KRIPTOGRAFI GABUNGAN MENGGUNAKAN ALGORITMA MONO ALPHABETIC DAN ONE TIME PAD

Sugianto, Teguh Yuniarto

Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak
Jalan Merdeka Barat No. 372 Pontiana, Kalimantan Barat
Telpn (0561) 735555, Fax (0561) 737777
[1]Sugi_xiang@yahoo.com, [2]guh_89@yahoo.com

Abstrak

Sistem keamanan yang lebih kompleks diperlukan dengan kesanggupan untuk mengikuti perkembangan yang ada agar dapat melindungi sistem dari berbagai ancaman yang mungkin timbul. Salah satu upaya pengamanan data digital yang dapat dilakukan adalah dengan Kriptografi. Teknik kriptografi klasik seperti Substitusi Monoalphabetic jarang digunakan lagi karena tidak dapat menyaingi kompleksitas teknik kriptografi yang lain oleh karena kesederhanaannya. Oleh karena itu timbul suatu gagasan untuk menggunakan kembali algoritma kriptografi klasik seperti Substitusi Monoalphabetic dengan cara menggabungkannya dengan algoritma modern. Dengan digunakannya algoritma gabungan antara Vernam/One Time Pad (OTP) dan Substitusi Monoalphabetic diharapkan akan menghasilkan algoritma yang lebih tangguh tanpa menghilangkan sifat asli dari Substitusi Monoalphabetic. Implementasi dari penggabungan dua algoritma tersebut adalah sebuah sistem pengamanan yang menganut kerahasiaan pada kunci (key). Setiap proses enkripsi akan menghasilkan kunci yang berbeda sehingga kunci yang digunakan dalam dekripsi hanya akan digunakan satu kali sehingga diharapkan mampu memberikan keamanan yang maksimal.
Kata Kunci : *Kriptografi, Monoalphabetic, Vernam , Enkripsi, Dekripsi*

Abstract

More complex security system is needed , with the ability to follow the changes in order to protect the system from various threats that may arise. One safeguard digital data to do is to Cryptography. Classical Cryptography techniques such as substitution Monoalphabetic rarely used anymore because it can not compete with other complexity cryptographic techniques because of its simplicity. Hence arose the idea to reuse classical cryptography algorithms such as substitution Monoalphabetic by means combines it with modern algorithms. With the use of a combined algorithm between Vernam / One Time Pad (OTP) and substitution Monoalphabetic expected to produce a more robust algorithm without losing the original nature of the substitution Monoalphabetic. Implementation of the merger of these two algorithms is a security system that adheres to the confidentiality of the key (key). Each encryption process will result in a different key so that the key used in the decryption will only be used one time so it is expected to provide maximum security.
Keywords: *Cryptography, Monoalphabetic, Vernam, Encryption, Decryption.*

1. PENDAHULUAN

Suatu instansi atau organisasi sangat membutuhkan keamanan infrastruktur teknologi informasi yang baik untuk melindungi aset-asetnya terutama data yang penting dan sensitif. Sehingga saat ini dibutuhkan sistem keamanan yang lebih kompleks dengan kesanggupan untuk mengikuti perkembangan yang ada agar dapat melindungi sistem dari berbagai ancaman yang

mungkin timbul. Komunikasi yang aman dimaksudkan untuk melindungi data ataupun informasi ketika dikirimkan atau ditransmisikan kepada pihak lain, sehingga data atau informasi yang ditransmisikan itu tidak dapat disadap dimanipulasi ataupun dirusak oleh pihak-pihak yang tidak bertanggung jawab.

Salah satu cara untuk mengamankan komunikasi adalah dengan menerapkan teknik penyandian. Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan (data atau informasi) dengan cara menyamarkannya kedalam bentuk sandi yang tidak mempunyai makna. Pesan yang dirahasiakan disebut plainteks *(plaintext)* dan hasil penyamaran disebut cipherteks *(chipertext)*. Proses dari plainteks ke cipherteks disebut enkripsi *(encryption)* dan proses pembalikan dari cipherteks menjadi plainteks kembali disebut dekripsi *(decryption)*.

Ada beberapa algoritma enkripsi yang biasa digunakan seperti *DES, Triple DES, Blowfish, IDEA* dan sebagainya. Algoritma-algoritma tersebut begitu rumit dan sulit dimengerti dengan dalih 'faktor keamanan', menurut pendapat umum menyatakan bahwa semakin sulit suatu algoritma dimengerti, maka semakin aman. Namun bagi para pengguna tidak memikirkan seberapa sulit algoritma dan aplikasinya, yang pengguna inginkan adalah menjaga kerahasiaan data.

