58	00111010	:	*Colon*
59	00111011	;	*Semicolon*
60	00111100	<	*Less than (or open angled bracket)*
61	00111101	=	*Equals*
62	00111110	>	*Greater than (or close angled bracket)*
63	00111111	?	*Question mark*
64	01000000	@	*At symbol*

65	01000001	A	*Uppercase A*
66	01000010	B	*Uppercase B*
67	01000011	C	*Uppercase C*
68	01000100	D	*Uppercase D*
69	01000101	E	*Uppercase E*
70	01000110	F	*Uppercase F*
71	01000111	G	*Uppercase G*
72	01001000	H	*Uppercase H*
73	01001001	I	*Uppercase I*
74	01001010	J	*Uppercase J*
75	01001011	K	*Uppercase K*
76	01001100	L	*Uppercase L*
77	01001101	M	*Uppercase M*
78	01001110	N	*Uppercase N*
79	01001111	O	*Uppercase O*
80	01010000	P	*Uppercase P*
81	01010001	Q	*Uppercase Q*
82	01010010	R	*Uppercase R*
83	01010011	S	*Uppercase S*
84	01010100	T	*Uppercase T*
85	01010101	U	*Uppercase U*

86	01010110	V	*Uppercase V*
87	01010111	W	*Uppercase W*
88	01011000	X	*Uppercase X*
89	01011001	Y	*Uppercase Y*
90	01011010	Z	*Uppercase Z*
91	01011011	[	*Opening bracket*
92	01011100	\	*Backslash*
93	01011101	]	*Closing bracket*
94	01011110	^	*Caret – circumflex*
95	01011111	_	*Underscore*
96	01100000	`	*Grave accent*
97	01100001	a	*Lowercase a*
98	01100010	b	*Lowercase b*
99	01100011	c	*Lowercase c*
100	01100100	d	*Lowercase d*
101	01100101	e	*Lowercase e*
102	01100110	f	*Lowercase f*
103	01100111	g	*Lowercase g*
104	01101000	h	*Lowercase h*
105	01101001	i	*Lowercase i*
106	01101010	j	*Lowercase j*

107	01101011	k	*Lowercase k*
108	01101100	l	*Lowercase l*
109	01101101	m	*Lowercase m*
110	01101110	n	*Lowercase n*
111	01101111	o	*Lowercase o*
112	01110000	p	*Lowercase p*
113	01110001	q	*Lowercase q*
114	01110010	r	*Lowercase r*
115	01110011	s	*Lowercase s*
116	01110100	t	*Lowercase t*
117	01110101	u	*Lowercase u*
118	01110110	v	*Lowercase v*
119	01110111	w	*Lowercase w*
120	01111000	x	*Lowercase x*
121	01111001	y	*Lowercase y*
122	01111010	z	*Lowercase z*
123	01111011	{	*Opening brace*
124	01111100	|	*Vertical bar*
125	01111101	}	*Closing brace*
126	01111110	~	*Equivalency sign – tilde*
127	01111111		*Delete*

128	10000000	€	*Euro sign*
129	10000001		
130	10000010	‚	*Single low-9 quotation mark*
131	10000011	ƒ	*Latin small letter f with hook*
132	10000100	„	*Double low-9 quotation mark*
133	10000101	…	*Horizontal ellipsis*
134	10000110	†	*Dagger*
135	10000111	‡	*Double dagger*
136	10001000	ˆ	*Modifier letter circumflex accent*
137	10001001	‰	*Per mille sign*
138	10001010	Š	*Latin capital letter S with caron*
139	10001011	‹	*Single left-pointing angle quotation*
140	10001100	Œ	*Latin capital ligature OE*
141	10001101		
142	10001110	Ž	*Latin capital letter Z with caron*
143	10001111		
144	10010000		
145	10010001	‘	*Left single quotation mark*
146	10010010	’	*Right single quotation mark*
147	10010011	“	*Left double quotation mark*
148	10010100	”	*Right double quotation mark*

