Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

Implementasi Algoritma Massey-Omura dan Algoritma Elias Delcta Code Pada Pengamanan dan Kompresi File Dokumen

Rosalina Manalu

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: rossamanalu3005@gmail.com

Abstrak–Semakin pesatnya perkembangan teknologi informasi mengharuskan para pengguna untuk lebih berhati-hati dalam menjaga kerahasiaan dan keamanan data agar data tidak diketahui atau dimanipulasi oleh pihak lain. Pengamanan data merupakan hal yang sangat penting bagi para penguna, keamanan data suatu data dapat dijaga oleh setiap orang. Kriptografi salah satu alat keamanan yang digunakan untuk menyandikan suatu file dokumen. Salah satunya dengan menggunakan Algoritma Massey-Omura yang merupakan algoritma kriptografi asimetris, sehingga memiliki kunci yang berbeda untuk melakukan enkripsi dan dekripsi. Dalam proses pengriman data, juga dibutuhkan durasi pengiriman file untuk tidak memberi kesempatan kepada pencuri data yaitu dengan memampatkan data atau mereduksi ukuran file supaya lebih kecil tanpa menghilangkan isi dari file dokumen, sehingga waktu proses pertukaran data dapat lebih singkat. Kompresi data merupakan proses dimana file text, audio, maupun video ditransformasikan menjadi file terkompresi dengan ukuran data yang lebih kecil namun tidak kehilangan informasi yang sebenarnya. Penelitian ini membahas jenis kompresi lossless, sehingga file yang didapatkan dari hasil dekompresi akan identik dengan data asli. Elias Delta Code merupakan jenis kompresi lossless. Parameter kinerja dari algoritma ini akan diukur dengan Ratio of Compression (RC, Space Saving (SS) dan waktu kompresi. Data yang digunakan pada proses pengujian yaitu data yang terdiri dari beberapa jenis karakter. Pada pembahasan ini akan meliputi penjelasan, cara kerja, kelebihan dan kekurangannya. Penulis juga akan mencoba memberikan ulasan cara untuk meningkatkan tingkat keamanan file dokumen.

Kata Kunci: Kriptografi, Pengamanan, Kompresi, File, Massey-Omura, Elias Delcta Code.

Abstract–The rapid development of information technology requires users to be more careful in maintaining data confidentiality and security so that data is not known or manipulated by other parties. Data security is very important for users, data security can be maintained by everyone. Cryptography is one of the security tools used to encode a document file. One of them is by using the Massey-Omura algorithm, which is an asymmetric cryptographic algorithm, so that it has different keys for encryption and decryption. In the process of sending data, it also requires the duration of sending files to not give an opportunity to data thieves by compressing the data or reducing the size of the file to be smaller without losing the contents of the document file, so the data exchange process can be shorter Data compression is a process where text, audio, and video files are transformed into compressed files with smaller data sizes but do not lose the actual information. This study discusses the type of lossless compression, so that files obtained from decompression will be identical to the original data. Elias Delta Code is a type of lossless compression. The performance parameters of this algorithm will be measured by Ratio of Compression (RC, Space Saving (SS) and compression time. Data used in the testing process is data that consists of several types of characters. In this discussion will include explanation, how to work, advantages and The author will also try to provide a review of ways to improve the security level of document files.

Keywords: Cryptography, Security, Compression, Files, Massey-Omura, Elias Delcta Code.

1. PENDAHULUAN

Pengamanan data dan informasi merupakan kebutuhan penting yang harus diperhatikan dalam bidang komunikasi. Terlebih kemajuan teknologi informasi dewasa ini, memberikan dampak positif dan negatif bagi masyarakat seperti kemudahan memperoleh informasi, penyebaran informasi, pengiriman data dan pertukaran data. Media komunikasi umum yang digunakan saat ini sangat rawan terhadap penyadapan informasi oleh pihak-pihak yang tidak berkepentingan terhadap informasi tersebut.

Kriptografi adalah salah satu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek pengamanan informasi seperti kerahasiaan data, integritas data dan ontentikasi data. Dengan kriptografi data akan diubah menjadi sandi yang hanya dapat diartikan oleh pihak yang memiliki kunci yang akan digunakan untuk mengubah sandi menjadi data asli.