Jika algoritma yang digunakan semakin sederhana semakin baik, karena semakin sederhana suatu algoritma, maka akan semakin sedikit proses komputasinya dan semakin sedikit waktu yang dibutuhkan untuk mengeksekusinya. Kesederhanaan itulah yang ditawarkan oleh algoritma *Vernam/One Time Pad (OTP)*, algoritma kriptografi yang secara teori dan praktik aman dari tangan-tangan penyadap, dan dikenal dengan sebutan *"unbreakable chiper"*.

Pemanfaatan *monoalphabetic cipher* sebagai sandi substitusi untuk memperkuat sandi subtitusi *one time pad cipher* diharapkan dapat mengatasi permasalahan terhadap kelemahan sandi subtitusi klasik. Oleh karena itu, diperlukan perangkat lunak sebagai pendukung untuk membantu menjaga kerahasiaan sebuah data.

Metode enkripsi yang akan dibangun di sini menerapkan teknik pada kriptografi modern, yang menganut kerahasiaan pada kunci *(key)*. Pada penulisan skripsi ini, penyandian data dibuat dengan menggunakan algoritma *Vernam/OTP* untuk sistem pengamanan pesan pada bahasa pemprograman *Visual Basic*. Apabila memasukkan sebuah data informasi, maka system akan mengubah data tersebut melalui teknik penyandian menggunakan algoritma *one-time pad/vernam*, maka keluaran dari sistem adalah sebuah data yang sudah terenkripsi dengan algoritma tersebut sehingga data tidak mudah diketahui informasi yang terdapat didalamnya oleh orang-orang yang tidak bertanggungjawab.

Penelitian serupa yang dilakukan oleh Fairuzabadi (2010), membahas tentang algoritma gabungan antara *Secure Hash Algorithm* dan *Substitusi Mono Alphabet* yang mana pada teknik transposisi huruf-huruf pada *plaintext* dan *chipertext* tetap sama, tetapi urutannya diubah. Disebut juga metode *permutasi*, karena *transpose* setiap karakter di dalam teks sama dengan mempermutasikan karakter-karakter tersebut. Penelitian lainnya oleh Munir (2011), membahas mengenai algoritma untuk mengenkripsi citra digital dengan memanfaatkan sistem chaos. Enkripsi dengan algoritma pseudo one-time pad, yang dalam hal ini barisan kunci dibangkitkan dengan sistem chaos yang mana membangkitkan bilangan-bilangan pseudo-random. Hasilnya memperlihatkan algoritma dapat mengenkripsi sembarang citra (citra grayscale atau citra berwarna) dengan baik sehingga citra terenkripsi terlihat acak sempurna.

Berdasarkan uraian tersebut, maka penulis membangun sebuah aplikasi *kriptografi hybrid* dengan menggabungkan dua algoritma, yaitu *monoalphabetic cipher* dan *one time pad cipher*, sehingga dapat dimanfaatkan sebagai alat bantu dalam menjaga kerahasiaan sebuah data berlapis serta dapat mempelajari dan memahami cara kerja dari kedua metode kriptografi tersebut.

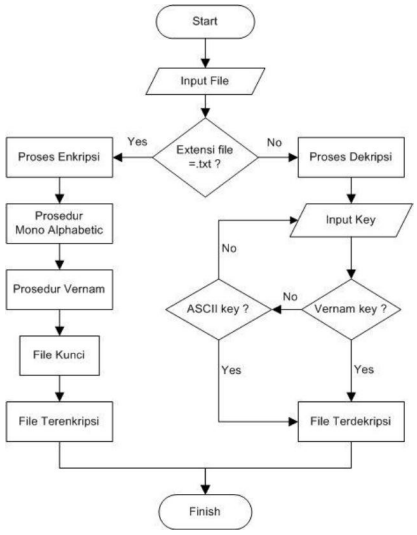
2. METODE PENELITIAN

Bentuk penelitian yang dilakukan oleh penulis dalam penelitian ini adalah tinjauan literatur (*literatur review*). Tinjauan literatur merupakan suatu kerangka, konsep, atau orientasi untuk melakukan analisis dan klasifikasi fakta yang dikumpulkan dalam penelitian yang dilakukan. Sumber-sumber rujukan dari buku maupun jurnal, yang diacu dalam penelitian ini adalah yang berhubungan langsung dengan objek yang diteliti, yaitu enkripsi plaintext. Metode penelitian yang digunakan adalah metode *research and development*, adalah suatu penelitian dimana perangkat lunak yang telah dibuat diujicobakan dan dilihat tingkat keefektifannya.