149	10010101	•	*Bullet*
150	10010110	–	*En dash*
151	10010111	—	*Em dash*
152	10011000	˜	*Small tilde*
153	10011001	™	*Trade mark sign*
154	10011010	Š	*Latin small letter S with caron*
155	10011011	›	*Single right-pointing angle quotation mark*
156	10011100	Œ	*Latin small ligature oe*
157	10011101		
158	10011110	Ž	*Latin small letter z with caron*
159	10011111	Ÿ	*Latin capital letter Y with diaeresis*
160	10100000		*Non-breaking space*
161	10100001	¡	*Inverted exclamation mark*
162	10100010	¢	*Cent sign*
163	10100011	£	*Pound sign*
164	10100100	¤	*Currency sign*
165	10100101	¥	*Yen sign*
166	10100110	¦	*Pipe, Broken vertical bar*
167	10100111	§	*Section sign*
168	10101000	¨	*Spacing diaeresis – umlaut*
169	10101001	©	*Copyright sign*

170	10101010	ª	*Feminine ordinal indicator*
171	10101011	«	*Left double angle quotes*
172	10101100	¬	*Not sign*
173	10101101		*Soft hyphen*
174	10101110	®	*Registered trade mark sign*
175	10101111	¯	*Spacing macron – overline*
176	10110000	°	*Degree sign*
177	10110001	±	*Plus-or-minus sign*
178	10110010	²	*Superscript two – squared*
179	10110011	³	*Superscript three – cubed*
180	10110100	´	*Acute accent - spacing acute*
181	10110101	µ	*Micro sign*
182	10110110	¶	*Pilcrow sign - paragraph sign*
183	10110111	·	*Middle dot - Georgian comma*
184	10111000	¸	*Spacing cedilla*
185	10111001	¹	*Superscript one*
186	10111010	º	*Masculine ordinal indicator*
187	10111011	»	*Right double angle quotes*
188	10111100	¼	*Fraction one quarter*
189	10111101	½	*Fraction one half*
190	10111110	¾	*Fraction three quarters*

191	10111111	¿	*Inverted question mark*
192	11000000	À	*Latin capital letter A with grave*
193	11000001	Á	*Latin capital letter A with acute*
194	11000010	Â	*Latin capital letter A with circumflex*
195	11000011	Ã	*Latin capital letter A with tilde*
196	11000100	Ä	*Latin capital letter A with diaeresis*
197	11000101	Å	*Latin capital letter A with ring above*
198	11000110	Æ	*Latin capital letter AE*
199	11000111	Ç	*Latin capital letter C with cedilla*
200	11001000	È	*Latin capital letter E with grave*
201	11001001	É	*Latin capital letter E with acute*
202	11001010	Ê	*Latin capital letter E with circumflex*
203	11001011	Ë	*Latin capital letter E with diaeresis*
204	11001100	Ì	*Latin capital letter I with grave*
205	11001101	Í	*Latin capital letter I with acute*
206	11001110	Î	*Latin capital letter I with circumflex*
207	11001111	Ï	*Latin capital letter I with diaeresis*
208	11010000	Ð	*Latin capital letter ETH*
209	11010001	Ñ	*Latin capital letter N with tilde*
210	11010010	Ò	*Latin capital letter O with grave*
211	11010011	Ó	*Latin capital letter O with acute*

212	11010100	Ô	*Latin capital letter O with circumflex*
213	11010101	Õ	*Latin capital letter O with tilde*
214	11010110	Ö	*Latin capital letter O with diaeresis*
215	11010111	×	*Multiplication sign*
216	11011000	Ø	*Latin capital letter O with slash*
217	11011001	Ù	*Latin capital letter U with grave*
218	11011010	Ú	*Latin capital letter U with acute*
219	11011011	Û	*Latin capital letter U with circumflex*
220	11011100	Ü	*Latin capital letter U with diaeresis*
221	11011101	Ý	*Latin capital letter Y with acute*
222	11011110	Þ	*Latin capital letter THORN*
223	11011111	ß	*Latin small letter sharp s - ess-zed*
224	11100000	à	*Latin small letter a with grave*
225	11100001	á	*Latin small letter a with acute*
226	11100010	â	*Latin small letter a with circumflex*
227	11100011	ã	*Latin small letter a with tilde*
228	11100100	ä	*Latin small letter a with diaeresis*
229	11100101	å	*Latin small letter a with ring above*
230	11100110	æ	*Latin small letter ae*
231	11100111	ç	*Latin small letter c with cedilla*
232	11101000	è	*Latin small letter e with grave*

