Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)
SAINTEKS 2020
ISBN: 978-602-52720-7-3
Februari 2020
Hal 415-419

Pengamanan File Docx Menerapkan Algoritma Gronsfeld

Berliana Oktaviani Sinaga, Doni Almahera, Sri Wahyuni, Imam Saputra

Program Studi Teknik Informatika, STMIK Budi Darma, Medan Indonesia
Email: [1]berlianaoktavianisinaga@gmail.com, [1]donialmahera18@gmail.com, [1]swahyuni2398@gmail.com, [1]saputraimam69@gmail.com

Abstrak−Perkembangan ilmu pengetahuan dan teknologi saat ini khususnya pada sistem keamanan data sangat berkembang dengan pesat. Berbagai manfaatnya bisa kita rasakan. Menjaga keamanan file ataupun data itu sangat penting dilakukan guna menjaga privasi dan mencegah berbagai kerusakan yang mungkin bisa terjadi. Salah satu teknik yang dapat dilakukan untuk proses pengamanan adalah dengan menerapkan teknik kriptografi dengan perhitungan algoritma Gronsfeld. Gronsfeld merupakan algoritma yang hanya menggunakan numeric sebagai key. Key tersebut ditetapkan oleh programmer. Hasil yang dapat diperoleh dari penelitian ini adalah isi file dari sebuah dokumen menjadi sulit dibaca karena sudah di enkripsi, dan hanya dapat dibaca jika orang tersebut mempunyai kunci.

Kata Kunci: Kriptografi, File Docx, Algoritma Gronsfeld

1. PENDAHULUAN

Dengan adanya teknologi informasi yang terus berkembang mumudahkan kita untuk saling bertukar informasi. Salah satu adalah dengan berbagi dokumen, format file yang sering kita pakai saat ini adalah docx. Docx sendiri merupakan format Microsoft Word 2007 ke atas. Dalam dunia pekerjaan komputerisasi sangat dibutuhkan sekali untuk menjalankan suatu aktivitas dan kegiatan. Begitu pun dalam hal menjaga keamanan, dimana hal tersebut menimbulkan tuntutan agar tersedianya sebuah sistem pengamanan data yang lebih baik lagi. Di dalam proses pengamanan data terdapat beberapa cara yang dapat dilakukan yaitu dengan kriptografi (teknik penyamaran data /pesan) dan steganografi (penyembunyian data / pesan) [1]**.**

Kriptografi adalah teknik pengamanan pesan melalui penyandian. Jadi seandainya data / file tersebut ada pada pihak yang tidak berwenang / tidak berhak, maka data tersebut tidak dapat dimengerti dan dibaca karena sudah di enkripsi. Enkripsi sendiri merupakan sebuah proses pengubahan data asli menjadi data yang sulit untuk dimengerti artinya dengan menggunakan sebuah algoritma tertentu. Nah prosess untuk mengembalikan data tersebut seperti semula disebut deskripsi [2].

Di kesempatan kali penulis akan mengamankan isi file berformat docx dengan sebuah contoh kasus sederhana menggunakan algoritma gronsfeld cipher.

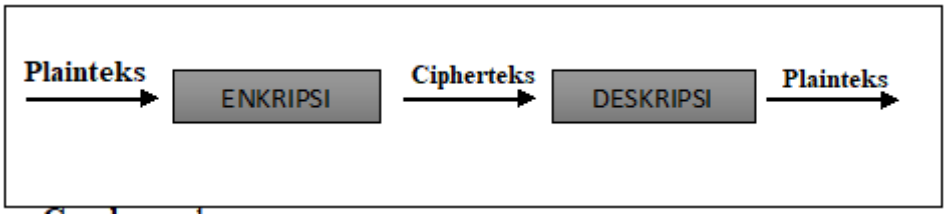
2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kata kriptografi berasal dari 2 kata yaitu, *kryptos* yang berarti rahasia dan *graphien* yang berarti tulisan. Jadi kriptografi adalah sebuah teknik yang mempelajari enkripsi dimana data / informasi diacak menggunakan suatu kunci enkripsi, sehingga data menjadi sulit dibaca jika orang tersebut tidak memiliki kunci deskripsi. Atau dapat diartikan dengan mudah kriptografi adalah teknik pengamanan pesan melalui penyandian. [3]