Metode yang digunakan peneliti dalam melakukan pengumpulan data adalah studi dokumentasi, yaitu peneliti mengumpulkan serta mempelajari bahan-bahan tertulis yang berhubungan dengan penggunaan Algoritma Kriptografi gabungan monoalphabetic cipher dan one time pad/Vernam cipher yang didapat melalui artikel, buku, e-book dan pencarian diinternet terhadap materi metode Kriptografi monoalphabetic cipher dan one time pad/Vernam cipher.

3. HASIL DAN PEMBAHASAN

Perancangan sistem dengan prinsip kerjanya dan flowchart aplikasi yang dibuat menggunakan metode gabungan monoalphabetic cipher dan one time pad/Vernam cipher sebagai berikut :



Gambar 1. Flowchart Aplikasi

Penulis mulai merancang form dan modul yang sudah didefinisikan sebelumnya sehingga membentuk aplikasi yang utuh. Prinsip enkripsi pada algoritma ini adalah dengan mengkombinasikan masing-masing karakter pada plainteks dengan satu karakter pada kunci. Oleh karena itu, panjang kunci setidaknya harus sama dengan panjang plainteks. Enkripsi dapat dinyatakan sebagai penjumlahan modulo 26 dari satu karakter plainteks dengan satu karakter kunci OTP:

$ci = (pi + ki) \bmod 26$

Dalam hal ini, pi adalah plainteks ke-i, dan ci adalah huruf cipherteks ke-i. Panjang kunci sama dengan panjang plainteks, sehingga tidak ada kebutuhan mengulang penggunaan kunci selama proses enkripsi. Setelah pengirim mengenkripsikan pesan dengan kunci, ia menghancurkan kunci tersebut. Penerimaan pesan menggunakan kunci yang sama untuk

mendekripsikan karakter-karakter cipherteks menjadi karakter-karakter plainteks dengan persamaan:

pi = (ci + ki) mod 26

Angka 26 muncul karena sistemnya menggunakan abjad. Artinya hanya abjad A – Z saja yang dapat dikodekan dengan sistem seperti ini. Bila diinginkan pengkodean sembarang data, baik teks, gambar, suara maupun video, maka OTP ini diperluas dengan penggunaan sistem bilangan biner. Semua tipe data dapat dianggap sebagai data biner. Dan karena bilangan biner hanya mengenal 0 dan 1, maka basis 26 diubah menjadi basis 2. Penjumlahan modulo 2 ini dinyatakan dengan XOR. Dan inilah yang sering digunakan dalam sistem digital sekarang ini. Cipherteks diperoleh dengan melakukan penjumlahan modulo 2 satu bit plainteks dengan satu bit kunci:

ci = (pi + ki) mod 2

Dalam hal ini, pi : bit plainteks, ki : bit kunci, ci : bit cipherteks. Plainteks diperoleh dengan melakukan penjumlahan modulo 2 satu bit chiperteks dengan satu bit kunci:

pi = (ci + ki) mod 2

Mengingat operasi penjumlahan modulo 2 identik dengan operasi bit dengan operator XOR, maka persamaan enkripsi dapat ditulis sebagai:

ci = pi ki

dan proses dekripsi menggunakan persamaan:

pi = ci ki

Pada proses chipering, bit hanya mempunyai dua buah nilai, sehingga proses enkripsi hanya menyebabkan dua keadaan pada bit tersebut, berubah atau tidak berubah. Dua keadaan tersebut ditentukan oleh kunci enkripsi yang disebut aliran kunci (keystream). Aliran kunci dibangkitkan dari sebuah pembangkit yang dinamakan pembangkit aliran kunci (keystream generator). Aliran kunci di-XOR-kan dengan aliran bit- bit plainteks p1, p2, ..., pi, untuk menghasilkan aliran bit-bit chiperteks:

ci = pi ki

Disisi penerima, bit-bit chiperteks di-XOR-kan dengan aliran kunci yang sama untuk menghasilkan bit-bit plainteks:

pi = ci ki

karena proses enkripsi dua kali berturut-turut menghasilkan kembali plainteks semula.

ci ki = (pi ki) ki = pi (ki ki) = pi 0 = pi

Pembangkit aliran kunci menghasilkan elemen bit kunci ki yang kemudian di-XOR-kan dengan bit plainteks menghasilkan bit chiperteks ci. Di sisi penerima, pembangkit yang sama digunakan untuk menghasilkan aliran kunci yang sama untuk selanjutnya di-XOR-kan dengan chiperteks ci dan memberikan kembali plainteks pi semula.

