PROGRAM APLIKASI KRIPTOGRAFI PENYANDIAN *ONE TIME PAD* MENGGUNAKAN SANDI *VIGENERE*

Lis Endah Pratiwi, Rini Marwati, Isnie Yusnitha

Departemen Pendidikan Matematika FPMIPA Universitas Pendidikan Indonesia

*Coresponding author: lisendahpratiwi@gmail.com

ABSTRAK: Penggabungan konsep algoritma kriptografi sandi *Vigenere* dan *One time pad* memiliki keunggulan dalam hal peringkasan/pemadatan data. Pada sandi One time pad, kunci yang digunakan untuk proses enkripsi dan dekripsi mempunyai panjang yang sama dengan data awal, sedangkan dalam penggabungan konsep ini kunci sandi yang digunakan cukup memiliki panjang kunci setengah dari panjang data awal. Teori dasar matemaika yang digunakan dalam penggabungan konsep dua sandi kriptografi ini yang menjadi masalah yang perlu dibahas, juga merancang dan membuat program aplikasi kriptografi gabungan dua sandi tersebut dalam penulisan skripsi ini. Metodelogi penelitian yang diterapkan dalam penulisan skripsi ini berupa studi literatur, pengembangan program, pembuatan program serta pengujian program aplikasi kriptografi. Hasil dalam penulisan skripisi ini berupa program aplikasi kriptografi yang dapat mempermudah proses enkripsi dan dekripsi sandi kriptografi tersebut. Sehingga dapat disimpulkan bahwa algoritma dari penggabungan dua sandi kriptografi dapat dijelaskan secara matematis dan program aplikasi kriptogradi dapat dibuat mengguanakan bahasa pemrograman Delphi 7.

Kata Kunci: program aplikasi, sandi *one time pad*, sandi *Vigenere*

PENDAHULUAN

Berkembangnya teknologi dalam berkomunikasi jarak jauh memberikan dampak negatif salah satunya kebocoran informasi rahasia. Salah satu upaya pencegahan kebocoran, adalah dengan penyandian informasi dengan kunci. Orang yang mengetahui kunci pada sandi data tersebut hanya orang yang saling bertukar informasi sehingga rahasia terjamin. Dalam dunia sains, terdapat ilmu yang mempelajari penyandian data yang disebut Kriptografi.

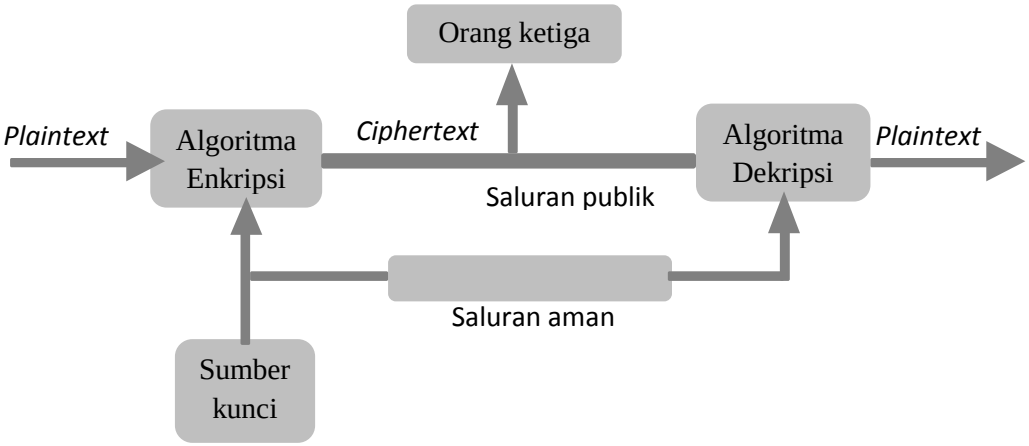
Kriptografi adalah ilmu menyembunyikan data atau pesan. Sandi one time pad dan Vigenere merupakan kriptografi klasik dengan teknik operasi sandi mengunakan metode subtitusi. Sandi *Vigenere* merupakan sistem sandi poli-alfabetik yang sederhana barisan karakter kunci yang berbeda. sandi *one time pad*, sandi ini merupakan sandi yang mencapai kerahasiaan sempurna (*perfect secrecy*). Kunci

yang digunakan sandi *one time pad* ini berbeda untuk setiap karakternya, dengan panjang kunci sama dengan panjang pesan yang akan dienkripsi dan kunci diberikan secara acak dan hanya digunakan sekali.

