Paper

Analisis Kinerja Sistem Kripto kompresi Pada File Dokumen Dengan Algoritma Asimetris RSA dan Even Rodeh Code

Author: Imam Prayoghi, Imran Lubis, Haida Dafitri

SNASTIKOM 10
Seminar Nasional Teknologi Informasi dan Komunikasi
UNIVERSITAS HARAPAN MEDAN 2012-2023
Program Studi Teknik Informatika
TEMA : DUNIA AKADEMIK DI ERA SOCIETY 5.0 : TANTANGAN DAN PELUANG
ISSN : 2964-1950

ISSN: 2964-1950

Analisis Kinerja Sistem Kripto kompresi Pada File Dokumen Dengan Algoritma Asimetris RSA dan Even Rodeh Code

Imam Prayoghi[1], Imran Lubis[2], Haida Dafitri[3*]
[1,2,3]Fakultas Teknik dan Komputer, Program Studi Teknik Informatika, Universitas Harapan Medan
Jalan. H.M. Joni No 70 C, Medan, Indonesia

[1]Imamyogi013@gmail.com, [2]imran.loebis.medan@gmail.com, [3]aida.stth@gmail.com

Abstrak

File dokumen merupakan data elektronik yang sering digunakan untuk menyimpan informasi yang sifatnya pribadi atau rahasia, oleh karena itu perlu dilakukan tindakan untuk tetap menjaga informasi tersebut hanya dapat diakses oleh pemilik data. Selain aspek keamanan, maka hal yang perlu diperhatikan juga adalah tentang memori penyimpanan. Oleh karena itu, diperlukan langkah tambahan untuk mengefisiensikan media penyimpanan serta mempercepat proses transmisi dengan melakukan kompresi terlebih dahulu supaya ukurannya menjadi lebih kecil. Dalam penelitian ini, akan menggabungkan teknik kriptografi dan kompresi menggunakan algoritma RSA untuk tujuan mengamankan data, lalu dikompresi menggunakan algoritma Even Rodeh Code guna memperkecil ukuran data. Penelitian ini menghasilkan sebuah aplikasi yang dapat dijadikan sebagai alternatif solusi dalam menjaga kerahasiaan data file dokumen sehingga hanya dapat diakses oleh pemilik data dan dapat menghemat kebutuhan akan ruang penyimpanan (storage) data menjadi lebih efisian. Dari hasil pengujian menunjukkan bahwa size file uji yang berbeda menghasilkan nilai perhitungan Compression Ratio (CR), Space Savings (SS) dan RT (Running Time) yang berbeda juga. Semakin besar size yang akan dikompresi maka semakin besar juga size yang dapat diperkecil ukurannya atau semakin besar ruang penyimpanan yang dapat di hemat. Sementara hasil RT mengalami peningkatan nilai secara konstan seiring dengan perubahan size file uji yang digunakan. Artinya semakin besar size file uji yang akan dikompresi maka semakin lama juga waktu yang dibutuhkan untuk mengkompresinya.

Kata Kunci: Kriptografi, Kompresi, Kripto kompresi, RSA, Even Rodeh Code

Abstract

The document files are electronic data that are often used to store information that is personal or confidential, therefore it is necessary to take action to keep the information accessible only to the owner of the data. In addition to the security aspect, the thing that needs to be considered is also about storage memory. Therefore, additional steps are needed to streamline the storage media and speed up the transmission process by compressing it first so that its size becomes smaller. In this study, will combine cryptography and compression techniques using the RSA algorithm for the purpose of securing data, then compressed using the Even Rodeh Code algorithm to reduce the size of the data. This research produces an application that can be used as an alternative solution in maintaining the confidentiality of document file data so that it can only be accessed by the data owner and can save the need for data storage space to be more efficient. The test results show that different test file sizes produce different Compression Ratio (CR), Space Savings (SS) and RT (Running Time) calculation values as well. The larger the size to be compressed, the larger the size that can be reduced or the greater the storage space that can be saved. While the RT results have increased in value constantly along with changes in the size of the test file used. This means that the larger the size of the test file to be compressed, the longer it will take to compress it.