Algoritma Massey-Omura merupakan algoritma kriptografi kunci publik berbasis eksponensial yang diusulkan oleh James Massey dan Jim K. Omura pada 1982. Algoritma ini didasarkan pada ide Adi Shamir yaitu Shamir's three-pass protocol atau Shamir's no-keys protocol. Pada algoritma Massey-Omura ukuran file setelah enkripsi akan bertambah besar dan didasarkan pada konsep three-pass protocol. Pertukaran pesan akan dilakukan sebanyak tiga kali melalui three-pass protocol ini. Konsumsi transfer atau Bandwidth jaringan juga menjadi salah satu faktor yang mempengaruhi waktu proses pengiriman pesan karena ukuran Bandwidth setara dengan kecepatan waktu pengiriman. Hal ini dapat menyebabkan proses pengiriman menjadi lambat.

Reza et al. melakukan penelitian pengamanan file teks dengan melakukan simulasi pengamanan file dengan menggunakan algoritma Massey-Omura. Dari hasil penelitian diperoleh suatu sistem yang menggunakan algoritma Massey-Omura untuk pengamanan file di mana sistem ini dapat digunakan untuk mengenkripsi dan mendekripsi file teks [1]. Pengujian dengan menganalisis keamanan pada kombinasi protocol Secret Sharing dan Three-Pass. Dari hasil pengujian didapatkan bahwa kombinasi protokol Secret Sharing dan Three-Pass berhasil mengamankan pesan dan mengirimkan pesan tersebut kepada setiap penerimanya [2].

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

Kompresi data atau pemampatan data adalah sebuah cara untuk memadatkan data sehingga hanya memerlukan ruang penyimpanan lebih kecil dan lebih efisien dalam penyimpanan atau mempersingkat pertukaran data. Untuk mengoptimalkan proses pengiriman file diterapkan dengan melakukan kompresi file atau pemampatan data. Untuk itu digunakan sebuah algoritma kompresi data untuk mengecilkan ukuran file enkripsi tanpa ada data yang hilang atau identik dengan data asli.

Untuk proses kompresi ini digunakan algoritma kompresi Elias Delta Code yang merupakan jenis kompresi lossless. Kompresi lossless adalah metode kompresi data yang memungkinkan data asli dapat disusun kembali dari hasil kompresi maka rasio kompresi pun tidak terlalu besar untuk memastikan semua data dapat dikembalikan ke bentuk semula. Kompresi lossless utamanya digunakan untuk data artikel, text, dll. Parameter kinerja algoritma akan diukur dengan Ratio of Compression (RC, Space Saving (SS) dan waktu kompresi.

2. METODE PENELITIAN

2.1 Pengamanan

Pengamanan adalah proses atau cara mengamankan sesuatu yang bertujuan untuk mencegah ancaman atau gangguan. Dalam ilmu kriptografi, pengamanan sama halnya dengan penyandian teks. Teks rahasia akan disandikan dengan karakter yang berbeda, sehingga karakter tersebut tidak memiliki arti[3].

2.2 Kompresi

Teknik kompresi berarti proses memampatkan sesuatu yang berukuran besar sehingga menjadi kecil. Maka kompresi adalah proses untuk memampatkan atau mereduksi data agar ukurannya menjadi kecil. Pemampatan ukuran data/berkas melalui proses kompresinya diperlukann saat data tersebut disimpan dan dikirim melalui media transmisi atau telekomunikasi. Apabila data/berkas tersebut akan ditampilkan lagi pada layar monitor, maka data yang terkompresi tersebut harus dibongkar lagi dan dikembalikan pada format semula agar dapat dibaca kembali. Proses pembongkaran data yang dimampatkan ini disebut dikompresi. Satuan yang mencakup penting dalam kompresi data adalah *Compression ratio* yang menggambarkan seberapa besar ukuran data setelah proses kompresi dan dibandingkan dengan ukuran data/berkas semula[4].

2.3 Massey-Omura

Massey-Omura Cryptosystem adalah *cipher* berbasis eksponensial yang oleh James Messey dan Jim K. Omura pada 1982. Yang didasarkan pada Shamir's *three-pass protocol*. Keuntungan dari kriptosistem ini adalah tidak diperlukan distribusi dan pertukaran kunci diantara kedua pihak yang berkomunikasi.