Dalam jurnal Amroodi *et al.* (2013) menjelaskan algoritma kriptografi modifikasi berupa penggabungan konsep sandi *one time pad* dan sandi *Vigenere*. Pada sandi *one time pad*, kunci yang digunakan untuk proses enkripsi dan dekripsi mempunyai panjang yang sama dengan data awal, sedangkan dalam penggabungan konsep ini kunci sandi yang digunakan cukup memiliki panjang kunci setengah dari panjang data awal. Hal ini berguna untuk peringkasan/pemadatan data. Penulis mencoba membuat program aplikasi berdasarkan ide kriptografi modifikasi tersebut.

KRIPTOGRAFI

"Kriptografi adalah ilmu pengetahuan dengan teknik matematika yang berhubungan dengan aspek keamanan jaringan, misalnya kerahasiaan data, keutuhan data, autentikasi data dan keaslian data" seperti dikutip dari Menezes *et al*. (1996). Sistem kriptografi klasik terdiri dari plaintext, yaitu teks asli yang akan disandikan, ciphertext adalah teks dalam bentuk sandi, enkripsi adalah proses mengubah plaintext menjadi ciphertext, key merupakan masukkan pada algoritma enkripsi dan dekripsi, dekripsi adalah proses mengubah ciphertext menjadi plaintext. Berikut ini merupakan gambar sistem kriptografi klasik.



Gambar 2.1 Sistem Kriptografi Klasik (Stinson, 2006: 2)

Dalam Pratiwi (2014), menjelaskan sandi Vigenere secara matematis jika bekerja dalam $\mathbb{Z}_m$, maka proses enkripsi dapat ditulis sebagai berikut,

$$c_i = (p_i + k_j)\, m \quad n$$

dimana, c_i = karakter ke-i *ciphertext*
p_i = karakter ke-i *plaintext*
k_j = karakter ke-j kunci
n = bilangan bulat modulo n
i, j = indeks (bilangan asli)

dan proses dekripsi sandi *Vigenere* ditulis dengan:

$$p_i = (c_i - k_j) \bmod n$$

Definisi 2.1: Sandi *one time pad* (Menezes *et al.*, 1996)

$$c_i = p_i \oplus k_i \quad \text{untuk} \quad i = 1, 2, 3, \ldots$$

dimana, c_i = karakter ke-i *ciphertext*
p_i = karakter ke-i *plaintext*
k_i = karakter ke-i kunci
i = indeks (bilangan asli)
$\oplus$= operator *bitwise* XOR

Jika barisan kunci diambil secara acak dan hanya digunakan sekali. Sedangkan untuk proses dekripsi sandi OTP adalah

$$p_i = c_i \oplus k_i .$$

KRIPTOGRAFI MODIFIKASI

Diberikan sebuah *plaintext* dengan panjang m. Proses enkripsi sandi OTP menggunakan sandi *Vigenere*, selanjutnya disebut sandi modifikasi, hampir sama dengan proses enkripsi sandi OTP, yaitu dengan melakukan operasi XOR antara p_i dan p_i^{-1}. Untuk memudahkan, notasi yang digunakan adalah:

m = panjang *plaintext*
$\bar{P}'$ = *plaintext* awal berupa teks $(a_1, a_2, \ldots, a_m)$

$$\bar{P} = \begin{cases} \bar{P}', \text{jika } m \text{ genap} \\ \bar{P}' + \text{' '}, \text{jika } m \text{ ganjil; untuk suatu ' ' karakter spasi} \end{cases}$$

