ENKRIPSI DEKRIPSI DATA MENGGUNAKAN METODE STREAM DAN VIGENERE CIPHER

F. Wiwiek Nurwiyati[1], Indra Yatini B[2]

[1, 2]Staf Pengajar Jurusan Teknik Informatika, STMIK AKAKOM Yogyakarta
Jalan Raya Janti 143 Karang Jambe Yogyakarta 55198
Email: Indrayatini@akakom.ac.id, Wiwiek@akakom.ac.ad

ABSTRACT

Security and confidentiality when exchanging data and information becomes very important in the era of information and communication technology today . One of the security techniques that can be learned and developed is cryptography . cryptography is the science and art to keep secret the way encrypting messages into a form that can no longer understand its meaning .

Message is data or information that can be read and understood its meaning or commonly called plaintext . Messages that have been encrypted called ciphertext . The process of encrypting a plaintext into ciphertext is called encryption , while the process of changing ciphertext into plaintext is called decryption . Lots of existing cryptographic methods , which are divided into two , namely Classical and Modern , among these methods vigenere ciphers and stream ciphers .

To ensure the security of encrypted information , the cryptographers trying to create a complex algorithm , one of them by combining several existing algorithms . This thesis discusses a stream cipher method , combined with vigenere cipher algorithms in order to get stronger . From the test results obtained that the combination of stream ciphers and cipher vigenere have good performance in terms of file size and the time to do exactly the encryption process .

***Keywords** : Cipher , Cryptography , Combination , Classic , Modern , Stream , Vigenere*

PENDAHULUAN

Keamanan dan kerahasiaan saat melakukan pertukaran data dan informasi menjadi hal yang sangat penting pada era teknologi informasi dan komunikasi saat ini. Salah satu teknik pengamanan yang bisa dipelajari dan dikembangkan adalah kriptografi.

Banyak sekali metode kriptografi yang ada, yang dibagi menjadi dua yaitu Klasik dan Modern, diantaranya ialah metode *Stream Cipher* dan *vigenere cipher*. *vigenere cipher* pertama kali dipopulerkan oleh diplomat (sekaligus seorang kriptologis) Prancis, Blaise de Vigenere pada abad 16.

Namun pada zaman sekarang ini teknik kriptografi klasik khususnya *vigenerecipher* tidak dapat menyaingi metode-metode baru yang lebih baik, karena kesederhanaannya. Oleh karna itu, maka muncul suatu ide untuk membangun sistem keamanan menggunakan metode *vigenere cipher* yang di kombinasikan dengan *stream cipher* agar mendapatkan algoritma yang kuat.

Dalam penelitian ini, akan dibuat program enkripsi dan dekripsi dengan menggunakan kombinasi metode *stream cipher dan vegenere cipher*. Dimana Pesan merupakan data atau informasi yang dapat dibaca dan dimengerti maknanya atau biasa disebut *plaintext*. Pesan yang sudah tersandikan disebut *ciphertext*. Proses menyandikan suatu *plaintext* menjadi *ciphertext* disebut enkripsi, sedangkan proses mengubah *ciphertext* menjadi *plaintext disebut* dekripsi.

Stream Cipher

Stream cipher merupakan *cipher* yang hamper sama dengan *caesar cipher*(*cipher* yang menggeser urutan alphabet sehingga urutannya alfabetnya berubah), tetapi *cipher* ini mempunyai kunci yang unik, yaitu menggunakan karakter sebelumnya sebagai kunci.

Fungsi matematikanya :

$C = (P + K) \bmod n$

$P = (C - K) \bmod n$

C = *ciphertext* K = kunci

P = *plaintext*

n = jumlah karakter

 ISSN 2088 – 3676

Stream Cipher pertama kali diperkenalkan oleh Vernam melalui algoritmanya yang dikenal dengan nama *vernam cipher* diadobsi dari *one-time pad cipher* yang dalam hal ini karakter diganti dengan bit (0 atau 1). Cipherteks diperoleh dari pejumlahan modulo 2 satu bit plainteks dengan satu bit kunci :
C = (P + K) mod 2