233	11101001	É	*Latin small letter e with acute*
234	11101010	Ê	*Latin small letter e with circumflex*
235	11101011	Ë	*Latin small letter e with diaeresis*
236	11101100	Ì	*Latin small letter i with grave*
237	11101101	Í	*Latin small letter i with acute*
238	11101110	Î	*Latin small letter i with circumflex*
239	11101111	Ï	*Latin small letter i with diaeresis*
240	11110000	Ð	*Latin small letter eth*
241	11110001	Ñ	*Latin small letter n with tilde*
242	11110010	Ò	*Latin small letter o with grave*
243	11110011	Ó	*Latin small letter o with acute*
244	11110100	Ô	*Latin small letter o with circumflex*
245	11110101	Õ	*Latin small letter o with tilde*
246	11110110	Ö	*Latin small letter o with diaeresis*
247	11110111	÷	*Division sign*
248	11111000	Ø	*Latin small letter o with slash*
249	11111001	Ù	*Latin small letter u with grave*
250	11111010	Ú	*Latin small letter u with acute*
251	11111011	Û	*Latin small letter u with circumflex*
252	11111100	Ü	*Latin small letter u with diaeresis*
253	11111101	Ý	*Latin small letter y with acute*

254	11111110	Þ	*Latin small letter thorn*
255	11111111	Ÿ	*Latin small letter y with diaeresis*

Langkah kedua membuat sebuah tabel yang bertujuan mengkonversi teks, dalam mengenkripsi teks memiliki aturan hitungan dimana *plaintext* akan ditambah dengan kunci yang mana hasil tambah akan menjadi *chipertext* yang sulit dibaca atau diartikan pihak ketiga seperti tabel dibawah ini :

Tabel 4.2 Hitungan Enkripsi

Plaintext	u	n	p	a	b
	117	**110**	**112**	**97**	**98**
Kunci	m	e	d	a	n
	109	**101**	**100**	**97**	**110**
Chipertext	â	Ó	Ô	Å	Đ
	226	**211**	**212**	**194**	**208**

.

Tabel 4.3 Hitungan Dekripsi

Plaintext	u	n	p	a	b
	117	**110**	**112**	**97**	**98**
Kunci	m	e	d	a	n
	109	**101**	**100**	**97**	**110**
Chipertext	Â	Ó	Ô	Å	Đ

	226	**211**	**212**	**194**	**208**
Dekripsi	u	n	p	a	b

Tabel 4.4 Hitungan Permutasi

Permutasi	e	m	a	d	n

Dalam algoritma *vigenere chiper* permutasi dilakukan dengan cara tukar posisi :

XY= Bilangan acak 0sp (jtp – 1)

contoh kunci = medan dengan jumlah tukar posisi 5

Tukar kunci = emadn

x = 2 y =1 medan > emadn

dimana x adalah letak bilangan acak tersebut dan y adalah berapa kali kunci di acak, jadi dari hitungan tersebut dapat disimpulkan bahwa kunci medan hanya di acak sebanyak satu kali dari letak bilangan ke dua.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari pembuatan model simulasi keamanan pesan teks menggunakan kunci pada algoritma vigenere cihper yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

a. Perangkat lunak ini dirancang agar pengguna mengetahui keamanan pesan teks menggunakan teknik permutasi pada algoritma vigenere chiper.

b. Pada aplikasi tersebut permutasi dibuat dengan cara tukar posisi.

c. Dalam aplikasi ini hanya mempunyai satu kunci.

5.2 Saran

Adapun saran yang ingin disampaikan adalah :

a. Untuk mempermudah dan memperlancar penggunaan program tersebut diharapkan adanya perangkat-perangkat yang mampu menunjang program ini seperti perangkat atau *device* yang memadai seperti komputer, laptop, dan lain-lain.

b. Perangkat lunak ini dapat dikembangkan sehingga dapat dijalankan di lebih dari satu computer.

DAFTAR PUSTAKA

Hartono. 2007. *Perancangan Aplikasi Kriptography Advanced Encryption Standard*. Jogiyanto. 1990. *Analisis dan Disain Sistem Informasi.* Yogyakarta: Andi Offset.