Sebelum melakukan komunikasi dengan menggunakan *Massey-Omura Cryptosystem*, para koresponden harus memiliki kunci enkripsi dan dekripsi terlebih dahulu dengan syarat[1]:

1. Semua pengguna menyetujui kelompok batasan terhadap bidang tetap batasan *Fp* dengan p sebagai kekuatan utama.
2. Masing-masing pengguna sistem secara rahasia memilih sebuah bilangan bulat acak e_A atau e_B antara 1 dan p – 1 sebagai calon kunci enkripsi dengan syarat gcd(e_A,p – 1) = 1 dan gcd(e_A,p – 1)=1
3. Dengan menggunkan algortima euclidean, hitung invers e_A dan e_{AB}. Invers e_A yaitu $e_A \equiv e_A^{-1}(mod\ p-1)$ sebagai kunci dekripsi pengirim (d_A) dan invers $e_B$yaitu $e_B \equiv e_B^{-1}(mod\ p-1$ () sebagai kunci dekripsi penerima (d_B).

2.4 Elias Delta Code

Elias delta code merupakan kode universal yang mengkodekan bilangan bulat pusat yang dikembangkan oleh Peter Elias. Elias Delta Code merupakan satu dari tiga Elias Code yang dipelopori oleh Peter Elias. Di dalam kode ini, setiap karakter direpresentasikan dengan pemetaan kode sumber ke sejumlah variabel bit. Aturan untuk mengkodekan sebuah bilangan dengan menggunakan Elias Delta Code adalah sebagai berikut :

1. *Encoding*
   Berikut langkah-langkah *Encoding* pada algortima *Elias Delta Code* :
   a. Tulis bilangan dalam bentuk biner
   b. Hilangkan bit paling kiri dari bentuk bilangan biner
   c. Hitung sisa bit pada langkah ke-2 dan tambahkan 1 kemudian tuliskan hasilnya dalam bentk biner di depan bilangan pada langkah ke-2.
   d. Kemudian hitung jumlah bit pada langkah ke-3 dan kurangi 1 kemudian tulisakan jumlahnya dalam bentuk biner 0 lalu sisipkan pada awalan bilangan langkah ke-3.

2. *Decoding*
   Langkah-langkah mendecoding algoritma *Elias Delta Code* sebagai berikut :
   a. Baca bilangan biner 0 sampai ditemukan angka 1,
   b. Jumlah bilangan biner 0 dengan 1 atau $n' + 1$

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

c. Baca bit bilangan setelah biner 0 sesuai hasil yang didapatkan pada langkah ke-2. Lalu kurrangkan dengan 1 sehingga didapatkan n.
d. Dapatkan binagn encodenya dengan menjumlahkan $2\hat{}n \quad + x$, x merupakan bilangan sisa[2].

Tabel 1. Elias Delta Code

Number	N	N + 1	Encoding	Probability
$1 = 2^0 + 0$	0	1	1	1/2
$2 = 2^1 + 0$	1	2	0 1 0 0	1/16
$3 = 2^1 + 0$	1	2	0 1 0 1	1/16
$4 = 2^2 + 0$	2	3	0 1 1 00	1/32
$5 = 2^2 + 1$	2	3	0 1 1 01	1/32
$6 = 2^2 + 2$	2	3	0 1 1 10	1/32
$7 = 2^2 + 3$	2	3	0 1 1 11	1/32
$8 = 2^3 + 0$	2	4	00 1 00 000	1/256
$9 = 2^3 + 1$	3	4	00 1 00 001	1/256
$10 = 2^3 + 2$	3	4	00 1 00 010	1/256
$11 = 2^3 + 3$	3	4	00 1 00 011	1/256
$12 = 2^3 + 4$	3	4	00 1 00 100	1/256
$13 = 2^3 + 5$	3	4	00 1 00 101	1/256
$14 = 2^3 + 6$	3	4	00 1 00 110	1/256
$15 = 2^3 + 7$	3	4	00 1 00 111	1/256
$16 = 2^4 + 0$	4	5	00 1 01 0000	1/512
$17 = 2^4 + 1$	4	5	00 1 01 001	1/512