Keywords: Cryptography, Compression, Cryptocompression, RSA, Even Rodeh Code

1. PENDAHULUAN

Kemajuan teknologi di era globalisasi saat ini semakin memudahkan manusia dalam berkomunikasi dan bertukar informasi. Kemajuan teknologi informasi memberikan banyak keuntungan bagi keberlangsungan kehidupan manusia, tetapi keuntungan yang ditawarkan juga menimbulkan kejahatan seperti pencurian data. File dokumen dengan format .docx dan .pdf merupakan data elektronik yang sering digunakan untuk menyimpan informasi yang sifatnya pribadi atau rahasia, oleh karena itu perlu dilakukan tindakan untuk tetap menjaga informasi tersebut hanya dapat diakses oleh pemilik data. Berbagai cara dilakukan untuk menjaga keamanan data elektronik seperti menyembunyikan data tersebut maupun penyandian data menjadi suatu kode-kode yang tidak

dimengerti, sehingga apabila disadap akan kesulitan untuk mengetahui dan memahami informasi yang sebenarnya.

Kriptografii merupakan bidang ilmu pengetahuan yang menggunakan persamaan matematis untuk melakukan proses enkripsi dan dekripsi [1] yang bertujuan untuk menjaga agar informasi tersebut tidak dapat dibaca oleh pihak-pihak yang tidak berwenang [2]. Terdapat banyak algoritma kriptografi yang masing-masing mempunyai kelebihan dan kelemahannya tersendiri. Pada penelitan ini mengggunakan algoritma RSA (Rivest Shamir Adleman) yang merupakan jenis kriptografi asimetris dengan menggunakan dua buah kunci (publik dan privat) untuk proses enkripsi dan dekripsi. Algoritma kriptografi asimetris lebih kuat keamanannya dibanding dengan algoritma simetris. Keuntungan utama kunci enkripsi asimetris adalah memiliki enkripsi yang kuat yang akan membuat dekripsi teks asli menjadi sulit dan tidak dapat diprediksi [3]. Mekanisme operasi algoritma RSA sangat mudah dipahami dan sederhana, tetapi kuat [2], sehingga algoritma RSA dinilai tepat diterapkan dalam penelitian ini untuk menjaga kerahasiaan data pada file dokumen.

Selain aspek keamanan, maka hal yang perlu diperhatikan juga adalah tentang efisiensi media penyimpanan. Pesatnya perkembangan teknologi yang serba digital saat ini, pertukaran data dapat dilakukan secara nirkabel melalui media digital dimana saja dan kapan saja, hal ini mengharuskan pengguna untuk memiliki media penyimpanan seperti harddisk yang memadai dan waktu pengiriman yang singkat. Semakin besar ukuran data maka media penyimpanan yang diperlukan juga semakin banyak serta semakin lama juga waktu yang dibutuhkan jika akan ditransmisikan. Oleh karena itu, diperlukan langkah tambahan untuk mengefisiensikan media penyimpanan serta mempercepat proses transmisi dengan melakukan kompresi terlebih dahulu supaya ukurannya menjadi lebih kecil. Kompresi adalah proses mengubah sekumpulan data menjadi bentuk kode untuk menghemat kebutuhan ruang penyimpanan dan waktu untuk transmisi data. Dengan kata lain, menggunakan kompresi data, ukuran file tertentu dapat dikurangi. Data dapat berupa karakter dalam file teks [4]. Pada penelitan ini mengggunakan algoritma Even Rodeh Code yang mengkodekan setiap karakter menggunakan beberapa rangkaian bit [5]. Algoritma Even Rodeh Code merupakan salah satu algoritma kompresi data berjenis lossless compression, yaitu suatu metode kompresi data yang digunakan untuk mengurangi ukuran data dan dapat mengembalikan data hasil kompresi ke data semula tanpa kehilangan informasi apapun dari data asli [6].

Dalam penelitian terdahulu, penulis mengambil beberapa referensi yang berkaitan dengan latar belakang masalah dalam penelitian ini. Penelitian-penelitian terdahulu akan dijadikan sebagai referensi yang bertujuan untuk mendapatkan bahan perbandingan dan acuan dalam penelitian ini yang dapat diambil dari jurnal penelitian. Penelitian yang berkaitan algoritma RSA yang pernah dilakukan oleh [1] menjelaskan bahwa algoritma RSA dapat merubah isi file asli (plaintext) menjadi karakter yang sulit dipahami (chipertext) sehingga file terjaga keamanannya. Penelitian yang berkaitan dengan algoritma Even Rodeh Code yang pernah dilakukan oleh [7] menjelaskan bahwa algoritma Even Rodeh Code (50%) lebih baik dari algoritma Subexponential Code (42%). Nilai rasio kompresi dipengaruhi oleh isi dari file yang dikompresi. Semakin banyak pengulangan karakter pada suatu file yang dikompresi maka semakin tinggi pula rasio kompresinya. Pada penelitian yang dilakukan oleh [8], jika dibandingkan dari segi urutan proses algoritma antara enkripsi-kompresi dan kompresi-enkripsi, maka yang memberikan hasil lebih efisien adalah urutan enkripsi-kompresi, hal ini dikarenakan ukuran file hasil proses bergantung dari proses kompresi bukan proses enkripsi yang dalam hal ini malah memperbesar ukuran data.