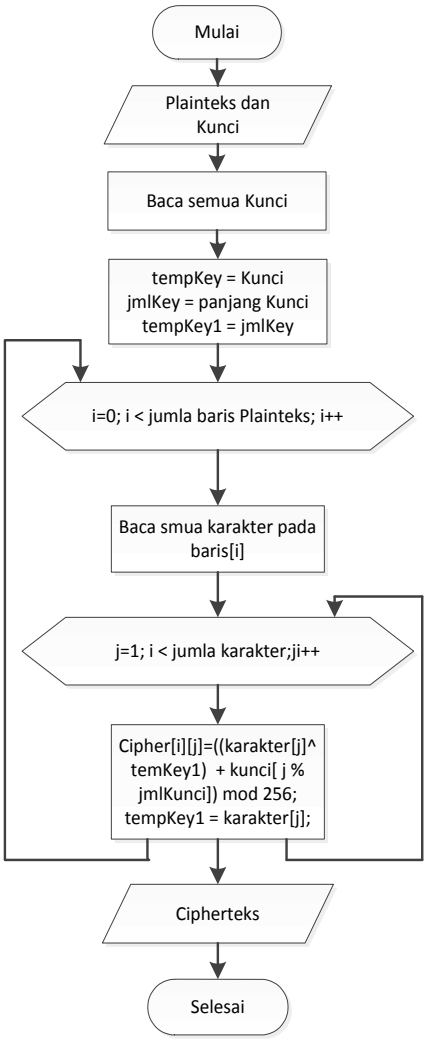
Mengingat operasi penjumlahan modulo 2 indentik dengan penjumlahan bit dengan operator *XOR*, maka persamaan dari rumus diatas dapat ditulis sebagai berikut :
C = P ⊕ K
C = *ciphertext* K = kunci
P = *plaintext*

Vigenere Cipher
Vigenere cipher merupakan *cipher* yang setiap *plaintext*-nya mempunyai beberapa kemungkinan *ciphertext*, ini terjadi karena panjang kuncinya lebih dari satu. *Cipher* ini mempunyai fungsi matematika yang sama dengan *caesar cipher*, yaitu :
C = (P + K) mod n
P = (C - K) mod n
C = *ciphertext*
K = kunci
P = *plaintext*
n = jumlah karakter
Jika hasil dekripsi (C - K) bernilai negatif (-), maka nilai ditambahkan dengan jumlah karakter (n).
Bagaimana membangun suatu aplikasi kriptografi yang kuat dengan mengkombinasikan metode *stream cipher* dan *vigenere cipher* untuk melakukan enkripsi dan dekripsi.

PERANCANGAN SISTEM

Flowchart merupakan sebuah diagram dengan symbol-symbol grafis yang menyatakan tipe operasi program yang berbeda. Sebagai representasi dari sebuah program, *flowchart* maupun algoritma dapat menjadi alat bantu untuk memudahkan perancangan alur urutan logika suatu program, memudahkan pelacakkan sumber kesalahan program, dan alat untuk menerangkan logika program.