3. HASIL DAN PEMBAHASAN

Analisa sistem (*system analysis*) merupakan proses awal pembangunan sistem dengan melakukan identifikasi terhadap komponen-komponen yang dibutuhkan oleh sistem. Analisa dapat dilakukan dengan menentukan komponen yang menjadi input, proses sistem, dari output dari sistem. *Analisa masalah*merupakan proses identifikasi sebab akibat dibangunnya sebuah sistem agar sistem dapat berjalan sesuai dengan tujuan sistem. Pada penelitian ini perlunya pengamanan *file* yang berkaitan dengan aspek pengamanan *file* untuk menghindari bahaya seperti manipulasi isi *file* oleh pihak yang tidak bersangkutan. algoritma *Elias Delta Code*. *File* yang akan diamankan adalah berekstensi .doc. Sistem akan melakukan empat jenis proses yaitu proses enkripsi, kompresi dekompresi dan dekripsi. Pertukaran pesan akan dilakukan sebanyak tiga kali sesuai dengan konsep *Three-Pass-Protocol.*

3.1 Analisa Pengamanan Algoritma *Massey-Omura*

Berikut ini adalah cara kerja algoritma *Massey-Omura* dalam pengamanan *file* dokumen. Sebagai contoh, A hendak mengirim dokumen kepada B dengan nama *file* "LAPORAN ROSALINA". Keduanya menyetujui bilangan prima lebih besar dari 256, misalnya p = 257.

1. Pengirim memilih eA, dimana 1 < eA < p-1 dan eA koprima p-1.
   a. Misalkan eA = 151, karena 59 adalah koprima dengan 257. Nilai eA digunakan dalam proses enkripsi.
   b. Pengirim menghitung invers eA berdasarkan persamaan pertama (1), dan kemudian disimpan dengan dA = 39, jika dihitung (151 x 39) mod 256 = 1 memenuhi syarat. Nilai dA digunakan dalam proses enkripsi.
2. Penerima menghitung eB, dimana 1 < eB<p-1 dan eB koprima dengan p-1
   a. Misalkan eB = 25, karena 25 adalah koprima dengan 257. Nilai eB digunakan dalam proses enkripsi.
   b. Penerima menghitung invers eB berdasarkan persamaan ketiga (3), maka penyimpanan dalam dB = 41, jika dihitung (25 x 41) mod 256=1 memenuhi syarat. Nilai dB digunakan dalam proses dekripsi.

Proses enkripsi pesan

Nama dokumen yang hendak dikirim adalah "LAPORAN ROSALINA"

1. Shinta mengkonversi karakter pesan dengan bilai ASCII sebagai berikut:

L=74	A=65	P=80	O=79	R=82	A=65	N=78	Spasi=32
R=82	O=79	S=83	A=65	L=76	I=73	N=79	A=65

2. Enkripsi setiap karakter pesan berdasarkan persamaan kedua (2).
   $C2 = C^{eA} \bmod p$
   eA = 151
   p = 257
   Untuk L : $76^{151} \bmod 257 = 220$

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

Untuk A : 65^{151} mod 257 = 80
Untuk P : 80^{151} mod 257 = 251
Untuk O : 79^{151} mod 257 = 104
Untuk R : 82^{151} mod 257 = 14
Untuk A : 65^{151} mod 257 = 80
Untuk N : 78^{151} mod 257 = 94
Untuk SPACE = 32^{151} mod = 8
Untuk R : 82^{151} mod 257 = 14
Untuk O : 79^{151} mod 257 = 104
Untuk S : 83^{151} mod 257 = 204
Untuk A : 65^{151} mod 257 = 80
Untuk L : 76^{151} mod 257 = 220
Untuk I : 73^{151} mod 257 = 22
Untuk N : 78^{151} mod 257 = 94
Untuk A : 65^{151} mod 257 = 80