Seperti kita ketahui bahwa untuk merancang unbreakable cipher, ada dua syarat yang harus dipenuhi, yaitu kunci harus dipilih secara acak (setiap kunci harus mempunyai peluang yang sama untuk terpilih) dan panjang kunci harus sama dengan panjang plainteks yang akan dienkripsikan. Kedua syarat tersebut dapat menyebabkan plainteks sama belum tentu dienkripsi menjadi cipherteks yang sama. Dengan kata lain kriptanalisis akan mendapatkan hasil bahwa sebuah cipherteks yang didekripsikannya mungkin menghasilkan beberapa plainteks bermakna.

Hasil ini akan membingungkannya dalam menentukan plainteks mana yang benar. Sebagai contoh, dipunyai sebuah plainteks yaitu RUSDI dan memiliki sebuah kunci yaitu

CRASH. Perlu diingat bahwa panjang kunci harus sama dengan plainteks dan sebaiknya tidak ada karakter yang diulang. Pertama kita harus mendapatkan kode ASCII dari plainteks kemudian diubah ke bentuk biner.

Tabel 1. Kode ASCII dan notasi biner plainteks

Karakter	ASCII	Notasi biner
R	82	0101 0010
U	85	0101 0101
S	83	0101 0011
T	68	0100 0100
I	73	0100 1001

Hal yang sama juga harus dilakukan pada kunci yang dipilih.

Tabel 2. Kode ASCII dan notasi biner kunci

Karakter	ASCII	Notasi biner
C	67	0100 0011
R	82	0101 0010
A	65	0100 0001
S	83	0101 0111
H	72	0100 1000

Setelah itu masing-masing karakter di XOR-kan dengan kunci.

Tabel 3. Hasil proses XOR plainteks dan kunci

Cipherteks		
0101 0010	=	Ctrl-Q
0101 0101	=	Ctrl-G
0101 0011	=	Ctrl-R
0100 0100	=	Ctrl-W
0100 1001	=	Ctrl-A

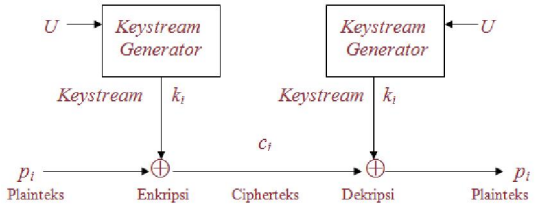
Proses dekripsi pesan juga melakukan operasi yang sama yaitu XOR antara Cipher dengan kunci.

Tabel 4. Kode ASCII dan notasi biner dekripsi data

Plainteks	Kunci	Cipherteks
Ctrl-Q = 0001 0001	C = 0100 0011	R = 0101 0010
Ctrl-Q = 0000 0111	R = 0101 0010	U = 0101 0101
Ctrl-Q = 0001 0010	A = 0100 0001	S = 0101 0011
Ctrl-Q = 0001 0111	S = 0101 0100	D = 0100 0100
Ctrl-Q = 0000 0001	H = 0100 1000	I = 0100 1001

Plainteks yang digunakan hanya mampu untuk men-enkripsi sebanyak 192 karakter / byte untuk menghindari waktu proses komputasi yang terlalu lama apabila banyak karakter yang dienkripsi, oleh sebab itu program ini di rancang untuk men-enkripsi plainteks dengan panjang 192 karakter / byte.

Pembangkit aliran bit kunci diimplementasikan sebagai prosedur algoritmik, sehingga bit-bit kunci (keystream) dapat dibangkitkan secara simultan oleh pengirim dan penerima pesan. Prosedur algoritmik tersebut menerima masukan sebuah kunci U. Keluaran dari prosedur merupakan fungsi dari U (Gambar 4.2). Pembangkit harus menghasilkan bit-bit kunci yang kuat secara kriptografi.



Gambar 2. Enchipering dengan pembangkit aliran kunci yang bergantung pada kunci U

Karena pengirim dan penerima harus menghasilkan bit-bit kunci yang sama, maka keduanya harus memiliki kunci U yang sama. Kunci U ini harus dijaga kerahasiaannya. Algoritma OTP menggunakan kunci U yang relatif pendek untuk membangkitkan bit-bit kunci yang panjang.