Flowchart Proses Aplikasi Enkripsi



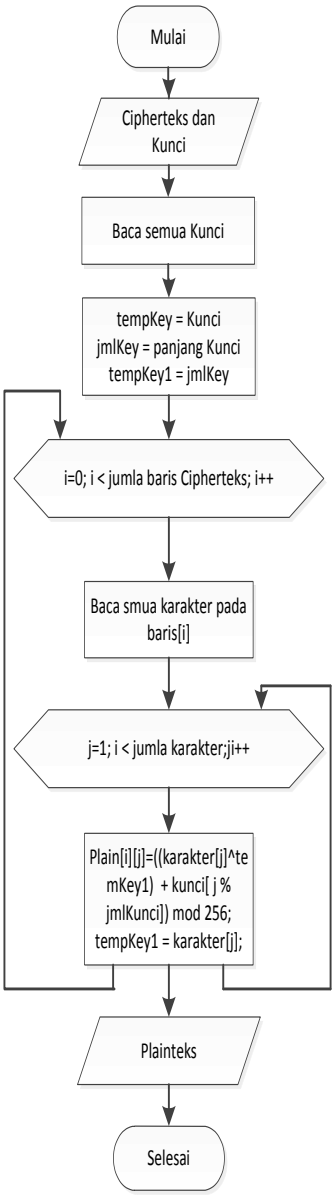
Gambar 1. *Flowchart* Proses Enkripsi

Algoritma dari proses enkripsi bagai berikut :
1. Masukan teks atau buka plainteks dan masukan Kunci.
2. Baca semua kunci yang telah dimasukan
3. Simpan kunci pada variable tempKey, panjang Kunci pada variable jmlKey, dan Simpan jmlKey pada variable tempKey1.
4. Untuk baris ke i jika lebih kecil dari jumlah baris Plainteks maka, baca baris ke i, jika tidak program selesai.
5. Untuk karakter ke j jika lebih kecil sama dengan baris yang ada, maka enkripsikan dengan rumus *Cipher[i][j] = ((karakter[j] ^ tempKey1) + kunci[j mod jmlKey]) mod 256*, lalu simpan nilai karakter[j]

pada variable tempKey1, lalu tampilkan Cipher[i][j].

6. Ulangi point ke 5 sampai semua karakter terbaca.
7. Ulangi point ke 4 sampai semua baris terbaca.

Sedangkan algoritma dari proses dekripsi sebagai berikut :



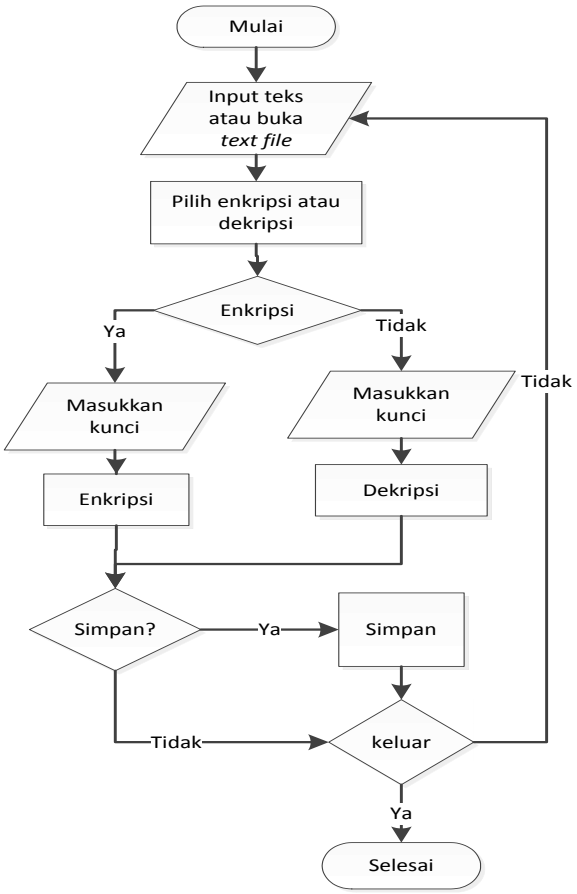
Gambar 2. *Flowchart* Proses Dekripsi

1. buka cipherteks dan masukan Kunci.
2. Baca semua kunci yang telah dimasukan
3. Simpan kunci pada variable tempKey, panjang Kunci pada variable jmlKey, dan Simpan jmlKey pada variable tempKey1.
4. Untuk baris ke i jika lebih kecil dari jumlah baris Plainteks maka, baca semua karakter dari baris ke i, jika tidak program selesai.
5. Untuk karakter ke j jika lebih kecil sama dengan baris yang ada, maka enkripsikan dengan rumus *Plain = ((karakter[j] + kunci[j mod jmlKey]) ^ tempKey1) mod 256*, lalu simpan nilai Plain pada variable tempKey1, lalu tampilkan Plain[i][j].
6. Ulangi point ke 5 sampai semua karakter terbaca.
7. Ulangi point ke 4 sampai semua baris terbaca.