$(\bar{P})^{-1}$ = invers/ balikan dari $\bar{P}$
$= (a_m, a_{m-1}, \ldots, a_1)$
$\overline{P^0} = (a_1, a_2, \ldots, a_{\frac{m}{2}})$
$\overline{P^1} = (a_{\frac{m}{2}}, a_{\frac{m}{2}+1}, \ldots, a_m)$
$\bar{K}$ = kunci
$\bar{C}$ = *ciphertext* berupa teks $(c_1, c_2, \ldots, c_m)$
$\overline{C^0} = (c_1, c_2, \ldots, c_{\frac{m}{2}})$

$\overline{C^1} = (c_{\frac{m}{2}}, c_{\frac{m}{2}+1}, \dots, c_m)$
p = *plaintext* dalam bentuk kode ASCII
$(p)^{-1}$ = invers/ balikan dari p
p_i = karakter ke-i dari p
$(p^{-1})_i$ = karakter ke-i dari p^{-1}
k = kunci dalam bentuk kode ASCII
k_i = karakter ke-i dari k
c = *ciphertext* dalam bentuk kode ASCII
c_i = karakter ke-i dari c
i = indeks (bilangan asli)

Berdasarkan Amroodi *et al.* (2013) secara matematis proses enkripsi sandi modifikasi ini ditulis dengan:

$$\bar{C} = \bar{P} \oplus (\bar{P})^{-1}$$

dengan notasi $\bar{C}$, $\bar{P}$, dan $(\bar{P})^{-1}$ seperti yang dimaksud di atas.

Dalam kriptografi modifikasi ini, karakter diubah kedalam kode ASCII sebelum melakukan proses enkripsi. ASCII (*American Standart Code for Information Interchange*) merupakan suatu standar internasional dalam kode huruf dan simbol seperti *Hex* dan *Unicode* tetapi ASCII lebih bersifat universal. Jumlah kode ASCII adalah 255 kode.

Contoh 3.1 Diberikan *Plaintext* = kesan

$$\bar{P}^r = k \quad ; m = 5$$
$$\bar{P} = k \quad ; m = 6$$
$$(\bar{P})^{-1} = n$$
$$p = (107, 101, 115, 97, 110, 32)$$
$$(p)^{-1} = (32, 110, 97, 115, 101, 107)$$

Proses enkripsi untuk $\bar{P} = k$ dan $(\bar{P})^{-1} = n$ adalah,

$$c_i = p_i \oplus (p^{-1})_i$$

$$c_1 = 107 \oplus 32$$
$$p_1 = 107 \quad 0110\ 1011$$
$$(p^{-1})_1 = 32 \quad 0010\ 0000 \ \oplus$$
$$0100\ 1011 = 75$$

sehingga diperoleh $c_1 = 75$,

$$c_2 = 101 \oplus 110$$
$$p_2 = 101 \quad 0110\ 0101$$
$$(p^{-1})_2 = 110 \quad 0110\ 1110 \ \oplus$$
$$0000\ 1011 = 11$$

sehingga diperoleh $c_2 = 11$,

$$c_3 = 115 \oplus 97$$
$$p_3 = 115 \quad 0111\ 0011$$
$$(p^{-1})_3 = 97 \quad 0110\ 0001$$
$$0001\ 0010 = 18 \ \oplus$$

sehingga diperoleh $c_3 = 18$,

$$c_4 = 97 \oplus 115 = 115 \oplus 97 = 18$$
$$c_5 = 110 \oplus 101 = 101 \oplus 110 = 11$$
$$c_6 = 32 \oplus 107 = 110 \oplus 32 = 75$$

setelah semua c_i diperoleh, maka

$$c = (75, 11, 18, 18, 11, 75)$$

diubah dalam karakter ASCII menjadi,

$$\bar{C} = K\text{VTDC1DC1VT}K$$

Untuk memudahkan pembacaan lihat tabel 3.1.