Dalam pertukaran informasi, pasti terdapat yang namanya *sender* (pengirim) yang ingin melakukan pengiriman pesan ke *reseptiont (* penerima) dengan menginginkan pesan tersebut aman dan kerahasiannya dapat terjaga. Sebab itu dibuttuhkan sebuah system yang dapat mengamankan isi pesan. Di dalam kriptografi terdapat beberapa terminologi yaitu:

1. *Plainteks*, yaitu pesan asli yang akan dikirim.
2. *Cipherteks*, yaitu bentuk pesan yang sudah tersandi.
3. *Enkripsi*, yaitu proses yang menyandikan *plainteks* menjadi *cipherteks.*
4. *Deskripsi*, yaitu proses yang mengembalikan *cipherteks* menjadi *plainteks.*



Gambar 1. Skema Proses Enkripsi dan Deskripsi [4]

2.2 File Docx

Docx adalah jenis format untuk file versi Microsoft Word 2007 ke atas dan tentunya mempunyai banyak keuntungan dibandingkan dengan format file sebelumnya. Diantaranya adalah mempunyai ukuran simpan yang relative lebih kecil, mempunyai ketahanan terhadap serangan virus saat berbagi dokumen dan masih banyak lagi. Dokumen dengan format docx dapat dengan muda dikonversikan ke format *file html, rtf, doc* dan juga format lainya [5].

ISBN: 978-602-52720-7-3
Februari 2020
Hal 415-419

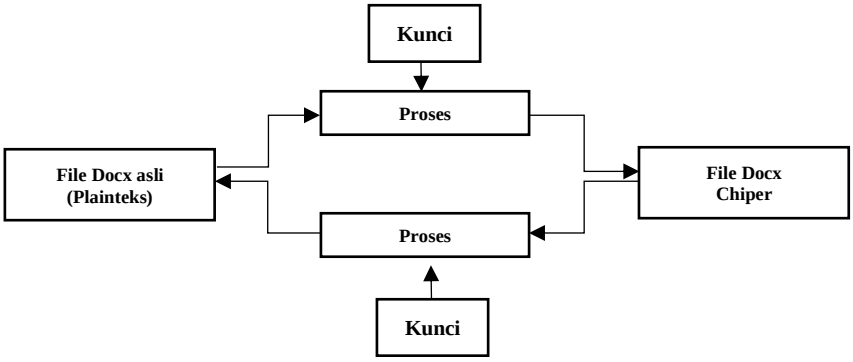
2.3 Algoritma Gronsfeld

Gronsfeld merupakan algoritma penyandian yang hanya menggunakan numeric sebagai *key*. *Key* tersebut ditetapkan oleh *programmer*. Cara kerjanya menyerupai sandi Vigenere, dimana akan pengulangan secara periodik untuk kunci dengan tujuan agar panjang plaintek dan kunci sama [6].

3. ANALISA DAN PEMBAHASAN

Pengamanan *file* yang dilakukan dengan teknik enkripsi biasanya memiliki dampak menurunnya sebuah performansi tersendiri untuk *file*, seperti lamanya waktu yang dibutuhkan untuk mengakses *file* tersebut. Padahal dalam sebuah organisasi maupun perusahaan sangat penting sekali pengaksesan *file* secara cepat. Sehingga sangat diperlukan sebuah pengamanan *file* menggunakan enkripsi tanpa harus menurunkan perfomansinya. [1]

Tujuan utama dilakukan penerapan teknik enkripsi ini adalah untuk melindungi dan menjaga kerahasiaan file ataupun data sehingga tidak dapat diakses oleh sembarang orang yang tidak memiliki hak otorisasi terhadap *file* tersebut. Hal yang dilakukan untuk melindungi dengan cara menjaga kunci *enkripsi* dan *deskripsi*. Berikut ini merupakan skema dari proses kriptografi sistem pengamanan *file* berformat docx. [7]



Gambar 2. Proses *Enkripsi dan Deskripsi*

2.4 Penerapan Algoritma Gronsfeld Chiper Dalam Menyandikan File Docx

Untuk menginplementasikan algoritma Gronsfeld Cipher penulis harus megubahya ke dalam bilangan ASCII. Persamaan Algoritma Gronsfeld Cipher:

Ci = (Pi + Ki) mod 256 ..... (1)

Keterangan:

Ci = nilai desimal (ASCII) karakter *cipher* ke-i
Pi = nilai desimal (ASCII) karakter *plainteks* ke-i
Ki = nilai desimal (ASCII) karakter kunci ke-i