Flowchart Sistem Aplikasi

Program ini dimulai dengan memilih memasukkan teks atau membuka *file*, setelah itu memilih proses apakah enkripsi atau dekripsi. Setelah itu masukkan kunci dan proses enkripsi dijalankan. Jika memilih proses dekripsi, maka masukkan kunci dan proses dekripsi dijalankan. Setelah diproses akan menampilkan lama waktu proses dan teks hasil, teks yang telah dienkripsi atau dekripsi bisa disimpan sebagai *file*. Flowchart sistem aplikasi dapat dilihat pada gambar 3 berikut ini,

ISSN 2088 – 3676



Gambar 3. *Flowchart* Sistem Aplikasi

Implementasi Proses Enkripsi

Enkripsi merupakan proses mengubah pesan asli menjadi pesan sandi. Berikut ini potongan program enkripsi dengan kombinasi metode *stream cipher* dan *vigenere cipher* :

```
for (int h=1; h<=jmlKey;h++)
        key[h]=tempKey[h];
for (int i=0; i<RichEditPlain->Lines->Count;
i++) {
        text=RichEditPlain->Lines-
>Strings[i];
        for (int j=1; j<=text.Length(); j++) {
                kar = static_cast<int>
(text[j]);
                int x = static_cast<int>
(key[1]);
                int y = static_cast<int>
(key[j%jmlKey]);
                if (j%jmlKey==0){
                        cipher =
((StrToInt(kar)^tempKey1)+x+256)%256;
                tempKey1=StrToInt(kar);
                        }
                else {
                        cipher =
((StrToInt(kar)^tempKey1)+y+256)%256;
                        tempKey1=
StrToInt(kar);
                }
                hasil =
static_cast<char>(cipher);
                temp=temp+hasil;
        }
                RichEditTemp->Lines-
>Add(temp);
                temp="";
}
```

Implementasi Proses Dekripsi

Dekripsi merupakan proses mengubah pesan sandi menjadi pesan asli. Berikut ini potongan program dekripsi dengan kombinasi metode *stream cipher* dan *vigenere cipher* :

```
for (int h=1; h<=jmlKey;h++)
        key[h]=tempKey[h];
for (int i=0; i<RichEditCipher->Lines-
>Count; i++) {
        text=RichEditCipher->Lines-
>Strings[i];
        for (int j=1; j<=text.Length(); j++) {
                kar = static_cast<int>
(text[j]);
                int x = static_cast<int>
(key[1]);
                int y = static_cast<int>
(key[j%jmlKey]);
                if (j%jmlKey==0){
                        plain =
((StrToInt(kar)-x+256)^tempKey1)% 256;
                        tempKey1 = plain;
                }
                else{
                        plain =
((StrToInt(kar)-y+256)^tempKey1)% 256;
                        tempKey1 = plain;
                }
                hasil =
static_cast<char>(plain);
                temp=temp+hasil;
        }
        RichEditTemp->Lines->Add(temp);
        temp="";
}
```

PEMBAHASAN SISTEM

Pada tampilan awal saat program dijalankan, terdapat menu yang terdiri dari Arsip, Sunting dan Bantuan. Pada tampilan *form* utama terdapat 5 tombol yaitu tombol Enkrip yang digunakan untuk melakukan proses enkripsi, tombol Dekrip yang digunakan untuk melakukan proses dekripsi, tombol Bersihkan yang digunakan untuk melakukan membersihkan semua *editor text*, tombol Buka yang digunakan untuk membuka *file*, dan tombol Simpan yang digunakan untuk menyimpan. Selain itu juga terdapat *plaintext editor* yang digunakan sebagai tempat *input* plainteks, *cihertext editor* yang digunakan untuk meletakkan hasil cipherteks, *dan keytext editor* untuk *input* kunci. Untuk melihat petunjuk pemakaian aplikasi disediakan submenu "Petunjuk" pada menu "Bantuan". Pada saat aplikasi program pertama kali dijalankan akan muncul tampilan pada gambar 4, sebagai berikut:

Gambar 4. Tampilan Utama Aplikasi

Aplikasi ini terdapat 2 proses yaitu proses enkripsi dan proses dekripsi. Cara menggunakan aplikasi ini pada proses enkripsi yaitu mengisi plainteks dengan teks atau bisa dengan membuka sebuah *file* (format didukung *.txt dan *.rtf), selanjutnya adalah menginputkan kunci pada kolom kunci, kemudian klik tombol Enkrip, hasil dari enkripsi dapat diliat pada kolom cipherteks. *File* dapat disimpan dengan ekstensi *.txt atau *.rtf.

Sedangkan untuk proses dekripsi, yaitu dengan membuka sebuah cipherteks *file* (yang berekstensi **.txt atau *.rtf*), lalu masukkan kunci. Setalah itu klik tombol Dekrip, hasilnya dapat dilihat pada kolom plainteks.