Tabel 3.1 Tabel Enkripsi Sandi Modifikasi

$\bar{P}$	k	e	s	a	n	
$(\bar{P})^{-1}$		n	a	s	e	k
$\bar{C}$	K	VT	DC1	DC1	VT	K

Sebagaimana yang dikemukakan oleh Amroodi *et al*. (2013) bahwa, misalkan

$$\bar{P} = \overline{P^G} \parallel \overline{P^1} \quad ; \quad \parallel = \text{operator rangkaian.}$$

dengan, $m(\overline{P^G}) = m(\overline{P^1})$.

Kunci yang digunakan untuk proses dekripsi,

$$\bar{K} = \overline{P^G}$$

Untuk arti notasi $\bar{P}, \overline{P^G}, \overline{P^1}, m$ dan $\bar{K}$ seperti yang di atas.

Contoh 3.2 Misalkan $\bar{P} = k$, maka

$$\bar{P} = k$$
$$\bar{P} = k \quad \parallel a$$

Jadi, $\overline{P^G} = k$ dan $\overline{P^1} = a$. Dan kunci yang digunakan untuk proses dekripsi adalah,

$$\bar{K} = \overline{P^G}$$
$$\bar{K} = k \quad .$$

Sedangkan proses dekripsi sandi modifikasi ini cukup unik, yaitu dengan men-XOR-kan setengah karakter pertama dari *ciphertext* yang telah didapat dari proses enkripsi. Karena *ciphertext* yang digunakan hanya setengah dari panjang *ciphertext* yang telah diperoleh, maka *plaintext* yang dapat dipecahpun hanya setengah sampai karakter terakhir dari *plaintext* yang asli.

Dalam jurnal Amroodi *et al.* (2013) dijelaskan bahwa secara matematis proses dekripsi sandi modifikasi ini dapat ditulis:

$$(\overline{P^1})^{-1} = \overline{C^c} \oplus \overline{K}$$

dengan notasi $(\overline{P^1})^{-1}$, $\overline{C^c}$ dan $\overline{K}$ seperti yang dimaksud di atas.

Contoh 3.3 Misalkan $\bar{C} = K$VTDC1DC1VTK, maka $\overline{C^c} = K$VTDC1 dan $\overline{K} = k$.

$$c^{ii} = (75, 11, 18)$$
$$k = (107, 101, 115)$$

Sehingga proses dekripsinya adalah

$$(p^1)^{-1}{}_i = (c^{ii})_i \oplus k_i$$

$$(p^1)^{-1}{}_1 = 75 \oplus 107$$
$$(c^{ii})_1 = 75 \quad 0100\ 1011$$
$$k_1 = 107 \quad 0110\ 1011 \ \oplus$$
$$0010\ 0000 = 32$$

sehingga diperoleh $(p^1)^{-1}{}_1 = 75$,

$$(p^1)^{-1}{}_2 = 11 \oplus 101$$
$$(c^{ii})_2 = 11 \quad 0000\ 1011$$
$$k_2 = 101 \quad 0110\ 0101 \ \oplus$$
$$0110\ 1110 = 110$$

sehingga diperoleh $(p^1)^{-1}{}_2 = 110$,

$$(p^1)^{-1}{}_3 = 18 \oplus 115$$
$$(c^{ii})_3 = 18 \quad 0001\ 0010$$
$$k_3 = 115 \quad 0111\ 0011 \ \oplus$$
$$0110\ 0001 = 97$$

sehingga diperoleh $(p^1)^{-1}{}_3 = 97$. Setelah semua $(p^1)^{-1}{}_i$ diperoleh, maka

$$(p^1)^{-1} = (32, 110, 97)$$

diubah dalam karakter menjadi,

$$(\overline{P^1})^{-1} = n$$

Untuk memudahkan pembacaan lihat tabel 3.2.