K merupakan jumlah pergeseran huruf pada table kunci. Dimana disini nilai K yang perlu dirahasiakan. Langkah-langkah yang harus dilakukan untuk proses enkripsi sebagai berikut:

a. Tentukan terlebih dahulu kunci dan *plainteks* yang ingin dienkripsi.
b. Jika panjang kunci dan *plainteks* tidak sama, maka lakukan pengulangan kunci secara berurut (periodik) hingga jumlah karakter kunci sama dengan plainteksnya.
c. Kemudian ubah terlebih dahulu nilai *plainteks* ke dalam bentuk ASCII (bilangan desimal) dan ditambahkan dengan nilai kunci. Apabila jumlah penambahan plainteks dan kunci lebih besar dari jumlah mod (256), maka ambil nilai hasil pembagiaanya.
d. Setelah mendapat hasil dari penjumlahannya, maka langkah selanjutnya adalah mengubah karakter ke bentuk semula. [8]

Dalam sebuah *file* berformat docx terdapat isi sebagai berikut :

"PAK IMAM DOSEN TERFAVORITE "

1. *Plainteks* = PAK IMAM DOSEN TERFAVORITE
2. Kunci = SUBSCRIBE

2.4.1 Proses Enkripsi

Tabel 1. Enkripsi *Plainteks*

Plainteks	P	A	K		I	M	A	M		D	O	S	E	N		T	E	R	F	A	V	O	R	I	T	E
Desimal	80	65	75	32	73	77	65	77	32	68	79	83	69	78	32	84	69	82	70	65	86	79	82	73	84	69
Kunci	S	U	B	S	C	R	I	B	E	S	U	B	S	C	R	I	B	E	S	U	B	S	C	R	I	B

ISBN: 978-602-52720-7-3
Februari 2020
Hal 415-419

Plaintteks	P	A	K		I	M	A	M		D	O	S	E	N		T	E	R	F	A	V	O	R	I	T	E
Desimal	83	85	66	83	67	83	73	66	69	83	85	66	83	67	82	73	66	69	83	85	66	83	67	82	73	66

Proses Enkripsi:
Ci = (Pi +Ki) mod 256

P C1= (80+83) mod 256
= 163 mod 256
= 163 (£)
A C2= (65+85) mod 256
= 150 mod 256
= 150 (-)
K C3= (75+66) mod 256
= 141 mod 256
= 141 ()
Spasi C4= (32+83) mod 256
= 115 mod 256
= 115 (s)
I C5= (73+67) mod 256
= 140 mod 256
= 140 (œ)
M C6= (77+82) mod 256
= 159 mod 256
= 159 (ÿ)
A C7= (65+73) mod 256
= 138 mod 256
= 138 (Š)
M C8= (77+66) mod 256
= 13 mod 256
= 143 ()
spasi C9= (32+69) mod 256
= 101 mod 256
= 101 (e)
D C10= (68+83) mod 256
= 151 mod 256
= 151 (_)
O C11= (79+85) mod 256
= 164 mod 256
= 164 (¤)
S C12= (83+66) mod 256
= 149 mod 256
= 149 (•)
E C13= (69+83) mod 256
= 152 mod 256
= 152 (-)
N C14= (78+67) mod 256
= 145 mod 256
= 145 (‘)
spasi C15= (32+82) mod 256
= 114 mod 256
= 114 (r)
T C16= (84+73) mod 256
= 157 mod 256
= 157 ()
E C17= (69+66) mod 256
= 135 mod 256
= 135 (‡)
R C18= (82+69) mod 256
= 151 mod 256
= 151 (_)
F C19= (70+83) mod 256
= 153 mod 256
= 153 (™)
A C20= (65+85) mod 256
= 150 mod 256
= 150 (-)
V C21= (86+66) mod 256
= 152 mod 256
= 152 (~)
O C22= (79+83) mod 256
= 162 mod 256
= 162 ()
R C23= (82+67) mod 256
= 149 mod 256
= 149 (•)
I C24= (73+82) mod 256
= 155 mod 256
= 155 (>)
T C25= (84+73) mod 256
= 157 mod 256
= 157 ()
E C26= (69+66) mod 256
= 135 mod 256
= 135 (‡)

Nilai desimal cipherteks:
163, 150, 141, 115, 140, 159, 138, 143, 101, 151, 164, 149, 152, 145, 114, 157, 135, 151, 153, 150, 152, 162, 149, 155, 157, 135