Munir, Rinaldi. 2006. Diktat Kuliah IF5054 kriptografi. Sekolah Teknik Elektro dan Informatika Intsititut Teknologi Bandung.

Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." Int. J. Recent Trends Eng. Res 2.12 (2016): 140-151.

Andrian, Yudhi, and Purwa Hasan Putra. "Analisis Penambahan Momentum Pada Proses Prediksi Curah Hujan Kota Medan Menggunakan Metode Backpropagation Neural Network." Seminar Nasional Informatika (SNIf). Vol. 1. No. 1. 2017.

Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." Seminar Nasional Teknologi Informasi Dan Multimedia, ISSN. 2015.

INDRA PERMANA, A. M. I. N. U. D. D. I. N. "SISTEM PAKAR MENDETEKSI HAMA DAN PENYAKIT TANAMAN KELAPA SAWIT PADA PT. MOEIS KEBUN SIPARE-PARE KABUPATEN BATUBARA." (2013).

Wahyuni, Sri. "Implementasi Rapidminer Dalam Menganalisa Data Mahasiswa Drop Out." Jurnal Abdi Ilmu 10.2 (2018): 1899-1902.

Arjuna, Putu H.dkk. 2012. Implementasi Enkripsi Data Dengan Algoritma Vigenere Chiper. Yogyakarta: Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012)

Pabokory, Fresly Nandar dkk 2015. Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard Vol: 10 No. 1 Februari 2015.

Yulianingsi Pricilia. Dkk. 2014. Aplikasi Chatting Rahasia Menggunakan Algoritma Vigenere Chiper. Vol 9 No. 1 Februari 2014.

Azlin. Fitriah Musadar. 2018. Aplikasi Kriptografi Keamanan Data Menggunakan Algoritma BASE64. Vol.10 No.2 Desember 2018

Arius D. 2008. Pengatar Ilmu Kriptografi Teori Analisis dan Implementas. Yogyakarta: Penerbit ANDI.

Barus, S., Sitorus, V. M., Napitupulu, D., Mesran, M., & Supiyandi, S. (2018). Sistem Pendukung Keputusan Pengangkatan Guru Tetap Menerapkan Metode Weight Aggregated Sum Product Assesment (WASPAS). Jurnal Media Informatika Budidarma, 2(2).

Batubara, Supina. "Analisis perbandingan metode fuzzy mamdani dan fuzzy sugeno untuk penentuan kualitas cor beton instan." IT Journal Research and Development 2.1 (2017): 1-11.

Batubara, Supina, Sri Wahyuni, and Eko Hariyanto. "Penerapan Metode Certainty Factor Pada Sistem Pakar Diagnosa Penyakit Dalam." Seminar Nasional Royal (SENAR). Vol. 1. No. 1. 2018.

Fachri, B. (2018). Perancangan Sistem Informasi Iklan Produk Halal Mui Berbasis Mobile Web Menggunakan Multimedia Interaktif. Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika), 3, 98-102.

Fachri, B. (2018, September). APLIKASI PERBAIKAN CITRA EFEK NOISE SALT & PAPPER MENGGUNAKAN METODE CONTRAHARMONIC MEAN FILTER. In Seminar Nasional Royal (SENAR) (Vol. 1, No. 1, pp. 87-92).

Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. Int. J. Secur. Its Appl, 10(8), 173-180.

Ginting, G., Fadlina, M., Siahaan, A. P. U., & Rahim, R. (2017). Technical approach of TOPSIS in decision making. Int. J. Recent Trends Eng. Res, 3(8), 58-64.

Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. Jurnal Teknik dan Informatika, 5(2), 13-19.

Aryza, S., Irwanto, M., Lubis, Z., Siahaan, A. P. U., Rahim, R., & Furqan, M. (2018). A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive. In IOP Conference Series: Materials Science and Engineering (Vol. 300, No. 1, p. 012067). IOP Publishing.

Tasril, V. (2018). Sistem Pendukung Keputusan Pemilihan Penerimaan Beasiswa Berprestasi Menggunakan Metode Elimination Et Choix Traduisant La Realite. INTECOMS: Journal of Information Technology and Computer Science, 1(1), 100-109.