Jadi hasil C1 =

220	80	251	104	14	80	94	8
14	104	204	80	220	22	94	80

3. Jika dikonversi ke karakter maka hasilnya adalah :

Ü	P	Û	H		P	^	
	H	Ì	P	Ü		^	P

4. Pengirim mengrim C1 kepada penerima
5. Penerima mengenkripsi C berdasarkan persamaan
   C2 = $C1^{eB}$ mod p
   eB = 25
   p = 257
   Untuk Ü : 220^{25} mod 257 = 149
   Untuk P : 80^{25} mod 257 = 107
   Untuk û : 251^{25} mod 257 = 154
   Untuk h : 104^{25} mod 257 = 13
   Untuk : 14^{25} mod 257 = 238
   Untuk P : 80^{25} mod 257 = 107
   Untuk ^ = 94^{25} mod 257 = 125
   Untuk : 8^{25} mod 257 = 249
   Untuk : 14^{25} mod 257 = 238
   Untuk h : 104^{25} mod 257 = 13
   Untuk Ì : 204^{25} mod 257 = 6
   Untuk P : 80^{25} mod 257 = 107
   Untuk Ü : 220^{25} mod 257 =149
   Untuk : 22^{25} mod 257 = 169
   Untuk ^ : 94^{25} mod 257 = 125
   Untuk P : 80^{25} mod 257 = 107

   Jadi hasil C2 :

149	107	154	13	238	107	125	249
238	13	6	107	149	169	125	107

6. Penerima mengiri C2 ke pengirim
7. Pengirim mengenkrip C2 berdasarkan persaman (5) dan menghasilkan C3
   C3 = $C2^{dA}$ mod p
   eB = 39
   p = 25
   Untuk 149 ^{39} mod 257 = 145
   Untuk 107^{39} mod 257 = 212
   Untuk 154 ^{39} mod 257 = 107
   Untuk 13 ^{39} mod 257 = 139
   Untuk 238^{39} mod 257 = 183
   Untuk 107^{39} mod 257 = 212
   Untuk 125^{39} mod 257 = 186
   Untuk 249^{39} mod 257 = 225
   Untuk 238^{39} mod 257 = 183

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

Untuk 13^{39} mod 257 = 139
Untuk 6^{39} mod 257 = 177
Untuk 107^{39} mod 257 = 212
Untuk 149^{39} mod 257 = 145
Untuk 169^{39} mod 257 = 46
Untuk 125^{39} mod 257 = 186
Untuk 107^{39} mod 257 =212
Jadi hasil C3 adalah :

145	212	107	139	183	212	186	225
183	139	177	212	145	46	186	212

Dikonversi kepada karakter, maka hasilnya adalah :

‘	Ô	K	‹	´	Ô	º	Þ
·	‹	±	Ô	‘	.	º	Ô

Diman C3 adalah cipher dari pesa asli dari pengirim ke penerima.

Proses Dekripsi file

1. Cipher yang telah diterima oleh penerima adalah :

45	212	107	139	183	212	186	225
183	139	177	212	145	46	186	212

‘	Ô	K	‹	´	Ô	º	Þ
·	‹	±	Ô	‘	.	º	Ô

M = $C3^{dB}$ mod p
eB = 41
p = 257
Maka pesan yang diterima adalah :
Untuk 145^{41} mod 257 = 76
Untuk 212^{41} mod 257 = 65
Untuk 107^{41} mod 257 = 80
Untuk 139^{41} mod 257 = 79
Untuk 183^{41} mod 257 = 82
Untuk 212^{41} mod 257 = 65
Untuk 186^{41} mod 257 = 78
Untuk 225^{41} mod 257 = 32
Untuk 183^{41} mod 257 = 82
Untuk 139^{41} mod 257 = 79
Untuk 177^{41} mod 257 = 83
Untuk 212^{41} mod 257 = 65
Untuk 145^{41} mod 257 = 76
Untuk 46^{41} mod 257 = 73
Untuk 186^{41} mod 257 = 78
Untuk 212^{41} mod 257 = 65
Maka hasilnya adalah:

L=74	A=65	P=80	O=79	R=82	A=65	N=78	Spasi=32
R=82	O=79	S=83	A=65	L=76	I=73	N=79	A=65

Hasil dekripsi diatas dikonversi ke dalam karakter sebgai pesan asli yang diterima oleh penerima yaitu : LAPORAN ROSALINA.

3.2 Analisa Kompresi Algoritma *Elias Delta Code*

Berikut ini analisa perhitungan untuk mengkompresi *File* Dokumen dengan *Elias Delta Code*.