Pada gambar 4.3, aliran kunci dihasilkan dengan fungsi acak (random function) yang diulang sebanyak plainteks dan fungsi luaran (output function) yang menghasilkan bit-bit aliran kunci

Sampai saat ini, ada beberapa metode yang dikenal dapat digunakan untuk membangkitkan bilangan acak pada OTP seperti LCG (Linear Congruental Generators), LFSR (Linear Feedback Shift Register), Generator Geffe (yang menggunakan 3 atau lebih LFSR), dan metode- metode yang lain. Metode pembangkitan bilangan acak yang cukup familiar adalah LFSR, sebab LFSR digunakan pada kriptografi dan teori pengkodean serta telah digunakan militer ketika dimulainya penggunaan alat elektronik.

Untuk penjelasan mengenai metode ini kita gunakan angka 0-25 untuk mensubtitusikan huruf A-Z. Semua tanda baca dan spasi diabaikan.

```
a  b  c  d  e  f  g  h  i  j  k  l  m
00 01 02 03 04 05 06 07 08 09 10 11 12
n  o  p  q  r  s  t  u  v  w  x  y  z
13 14 15 16 17 18 19 20 21 22 23 24 25
```

Untuk setiap huruf ke-i plainteks Pi yang diberikan memiliki subtitusi huruf chiperteks ke-i Ci dengan rumus sebagai berikut :

Ci = (Pi + ki) mod 26 …(1)

Dimana Ki adalah huruf ke-i pada kunci. Akan tetapi Sebagai contoh, terdapat plainteks yang berisi “A quick brown fox jumps” dan kunci yang digunakan adalah “my key sequence”. Di bawah ditunjukan perulangan kunci yang dituliskan di bawah huruf plainteks dengan urutan yang sama. Di bawah plainteks dan kunci dituliskan angka-angka yang berkorespondensi dengan setiap hurufnya.

```
A  Q  U  I  C  K  B  R  O  W  N  F  O  X  J  U  M  P  S
00 16 20 08 02 10 01 17 14 22 13 05 14 23 09 20 12 15 18
K  E  Y  K  E  Y  K  E  Y  K  E  Y  K  E  Y  K  E  Y  K
10 04 24 10 04 24 10 04 24 10 04 24 10 04 24 10 04 24 10
-------------------------------------------------------
K  U  S  S  G  I  L  V  M  G  R  D  Y  B  H  E  Q  N  C
10 20 18 18 06 08 11 21 12 06 17 03 24 01 07 04 16 13 02
```

Sekumpulan huruf di bawah garis horizontal adalah chiperteks. Sebagai contoh, huruf ketiga plainteks U (=20) dan huruf pertama kunci Y (=24) bila dijumlahkan jumlahnya adalah 44 kemudian dimodulo 26 sesuai persamaan (1) sehingga menghasilkan 18 yang berdistribusi dengan huruf S.

Untuk dekripsi chiperteks digunakan rumus sebagai berikut:

Pi = (Ci - ki) mod 26…(2)

Sebagai contoh, huruf terakhir dari chiperteks C(=02), dikurangi huruf K(=10) pada kunci menghasilkan nilai-8 dengan menggunakan mod 26 sesuai persamaan (2), kita tambahkan 26 sehingga hasil yang diperoleh adalah 18 yaitu huruf S. Selain cara di atas, proses enkripsi dan dekripsi dapat dilakukan dengan menggunakan bujursangkar Vigènere. Bujursangkar Vigènere sangat membantu proses chipering yang dilakukan secara manual karena setiap huruf plainteks dan kunci sudah dipetakan menjadi sebuah huruf chiperteks. Sehingga tidak diperlukan perhitungan matematis dalam chipering secara manual.

Pengujian sistem perangkat lunak kriptografi menggunakan metode gabungan monoalphabetic cipher dan one time pad/Vernam cipher dirancang dengan melakukan pengujian terhadap aplikasi untuk diuji apakah proses yang diharapkan dapat berjalan atau tidak.

Berikut ini merupakan proses penyandian pesan dengan tahapan sebagai berikut :

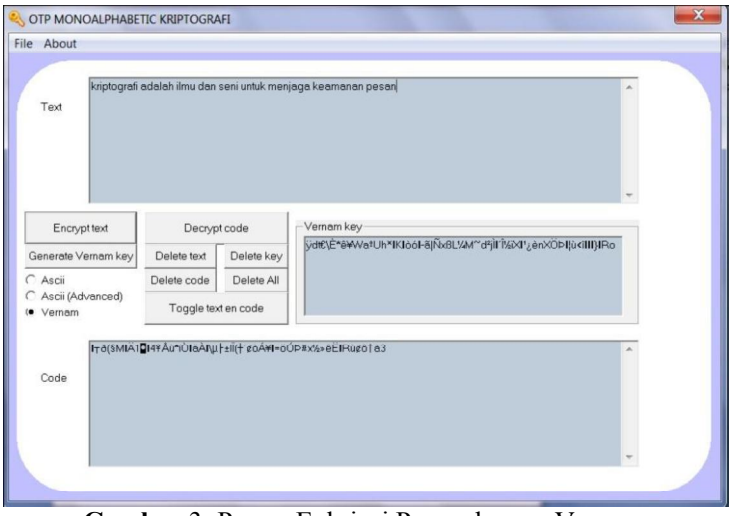
a. Masukan teks yang akan disandikan kedalam kotak plainteks
b. pilih ”vernam” pada pilihan option pada kanan bawah
c. klik ”generate key” agar mendapatkan hasil kunci acak
d. klik ”encrypt teks” untuk mendapatkan hasil enkripsi pada kotak cipherteks