Karateristik Cipherteks:
£- sœÿŠ e_¤•-‘r ‡_™-~ •> ‡

2.4.2 Proses Dekripsi

Tabel 2. Dekripsi Cipherteks

Cipherteks	£	-		s	œ	ÿ	Š		e	_	¤	•	-	‘	r		‡	_	™	-	~		•	>		‡
Desimal	163	150	141	115	140	159	138	143	101	151	164	149	152	145	114	157	135	151	153	150	152	162	149	155	157	135
Kunci	S	U	B	S	C	R	I	B	E	S	U	B	S	C	R	I	B	E	S	U	B	S	C	R	I	B
Desimal	83	85	66	83	67	83	73	66	69	83	85	66	83	67	82	73	66	69	83	85	66	83	67	82	73	66

ISBN: 978-602-52720-7-3
Februari 2020
Hal 415-419

Proses Deskripsi:
Pi = (Ci -Ki) mod 256

P1= (163-83) mod 256
= 80 mod 256
= 80 (P)
P2= (150-85) mod 256
= 65 mod 256
= 65 (A)
P3= (141-66) mod 256
= 75 mod 256
= 75 (K)
P4= (115-83) mod 256
= 32 mod 256
= 32 ()
P5= (140-67) mod 256
= 73 mod 256
= 73 (I)
P6= (159-82) mod 256
= 77 mod 256
= 77 (M)
P7= (138-73) mod 256
= 65 mod 256
= 65 (A)
P8= (143-66) mod 256
= 77 mod 256
= 77 (M)
P9= (101-69) mod 256
= 32 mod 256
= 32 ()
P10= (151-83) mod 256
= 68 mod 256
= 68 (D)
P11= (164-85) mod 256
= 79 mod 256
= 79 (O)
P12= (149-66) mod 256
= 83 mod 256
= 83 (S)
P13= (152-83) mod 256
= 69 mod 256
= 69 (E)
P14= (145-67) mod 256
= 78 mod 256
= 78 (N)
P15= (114-82) mod 256
= 32 mod 256
= 32 ()
P16= (157-73) mod 256
= 84 mod 256
= 84 (T)
P17= (135-66) mod 256
= 69 mod 256
= 69 (E)
P18= (151-69) mod 256
= 82 mod 256
= 82 (R)
P19= (153-83) mod 256
= 70 mod 256
= 70 (F)
P20= (150-85) mod 256
= 65 mod 256
= 65 (A)
P21= (152-66) mod 256
= 86 mod 256
= 86 (V)
P22= (162-83) mod 256
= 79 mod 256
= 79 (O)
P23= (149-67) mod 256
= 82 mod 256
= 82 (R)
P24= (155-82) mod 256
= 73 mod 256
= 73 (I)
P25= (157-73) mod 256
= 84 mod 256
= 84 (T)
P26= (135-66) mod 256
= 69 mod 25
=69 (E)

Nilai desimal *plainteks*:
80,65,75,32,73,77,65,77,32,68,79,83,69,78,32,84,69,82,70,65,86,79,82,73,84,69
Karateristik *plainteks*:
PAK IMAM DOSEN TERFAVORITE

4. KESIMPULAN

Berdasarkan pembahasan yang telah dilakukan dalam Pengamanan isi file docx menggunakan algoritma Gronsfeld adalah sebagai berikut:

1. Penerapan Algoritma Gronsfeld sangat mudah untuk diterapkan, karena hanya mengubah nilai *plainteks* menjadi nilai decimal kemudian ditambahkan dengan kunci.
2. Hasil dari data tersebut menjadi sulit untuk dibaca.

REFERENCES

[1] F. N. Pabokoy, I. F. Astuti and A. H. Kridalaksana, "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," Jurnal Informatika Mulawarman, vol. 10, p. 20, 2015.
[2] "kriptografi," 2013. [Online]. Available: https://yanuarkemal.blogspot.com.
[3] R. Munir, Kriptografi, Bandung: Informatika Bandung, 2006.
[4] muchad, "Penerapan Matriks Dalam Kriptografi," 2010. [Online]. Available: muchad.com.

ISBN: 978-602-52720-7-3
Februari 2020
Hal 415-419

[5] "TermasMedia," 24 july 2015. [Online]. Available: termasmedia.com.
[6] D. Apriadi, "Kriptografi Kunci Simetris Gronsfeld Cipher," 11 February 2016. [Online]. Available: dodi-apriadi.blogspot.com.
[7] "keamanan informasi," 2015. [Online]. Available: https://ernaparj.blogspot.com.
[8] Mesran, "Gronsfeld Cipher," 3 July 2011. [Online]. Available: mesran.wordpress.com.