1. Langkah *Encoding*
   a. Contoh pada bilangan 20 → 10100
   b. Hilangkan bilangan bit paling kiri → 0100
   c. Sisa bit pada langkah ke- 2 + 1 → 4 + 1 = 5, dalam bentuk biner ditulis 101, maka bilangan sementara 101|0100
   d. Jumlah bit pada bilangan 101 adalah 3, maka 3 -1 = 2, sehingga bilangan biner dari langkah ini adalah 00. Dimasukkan pada bilangan langkah ke-3, sehingga hasilnya adalah 00|101|0100 atau 001010100.
2. Langkah *Decoding*

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

Contoh pada langkah encoding dari algoritma ini dapat diterapkan pada bilangan yang telah di decode sebelumnya yaitu : 001010100.

1. Baca bilangan 001010100, maka pada bilangan 0 yaitu 2 buah.
2. Jumlahkan N' + 1 = 2 + 1 = 3.
3. Selanjutnya adalah membuka 3 bit selanjutnya dari bilangan 0 yaitu 101 = 5 lalu dikurangkan dengan 1 maka N = 5 – 1 = 4
4. Maka bilangan encodenya adalah (2 ^ 4) + 0 = 16

Tabel 2. Tabel *Elias Delta Code*

Number	N	N + 1	Encoding	Probability
$1 = 2^0 + 0$	0	1	1	1/2
$2 = 2^1 + 0$	1	2	0 1 0 0	1/16
$3 = 2^1 + 0$	1	2	0 1 0 1	1/16
$4 = 2^2 + 0$	2	3	0 1 1 00	1/32
$5 = 2^2 + 1$	2	3	0 1 1 01	1/32
$6 = 2^2 + 2$	2	3	0 1 1 10	1/32
$7 = 2^2 + 3$	2	3	0 1 1 11	1/32
$8 = 2^3 + 0$	2	4	00 1 00 000	1/256
$9 = 2^3 + 1$	3	4	00 1 00 001	1/256
$10 = 2^3 + 2$	3	4	00 1 00 010	1/256
$11 = 2^3 + 3$	3	4	00 1 00 011	1/256
$12 = 2^3 + 4$	3	4	00 1 00 100	1/256
$13 = 2^3 + 5$	3	4	00 1 00 101	1/256
$14 = 2^3 + 6$	3	4	00 1 00 110	1/256
$15 = 2^3 + 7$	3	4	00 1 00 111	1/256
$16 = 2^4 + 0$	4	5	00 1 01 0000	1/512
$17 = 2^4 + 1$	4	5	00 1 01 001	1/512
$18 = 2^4 + 2$	4	5	00 1 01 0010	1/512

Selanjutnya pengujian kompresi dilakukan dengan input string LAPORAN ROSALINA meliputi tahapan sebagai berikut :

1. Hitung panjang teks
2. Buat dan hitung panjang himpunan karakter
3. Hitung panjang bit
4. Menentukan Padding bit.

Di bawah ini adalah tabel menunjukkan string asli sebelum proses kompresi. String memiliki 16 karakter.

Tabel 2. String Asli

No.	Char	ASCII	Binary
1	L	76	1001100
2	A	65	1000001
3	P	80	1010000
4	O	79	1001111
5	R	82	1010010
6	A	65	1000001
7	N	78	1001110
8	Spasi	32	100000
9	R	82	1010010
10	O	79	1001111
11	S	83	1010011
12	A	65	1000001
13	L	76	1001100
14	I	73	1001001
15	N	78	1001110
16	A	65	1000001

Kemudian pada tahap selanjutnya karakter berulang akan dihilangkan sehingga hanya tinggal satu karakter. Setelah proses ini, hanya tersisa 9 karakter dengan frekuensi yang berbeda.

Tabel 3. Frekuensi Karakter *Elias Delta Code*

No.	Char	ASCII	Binary	Frekuensi	Jumlah Bit

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

1	L	76	1001100	2	16
2	A	65	1000001	4	32
3	P	80	1010000	1	8
4	O	79	1001111	2	16
5	R	82	1010010	2	16
6	N	78	1001110	2	16
7		32	100000	1	8
8	S	83	1010011	1	8
9	I	73	1001001	1	8
				16	128

Langkah selanjutnya adalah mengurutkan nilai frekuensi dan mengkonversi nilai *Elias Delta Code* :