Sebagai contoh diatas :

Plainteks : kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan

Key : ÿdt€\È*ê¥WaºUh¤•K…òóˆ-ã|ÑxßL¼M~d²jÌ†´Î½ïX˜¹¿ènXÖÞ‹¦ù<î—„„}“Ro

Chiperteks : ˆ¢ö‡°É…\¥¥åqš"Ö ÝYã–ªÉ• ”, ä=8á¶£<-ôÇµY¦Uó-ô• ›-AªÃ-‡ÔŒŠD• MÅë>µ• ?ù• ,±DóüDæáSØ‘Ÿµ KÈ$?Í_áRñ°Mø-˜¹[÷Jëí¡ç3÷˜Rß€



Gambar 3. Proses Enkripsi Pesan dengan Vernam

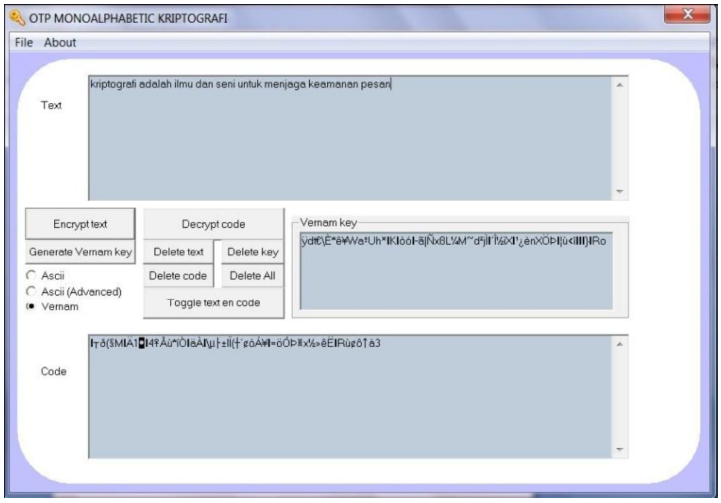
Pada proses dekripsi denga Vernam juga dilakukan dengan hal yang sama, tetapi teks file yang telah dienkripsi dimasukkan ke dalam kotak cipherteks kemudian masukkan kunci lalu tekan tombol ”Decrypt code” pada program. Lalu pada kotak plainteks akan muncul teks file yang telah di dekripsikan.

Chiperteks: ˆ¢ö‡°É…\¥¥åqš"Ö ÝYã–ªÉ• ”, ä=8á¶£<-ôÇµY¦Uó-ô• ›-AªÃ–‡ÔŒŠD• MÅë>µ• ?ù• ,±DóüDæáSØ‘Ÿµ KÈ$?Í_áRñ°Mø–˜¹[÷Jëí¡ç3÷˜Rß€

Key :

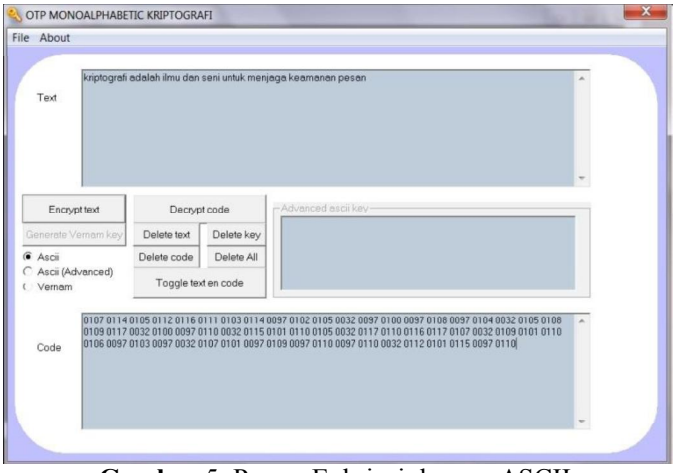
ÿdt€\È*ê¥Wa°Uh¤•K…òóˆ-ã|ÑxßL¼M~d²jÌ†´Î½ïX˜¹¿ènXÖÞ‹¦ù<î—„„}“Ro