Proses Enkripsi dan Dekripsi

Proses enkripsi dan dekripsi merupakan proses utama didalam program ini. Proses dimulai ketika pengguna (*user*) memasukan input berupa pesan asli (*plaintext*) maupun kunci (*keytext*). Proses memasukan input dapat dilakukan dengan 2 cara yaitu dengan menekan submenu "Buka" pada menu "Arsip" atau menekan tombol Buka yang akan memunculkan jendela Buka, kemudian tekan tombol Plainteks, untuk membuka file dari tempat penyimpanan atau dengan mengetik pada *plaintext editor* secara manual. Berikut ini gambar 5 tampilan *Form* Buka :

Gambar 5. *Form* Buka

Sedangkan untuk menginputkan kunci dapat dilakukan dengan mengetik pada *keytext editor*. *Plaintext editor* dapat menerima karakter meliputi alfabet, angka, tanda baca, sampai karakter khusus, sedangkan untuk *keytext* editor hanya dapat menerima karakter alfabet saja dengan panjang maksimal 100 karakter. Langkah selajutnya ialah menekan tombol Enkrip untuk melakukan proses enkripsi, hasilnya akan ditampilkan pada *ciphertext editor*.

Berikut ini merupakan tampilan proses enkripsi dilakukan :

Gambar 6 Tampilan Proses Enkripsi

Untuk menyimpan hasil enkripsi dapat dilakukan dengan mekan submenu "Simpan" atau menekan tombol Simpan maka akan

ISSN 2088 – 3676

muncul jendela Simpan kemudian tekan tombol Cipherteks anda dapat memilih *directory* penyimpanan dan memberikan nama *file.*
Berikut gambar tampilan Form Simpan :

Gambar 7 *Form* Simpan

Proses enkripsi dilakukan dengan menggunakan rumus gabungan dari metode *stream cipher* dan *vigenere cipher* yaitu :

- Cs = (Ps + Ks) -> stream cipher
  Cv = (Pv + Kv) mod 52 -> vigenere cipher

menjadi :

- Csv = ((Ps + Ks)+ Kv) mod 256

Perubahan yang terjadi adalah modulo 52 menjadi 256 karakter. Pada landasan teori menggunakan modulo 52 sebab input berupa alfabet saja, sedangkan pada implementasinya berupa 256 karakter kode *ASCII*, akan tetapi tidak semua karakter khusus dapat ditampilkan sesuai dengan karakter kode *ASCII*.

Berikut ini tampilan saat proses dekripsi dilakukan :

Gambar 8. Tampilan Proses Dekripsi

Proses dekripsi dimulai dengan menginputkan *ciphertext* kedalam *ciphertext editor*, dengan menekan submenu “Buka” pada menu Arsip atau menekan tombol “Buka”, kemudia akan muncul jendela “Buka” seperti pada Gambar di atas kemudian tekan tombol “Cipherteks”

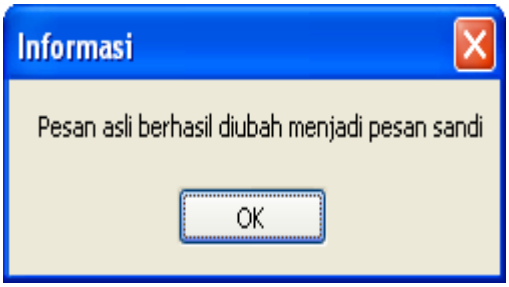
Tabel .1 Tabel Ukuran dan Waktu Enkripsi

Tipe File	Nama File	Ukuran Awal (byte)	Waktu Ekripsi (mili detik)	Ukuran File Enkripsi (byte)	
				*.txt	*.rtf
*.txt	Stegano _file	27.985	813	23.918	23.918
*.rtf	Jar_ kom	30.998	203	24.923	24.923