Berdasarkan permasalahan yang telah telah diuraikan serta penjelasan dari hasil penelitian terdahulu, maka pembaruan yang dilakukan dalam penelitian ini adalah terletak pada konsep dan algoritma yang digunakan dengan mengkombinasikan kriptografi algoritma asimetris RSA dan kompresi algoritma Even Rodeh Code kedalam sistem kripto kompresi. Kripto kompresi merupakan teknik yang memanfaatkan dua metode kedalam satu proses, artinya file dokumen akan dienkripsi terlebih dahulu dengan algoritma RSA untuk tujuan mengamankan data, lalu dikompresi menggunakan algoritma Even Rodeh Code guna memperkecil ukuran data. Penelitian ini diberi judul “Analisis Kinerja Sistem Kripto kompresi Pada File Dokumen Dengan Algoritma Asimetris RSA dan Even Rodeh Code”.

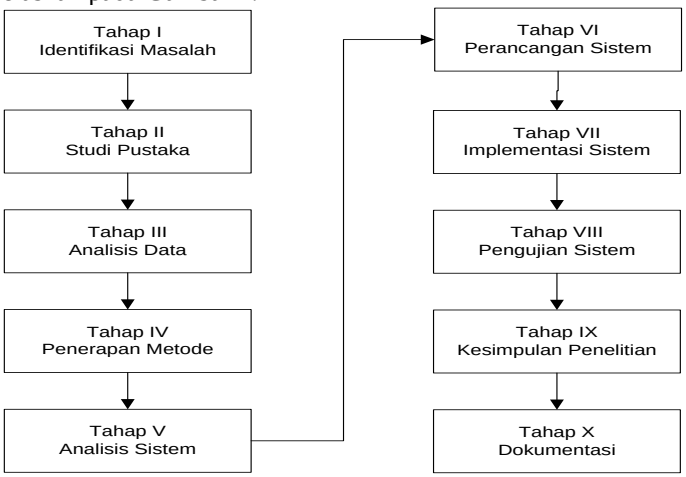
2. METODE PENELITIAN

2.1 Tahapan Penelitian

Pada penelitian ini akan dibahas mengenai analisis kinerja sistem kripto kompresi pada file dokumen dengan menerapkan algoritma kriptografi asimetris RSA dan algoritma kompresi Even Rodeh Code. Sistem kripto kompresi yang dibangun pada penelitian ini bertujuan untuk mengamankan isi berupa teks yang terdapat pada file dokumen dengan menggunakan algoritma kriptografi asimetris RSA serta memperkecil ukurannya dengan menggunakan algoritma kompresi Even Rodeh Code. Hasil dari proses enkripsi dan kompresi kemudian akan dianalisis menggunakan parameter Compression Ratio (CR), Space Savings (SS) dan serta RT (Running Time)

ISSN: 2964-1950

atau waktu enkripsi dan kompresi untuk tujuan menguji kinerja dari sistem kripto kompresi. Adapun kerangka kerja pada penelitian ini terdiri dari sembilan tahapan, yaitu identifikasi masalah, studi pustaka, analisis data, penerapan metode, kinerja metode, perancangan sistem, implementasi sistem, pengujian sistem, dan kesimpulan penelitian yang dapat dijelaskan pada Gambar 1.



Gambar 1. Tahapan Penelitian

Tahapan dalam penelitian pada Gambar 1 dimulai dengan identifikasi masalah, studi pustaka, analisis data, penerapan metode, kinerja metode, perancangan sistem, implementasi sistem, pengujian sistem, dan kesimpulan penelitian.