Plainteks : kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan



Gambar 4. Proses Dekripsi Pesan dengan Vernam

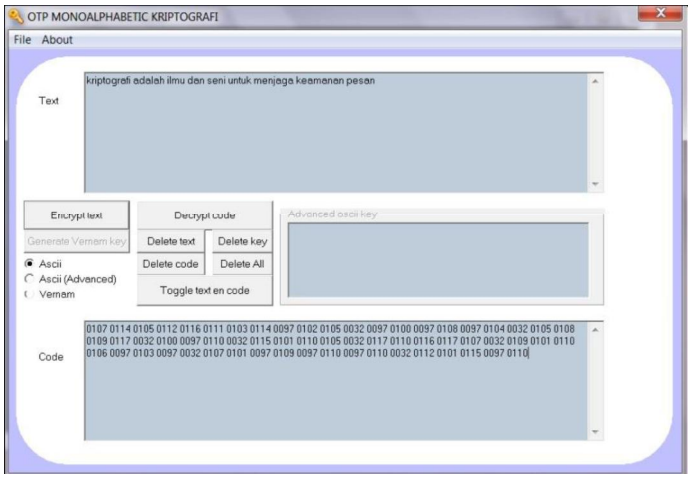
Pengujian Proses Enkripsi dengan ASCII
Plaintext : kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan
Hasil Ciphertext : 0107 0114 0105 0112 0116 0111 0103 0114 0097 0102 0105 0032 0097 0100 0097 0108 0097 0104 0032 0105 0108 0109 0117 0032 0100 0097 0110 0032 0115 0101 0110 0105 0032 0117 0110 0116 0117 0107 0032 0109 0101 0110 0106 0097 0103 0097 0032 0107 0101 0097 0109 0097 0110 0097 0110 0032 0112 0101 0115 0097 0110



Gambar 5. Proses Enkripsi dengan ASCII

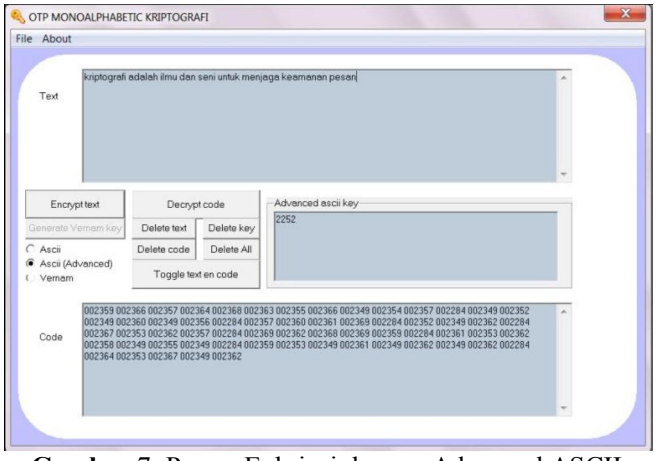
Pengujian Proses Dekripsi dengan ASCII
Ciphertext : 0107 0114 0105 0112 0116 0111 0103 0114 0097 0102 0105 0032 0097 0100 0097 0108 0097 0104 0032 0105 0108 0109 0117 0032 0100 0097 0110 0032 0115 0101 0110 0105 0032 0117 0110 0116 0117 0107 0032 0109 0101 0110 0106 0097 0103 0097 0032 0107 0101 0097 0109 0097 0110 0097 0110 0032 0112 0101 0115 0097 0110