File

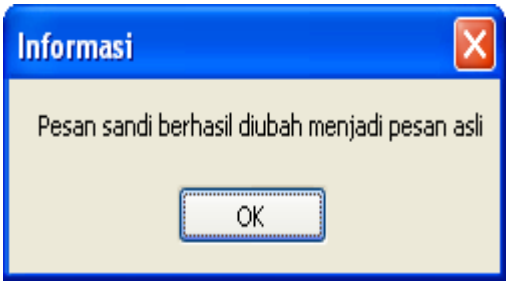
lalu pilih *file ciphertext* yang ingin di endkripsi. Setelah lakukan proses *input* dilakukan maka untuk mendekripsi *file* tersebut dapat dilakukan dengan menekan tombol “Dekrip”. Hasil dekripsi dapat dilihat pada *plaintext editor*. Pengguna dapat menyimpan dengan menekan submenu “Simpan” atau tombol “Simpan”, setelah keluar jendela Simpan seperti pada Gambar 4.4, tekan tombol “Plainteks” kemudia pilih directory penyimpanan dan berinama file.

Pada setiap proses enkripsi maupun dekripsi maka akan tampil message box seperti yang ditampilkan pada Gambar 9 dan 10 yang memberi keterangan bahwa file telah diperoses.
Berikut adalah message box yang akan ditampilkan setelah pengguna melakukan proses enkripsi :



Gambar 9. Message Proses Enkripsi

Berikut adalah message box yang akan ditampilkan setelah pengguna melakukan proses dekripsi :



Gambar 10. Message Proses Dekripsi

Pengujian Sistem dan Pembahasan yang dijadikan pengujian kombinasi metode stream cipher dan vigenere cipher adalah kinerja dari metode tersebut. Kinerja di sini berupa waktu yang dibutuhkan untuk mengenkripsi data dan perubahan ukuran *file*, berikut ini merupakan hasil dari pengujian :

Dari hasil percobaan pada tabel 1 diatas dapat dilihat *file* hasil enkripsi baik dengan format **.txt* dan **.rft* memiliki ukuran *file* yang sama, juga mengalami penyusutan ukuran dari *file* aslinya, hal ini dikarnakan file yang enkripsi yang disimpan menrupakan karakter bertipe char (1 byte / karakter).

KESIMPULAN

Kombinasi algoritma metode *stream cipher* dan *vigenere cipher* ini menghasilkan algoritma yang cukup hadal karena menggunakan 2 kunci berbeda, satu kunci dibangkitkan dengan kerakter plaintext dan satu kunci di inputkan secara manual.

Masukan (*input*) berupa karakter kode *ASCII* (*American Standard Code Infomation Interchange*), sedangkan untuk masukan kunci berupa huruf alfabet.

Aplikasi mampu mengenkripsi dan dekripsi *file* berekstensi *.txt (*Text Document*), *.rtf (*Rich Text Format*).

Ukuran file hasil enkripsi dan dekripsi menggunakan kombinasi stream cipher dan vigenere cipher memiliki ukuran yang tetap bahkan mengalami pengecilan, hal ini dikarnakan file enkripsi tidak menyimpan format penulisan dari file asli, file yang disimpan bertipe char (1 byte / karakter).

SARAN

Saran – saran yang dapat diberikan untuk pengembangan sistem ini yaitu :

1. Penggunaan IDE(integrated development Environment) yang lebih stabil dan memiliki sedikit kelemahan(*bug*) seperti Netbeans dan Qt.
2. Penggunaan algoritma kriptografi klasik ataupun modern untuk mengenkripsi byte-byte file, agar dapat mengenkripsi gambar dan grafik dan format file lainnya.
3. Perangkat Lunak ini dapat dikembangkan sebagai salah satu alternatif penyadian data pada e-mail ataupun SMS (Short Message Service) dengan kebutuhan proses dan kebutuhan yang cukup singkat.

DAFTAR PUSTAKA

Menezes A, and P. van Oorschot, 1996, **Handbook of Applied Cryptography**, CRC Press.

B. Schneier, 1996, **Applied Cryptography - Protocol, Algorithm, and Source Code in C**, Second edition, John Willey & Sons.

Dauglas Stinson, 1995, **Cryptography: Theory and Practice**, CRC Press, United States.

Dony Ariyus, 2006, **Kriptografi: Keamanan Data dan Komunikasi**, Graha Ilmu, Yogyakarta.

Rinaldi Munir, 2006, **Kriptografi**, Infomatika, Bandung.

ISSN 2088 – 3676