2.2 Kriptografi Algoritma RSA

Kriptografi didefiniskan sebagai suatu studi teknik matematika yang berhubungan dengan aspek keamanan informasi seperi kerahasiaan, integritas data, otentikasi entitas dan otentikasi keaslian data [9]. Sistem kriptografi (cryptosystem) sering disebut juga dengan sistem cipher (cipher system) adalah sistem yang terdiri dari algoritma enkripsi, algoritma dekripsi dan tiga komponen teks (plaintext, ciphertext dan kunci) [10]. Hal yang harus dicapai dalam penerapan algoritma kriptografi adalah confusion (pembingungan), yaitu harus mampu mempersulit pihak lain dalam merekonstruksi ulang cipher yang dihasilkan serta diffusion (peleburan) yaitu harus mampu menyembunyikan pola dari pesan asli [11]. Kriptografi berdasarkan jenis kuncinya dibedakan menjadi dua yaitu kriptografi simetris dan asimetris [3]. Pada kriptografi simetris, proses enkripsi dan dekripsi dilakukan menggunakan kunci rahasia yang sama. Sedangkan pada kriptografi asimetris, proses enkripsi dan dekripsinya menggunakan kunci yang berbeda, yaitu kunci publik untuk enkripsi, dan kunci rahasia yang digunakan untuk dekripsi [12].

Algoritma kriptografi RSA merupakan algoritma kriptografi kunci publik (asimetris) [8], dimana kunci enkripsi tidak sama dengan kunci dekripsinya. Algoritma kriptografi RSA terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi.

1.Proses pembentukan kunci algoritma RSA

Hal pertama yang dilakukan dalam pembangkit kunci adalah dengan membangkitkan 2 bilangan prima besar yang penyelesaiannya [10] dapat dijelaskan yaitu sebagai berikut:

a. Pilih dua bilangan prima sembarang, p dan q. Nilai p dan q harus dirahasiakan.
b. Hitung nilai n dengan menggunakan rumus $n = p * q$
c. Hitung nilai totient atau $\varphi(n)$ dengan menggunakan rumus $\varphi(n) = (p-1)(q-1)$
d. Pilih sebuah bilangan bulat sebagai kunci publik, disebut namanya e, yang relatif prima terhadap $\varphi(n)$. Nilai e relatif prima terhadap $\varphi(n)$ artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut $gcd(e, \varphi(n)) = 1$
e. Hitung kunci privat, disebut namanya d sedemikian agar $d * e \, mod \, \varphi(n) = 1$
f. Pasangan kunci enkripsi dan dekripsi dari proses pembentukan kunci algoritma RSA di atas adalah kunci enkripsi (public key) adalah pasangan (e, n) dan kunci dekripsi (private key) adalah pasangan (d, n).

2.Proses enkripsi algoritma RSA

Pengamanan data berdasarkan algoritma teknik kriptografi dilakukan dengan merubah pesan yang akan dirahasiakan (plaintext) menjadi sandi (ciphertext). Setiap blok plaintext (m_i) akan dienkripsi menjadi blok ciphertext (c_i) dengan persamaan (1) berikut:

ISSN: 2964-1950

$$c_i = m_i^e \bmod n \quad (1)$$

3.Proses dekripsi algoritma RSA

Proses untuk mengkonversi ciphertext menjadi plaintext disebut dengan proses dekripsi dengan persamaan (2) berikut:

$$m_i = c_i^d \bmod n \quad (2)$$

2.2 Kompresi Algoritma Even Rodeh Code

Kompresi data merupakan cabang ilmu komputer yang bersumber dari teori informasi. Teori informasi memfokuskan pada berbagai metode tentang informasi termasuk penyimpanan dan pemrosesan pesan. Teori informasi mempelajari pula tentang redundancy (informasi tidak berguna) pada pesan. Semakin banyak redundancy semakin besar pula ukuran pesan dan upaya mengurangi redundancy inilah yang akhirnya melahirkan subjek ilmu tentang kompresi data [8]. Pada kompresi data terdapat dua buah tipe teknik kompresi yang pertama teknik lossless dan yang kedua adalah teknik lossy [13]. Perbedaan utama antara kompresi lossy dan lossless terletak pada kualitasnya. Pada kompresi terdapat beberapa faktor penting yang perlu diperhatikan sebagai bahan pertimbangan untuk mengukur kualitas dari suatu metode kompresi, serta mendapatkan hasil perbandingan dari metode yang diuji yaitu Compression Ratio (CR), Space Savings (SS), dan RT (Running Time) waktu enkripsi kompresi yang dapat dijelaskan yaitu sebagai berikut:

1.Compression Ratio (CR)

Compression Ratio atau Rasio kompresi adalah menghitung kinerja dari repersentase data yang sudah dikompresi dan sebelum dikompresi. Adapun formula yang digunakan untuk mencari nilai Compression Ratio yaitu dengan menggunakan persamaan (3).

$$CR = \frac{Compressed\ bits}{Uncompressed\ bits} \quad (3)$$

2.Space Savings (SS)

Space Savings adalah persentase penghematan ruang (memori) setelah file dikompresi dengan mencari persentase selisih antara data awal sebelum dikompresi dengan hasil data yang telah dikompresi. Adapun formula yang digunakan untuk mencari nilai Space Savings yaitu dengan menggunakan persamaan (4).

$$SS = \left(1 - \frac{Compressed\ bits}{Uncompressed\ bits}\right) x\ 100 \quad (4)$$

3.Running Time (RT)

Running Time waktu adalah lama waktu yang dibutuhkan untuk melakukan proses kompresi dan dekompresi dari mulai pembacaan data hingga proses encoding pada data tersebut.