Hasil Plaintext : kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan



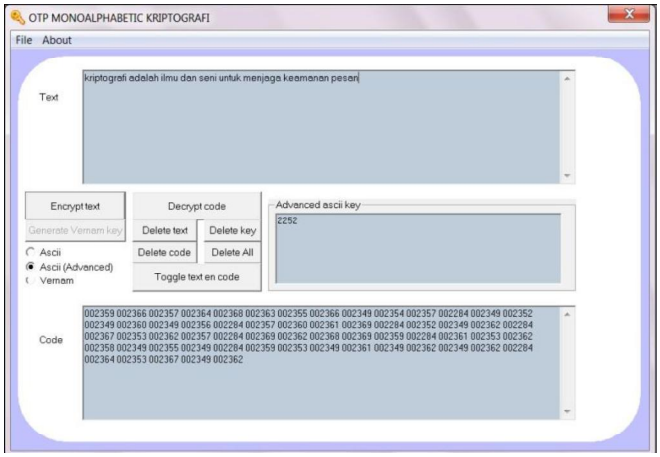
Gambar 6. Proses Dekripsi dengan ASCII

Pengujian Proses Enkripsi dengan Advanced ASCII
Plaintext : kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan
Advanced Key ASCII: 2252
Hasil Ciphertext : 002359 002366 002357 002364 002368 002363 002355 002366
002349 002354 002357 002284 002349 002352 002349 002360 002349 002356
002284 002357 002360 002361 002369 002284 002352 002349 002362 002284
002367 002353 002362 002357 002284 002369 002362 002368 002369 002359
002284 002361 002353 002362 002358 002349 002355 002349 002284 002359
002353 002349 002361 002349 002362 002349 002362 002284 002364 002353
002367 002349 002362



Gambar 7. Proses Enkripsi dengan Advanced ASCII

Pengujian Proses Dekripsi dengan Advanced ASCII
Ciphertext : 002359 002366 002357 002364 002368 002363 002355 002366
002349 002354 002357 002284 002349 002352 002349 002360 002349 002356
002284 002357 002360 002361 002369 002284 002352 002349 002362 002284
002367 002353 002362 002357 002284 002369 002362 002368 002369 002359
002284 002361 002353 002362 002358 002349 002355 002349 002284 002359
002353 002349 002361 002349 002362 002349 002362 002284 002364 002353
002367 002349 002362
Advanced Key ASCII: 2252
Hasil Plaintext : kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan



Gambar 8. Proses Dekripsi dengan Advanced ASCII

4. KESIMPULAN

Berdasarkan hasil perancangan dan pengujian sistem yang dilakukan penulis, dan dari penelitian yang dilakukan oleh penulis seperti yang telah dipaparkan pada bab-bab sebelumnya, maka penulis menarik beberapa kesimpulan yang dapat diambil dari perancangan perangkat lunak kriptografi gabungan menggunakan algoritma *mono alphabetic* dan *one time pad*, yaitu sebagai berikut: a) Prinsip enkripsi pada algoritma gabungan ini adalah dengan mengkombinasikan masing-masing karakter pada plainteks dengan satu karakter pada kunci; b) Teknik pengenkripsian ini relative sederhana, mudah digunakan dan aman dalam menjamin kerahasiaan informasi atau data yang ingin dikirimkan oleh pengirim pesan kepada penerima pesan tanpa dapat diketahui oleh pihak lain; c) Keamanan algoritma pengenkripsian ini sangat bergantung pada kerahasiaan kunci rahasia (secret key) dan pad yang digunakan baik dalam mengenkripsi maupun mendekripsi data atau informasi; d) Kunci yang di-generate secara acak dan hanya dapat dipergunakan sebanyak satu kali saja.

5. SARAN

Mekanisme enkripsi yang digunakan dalam penelitian kali ini memang masih sederhana, akan tetapi diharapkan dapat berguna sebagai langkah awal untuk masuk ke dunia kriptografi, khususnya dalam implementasi pengamanan pesan dengan menggunakan algoritma kombinasi yang lain. Untuk kedepannya, diharapkan penelitian ini dapat dikembangkan, digunakan serta diterapkan pada bidang-bidang kehidupan yang lain yang lebih kompleks.

DAFTAR PUSTAKA

[1]. Ariyus, Dony., 2008, *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi,* Andi, Yogyakarta.
[2]. Budiharto, Widodo., 2005, *Aplikasi Database Oracle 10g dengan VB6/Vb.NET*, PT Elex Media Komputindo, Jakarta.
[3]. Munir, Rinaldi., 2000, *Algoritma dan Pemrograman dalam Bahasa Pascal dan C (Buku Satu)*, Informatika, Bandung.
[4]. Munir, Rinaldi., 2006, *Kriptografi,* PT. Infomedika Bandung.
[5]. Sadikin, Rifki., 2012, *Kriptografi untuk keamanan jaringan,* Andi, Yogyakarta.

[6]. Simarmata, Janner., 2003, *Rekayasa Perangkat Lunak*, Nikodemus WK, Andi, Yogyakarta.
[7]. Wahid, Fathul., 2004, *Dasar – dasar Algoritma dan Pemrograman,* Edisi kesatu, Andi, Yogyakarta.
[8]. Yudi, Retanto., 2011, Pengembangan Aplikasi Simulasi Pemecahan Rail Fence Cipher, Columnar Transposition dan Scytale, Makalah IF3058 Kriptografi – Sem. II Tahun 2010/2011.