Tabel 4. Pengurutan dan Konversi *Elias Delta Code*

No.	Char	Frekuensi	Elias Delta Code	Jumlah Bit	Total Bit
1	A	4	1	1	4
2	L	2	0100	4	8
3	O	2	0101	4	8
4	R	2	01100	5	10
5	N	2	01101	5	10
6	P	1	01110	5	5
7	S	1	01111	5	5
8	I	1	00100000	8	8
9	Sp	1	00100010	8	8
					66

Adapun parameter algoritma kompresi

1. *Ratio Of Compression (Rc)*

$$\text{Rc} = \frac{bit\ sebelum\ kompresi}{bit\ setelah\ kompresi}$$

$$\text{Rc} = \frac{128}{66}$$

$$\text{Rc} = 1.93$$

2. *Compression Ratio*

$$\text{C} = \frac{bit\ setelah\ kompresi}{bit\ sebelum\ kompresi}\ x\ 100\%$$

$$\text{Cr} = \frac{66}{128}\ x\ 100\%$$

$$\text{Cr} = 51.5625\%$$

3. *Redundancy*

$$Rd = \frac{bit\ sebelum\ kompresi - \ bit\ setelah\ kompresi}{bit\ sebelum\ kompresi}\ x\ 100\%$$

$$= \frac{128 - 66}{128}\ x\ 100\%$$

$$= \frac{62}{128}\ x\ 100\%$$

$$= \mathbf{48.4375}\%$$

Proses kompresi telah menyimpan data adalah **48.4375%** dari data asli. tingkat penyimpanan tergantung pada urutan dan pola karakter dari pesan asli.

4. KESIMPULAN

Berdasarkan studi literatur, analisa dan implementasi sistem algoritma Massey-Omura dan Elias Delta Code dalam pengamanan dan kompresi file dokumen dapat disimplkan berikut ini :

1. Pengamanan dan kompresi file dokumen dengan konsep Three-pass protocol memenuhi syarat keutuhan file dokumen. Dalam pengamanan pertukaran file dengan menggunakan algoritma Massey-Omura sistem ini melakukan enkripsi dengan menggunakan bilangan prima dan koprima dengan kunci enkripsi yang bersifat rahasia.
2. Kompresi file menggunakan algoritma Elias Delta Code tergantung pada parameter. Semakin besar nilai dari Parameter Compression Ratio, Ratio Compression, Redundancy, dan Space Saving maka semakin kompresi..

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2152

REFERENCES

[1] Reza, M., Budiman, M. A. & Arisandi, D. Simulasi Pengamanan File Teks Menggunakan Algoritma Massey-Omura.Jurnal Dunia Teknologi Informasi 1(1): 20-27, 2012
[2] M Katz, J. & Lindell, Y. Introduction to modern cryptography (2nd ed., Chapman & Hall/CRC Cryptography and Network Security Series). Chapman and Hall/CRC. 2014
[3] B Julita, Perbandingan dan Anlisis Algoritma Elias Delta Code dan Algortima Unary Coding Dalam Mengkompresi File Text.Skripsi.USU.2017
[4] Rachmat Antonius, Algoritma dan Pemograman dengan bahasa, Andy:2010
[5] Alfredo S. Implementasi Algoritma Massey-Omura Dalam Pengamanan Teks. Skripsi. STMIK Budidarma: 2015
[6] Https://en.wikipedia.org/File.2018
[7] Sedianingsih. Teori dan Praktik Administrasi Kesekretariatan. Andy.Yogyakarta:2015.
[8] Dony Ariyus, Pengantar Ilmu Kriptografi Teori Analisis dan Impementasi, Yogyakarta , Andy Offset:2008
[9] Rifki Sadikin, Kriptografi untuk Keamanan Jaringan dan Implementasinya Dalam Bahasa Java, Yogyakarta: Andy: 2012
[10] Haramaini, T. Algoritma One Time Pad pada Three-Pass Protocol. Tesis. Universitas Sumatra Utara.2014Khavita, Kompresi Data Lossless, Skripsi.USU:2016
[11] A.S. Rosa dan Sahaludin. Model Pembelajaran Rekayasa Perangkat Lunak Terstruktur Dan Beriorientasi Object. Bandung: Modula: 2011
[12] P.D.Dr.Suarga.M.Sc.M.Math, ALgoritma dan Pemograman, Yogyakarta: Andy Offset:2012

Submitted: **04/05/2020**; Accepted: **13/05/2020**; Published: **20/05/2020**