Tabel 3.2 Tabel Dekripsi Sandi Modifikasi

$\overline{C^0}$	K	VT	DC1
$\overline{K}$	k	e	s
$(\overline{P^1})^{-1}$		n	a

Setelah memperoleh setengah sampai karakter terakhir dari *plaintext* yang asli dari proses dekripsi, hal selanjutnya yang dilakukan adalah penggabungan kunci dengan invers dari hasil dari proses dekripsi tadi. Proses penggabungan dengan berdasarkan teorema berikut ini,

Teorema 3.4 (Amroodi *et al.*, 2013)

Diketahui, $\overline{P}, \overline{P^0}, \overline{P^1}, \overline{C^0}$ *dan* $\overline{K}$ *seperti yang di atas. Maka,*

$$\overline{P} = \overline{K} \parallel (\overline{C^0} \oplus \overline{K})^{-1}$$

Bukti:

Karena, $(\overline{P^1})^{-1} = \overline{C^0} \oplus \overline{K}$ sedemikian sehingga,

$$\begin{aligned} \overline{K} \parallel (\overline{C^0} \oplus \overline{K})^{-1} &= \overline{K} \parallel ((\overline{P^1})^{-1})^{-1} \\ &= \overline{K} \parallel \overline{P^1} \\ &= \overline{P^0} \parallel \overline{P^1} \\ &= \overline{P} \end{aligned}$$

■

Menurut teorema 3.4, jika gabungan dari kunci dan operasi XOR dari $\overline{C^0}$, yaitu setengah karakter pertama dari *ciphertext* dan kunci yang diinverskan akan menghasilkan *plaintext*.

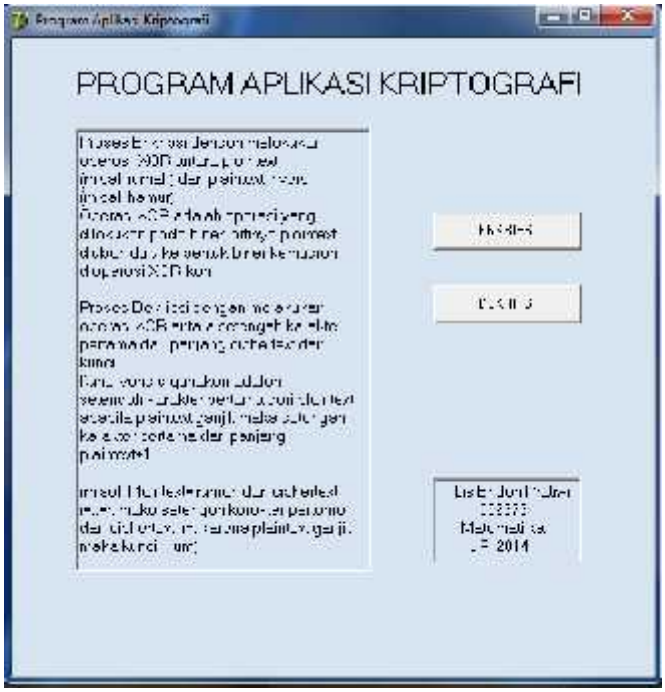
Contoh 3.5 Misalkan $\overline{C^0} = K$VTDC1 dan $\overline{K} = k$.

$$\begin{aligned} \overline{P} &= \overline{K} \parallel (\overline{C^0} \oplus \overline{K})^{-1} \\ &= \overline{K} \parallel ((\overline{P^1})^{-1})^{-1} \\ &= k \parallel (n)^{-1} \\ &= k \parallel a \\ &= k \end{aligned}$$

Terlihat pada contoh 3.5, bahwa benar jika gabungan dari kunci (k) dan operasi XOR dari $\overline{C^0}$ dan kunci yang diinverskan, $\overline{K}$, ($(n)^{-1} = a$) akan menghasilkan *plaintext* secara utuh (k).

PROGRAM APLIKASI KRIPTOGRAFI

Pada subbab ini dibahas tentang tampilan program aplikasi yang telah dibuat dan hasil pengujian program aplikasi terhadap perhitungan manual. Berikut tampilan antarmuka pengguna program aplikasi yang dibuat menggunakan Borland Delphi 7.



Gambar 4.1 Tampilan Antarmuka *Home*

Gambar 4.1 adalah tampilan antarmuka dari halaman *home* program aplikasi kriptografi. Pada program aplikasi ini jumlah kode yang digunakan adalah 95 kode ASCII terdiri dari abjad (A..Z), numerik (0..9), karakter khusus (?/>.<,":';{[}]|\+=_-)(*&^%$#@!~`) dan karakter spasi ( ). Sehingga pada perhitungannya nanti kode ASCII yang asli akan dikurangkan dengan 32, misalkan karakter ! pada kode ASCII adalah 33, tetapi pada program aplikasi akan bernilai 0, karakter M pada kode ASCII bernilai 77, tetapi pada program aplikasi akan bernilai 45.

Gambar 4.2 merupakan tampilan antarmuka dari halaman proses enkripsi, dimana masukkan plaintext adalah mathematician. Sedangkan pada tabel 4.2 adalah hasil perhitungan manual dengan plaintext yang sama.

Tabel 4.2 Tabel Enkripsi *Plaintext* m

$\bar{P}$	m	a	t	h	e	m	a	t	i	c	i	a	n	
$(\bar{P})^{-1}$		n	a	i	c	i	t	a	m	e	h	t	a	m
$\bar{C}$	m	/	5	!	&	$	5	5	$	&	!	5	/	m

Gambar 4.2 Tampilan Proses Enkripsi

Terlihat pada hasil yang diperoleh dari perhitungan manual dan hasil yang didapatkan program telah sama. Hal ini menunjukkan tidak terdapat kesalahan perhitungan pada program aplikasi.

Selanjutnya perhitungan manual proses dekripsi terdapat pada tabel 4.3.

Tabel 4.3 Tabel Dekripsi *Ciphertext* $m/5! \&\$5$

$\overline{C^0}$	m	/	5	!	&	$	5
$\overline{K}$	m	a	t	h	e	m	a
$(\overline{P^1})^{-1}$		n	a	i	c	i	t

Setelah proses dekripsi, hal selanjutnya adalah mendapatkan kembali *plaintext* secara utuh, maka lakukan sesuai teorema 3.4, sehingga diperoleh kembali plaintext secara utuh, yaitu m *he* .

Dapat dilihat pada perhitungan manual proses dekripsi dan perhitungan yang dilakukan oleh program aplikasi pada gambar 4.3, menunjukkan hasil yang sama.

Hal ini menunjukkan tidak ada kesalahan dalam proses dekripsi pada program aplikasi.

Gambar 4.3 Tampilan Proses Dekripsi

KESIMPULAN

Berdasarkan pembahasan padah subbab di atas, dapat disimpulkan bahwa, pembuatan program aplikasi kriptografi dapat dibuat menggunakan bahasa pemrograman Delphi 7 dengan tampilan *end user*. Pengujian program aplikasi kriptografi dilakukan dengan perbandingan hasil yang diperoleh dari program aplikasi dan hasil yang diperoleh dari perhitungan manual. Berdasarkan hasil perbandingan tersebut, hasil yang diperoleh dari keduanya sama. Hal ini menunjukkan bahwa program aplikasi yang telah dibuat sesuai dengan apa yang diinginkan. Pada kenyataannya program aplikasi dapat dikembangkan dengan penggunaan kode ASCII yang lebih lengkap.

DAFTAR PUSTAKA

Amroodi, A., Toonabi, A., dan Zaghian, A. (2013). *The produce of One time pad cipher using Vigenere cipher*. Journal of Science, Engineering and Technology. 8. (2322-2441). 35-38.

Menezes, A., Oorschot, P. dan Vanstone, A..(1996).*Handbook of Applied Cryptography*. USA: CRC Press.

Stinson, D. R. (2006). "*Cryptography Theory and Practice*". New York: Chapman & Hall/CRC.

Pratiwi, Lis Endah. (2014). "*Program Aplikasi Kriptografi Penyandian One Time Pad Menggunakan Sandi Vigenere*". Skripsi pada Program Studi Matematika FPMIPA UPI. Bandung: Tidak diterbitkan.