Even Rodeh Code adalah salah satu algoritma kompresi yang dikembangkan oleh Shimon Even dan Michael Rodeh pada tahun 1978 [7]. Kode Even Rodeh hampir sama dengan kode Omega, perbedaannya adalah bahwa panjang kode ditopang sampai panjang 3-bit tercapai dan menjadi grup kode paling kiri [14]. Algoritma Even Rodeh Code ini bersifat lossless. Berikut ini adalah tahap membangun kode Even Rodeh Code dengan n sebagai indeks dari karakter [7] yaitu sebagai berikut:

1. Menghitung panjang bit.
2. Jika panjang bit $0 \leq n \leq 3$ maka nilai n diubah ke biner, tambahkan 0 didepan nilai biner sehingga bit menjadi 3 digit.
3. Jika panjang bit $4 \leq n \leq 7$ maka nilai n diubah ke biner, tambahkan 0 dibelakang nilai biner sehingga bit menjadi 4 digit.
4. Jika panjang bit $n \geq 8$ maka nilai n diubah ke biner, tambahkan 0 dibelakang nilai biner kemudian angka ditambahkan didepan nilai biner sebanyak jumlah digit nilai dalam biner.

Berikut ini adalah urutan kode Even Rodeh Code dan jumlah bit berdasarkan nilai n dari panjang karakter [14] bisa dilihat pada Tabel 1.

Tabel 1. Kode Algoritma Even Rodeh Code

N	Even Rodeh Code
0	000
1	001
2	010
3	011
4	100 0
7	111 0

ISSN: 2964-1950

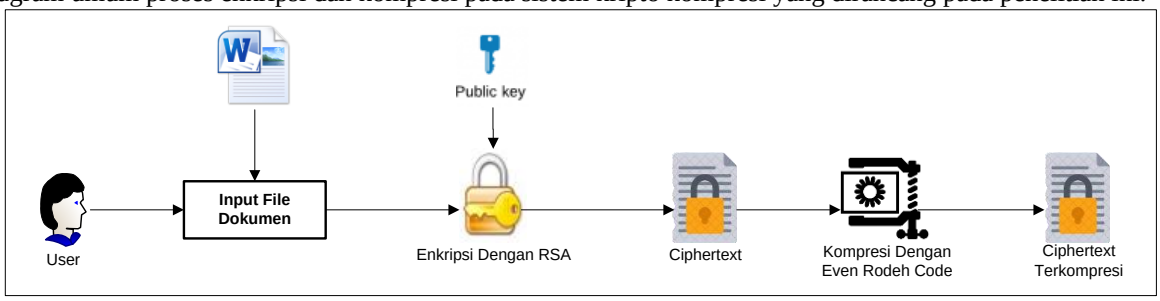
8	100 1000 0
15	100 1111 0
16	100 10000 0
32	110 100000 0
100	111 1100100 0
1000	110 1100100 0

3. HASIL DAN PEMBAHASAN

Sistem kripto kompresi yang dibangun pada penelitian ini bertujuan untuk mengamankan isi berupa teks pada file dokumen dengan menggunakan algoritma kriptografi asimetris RSA serta memperkecil ukurannya dengan menggunakan algoritma kompresi Even Rodeh Code. Cara kerja sistem bermula dari teks (plaintext) yang terdapat dalam file dokuemen akan dienkripsi menggunakan algoritma kriptografi asimetris RSA yang akan menghasilkan teks terenkripsi (ciphertext). Lalu ciphertext akan diperkecil ukurannya dengan melakukan proses kompresi menggunakan algorima Even Rodeh Code yang akan menghasilkan ciphertext terkompresi. Hasil dari proses enkripsi dan kompresi kemudian akan dianalisis menggunakan parameter Compression Ratio (CR), Space Savings (SS) dan serta RT (Running Time) atau waktu enkripsi dan kompresi untuk tujuan menguji kinerja dari sistem kripto kompresi. Sedangkan untuk mengembalikan file dokumen yang telah di enkripsi dan dikompresi, maka tahapan yang harus dilakukan dimulai dengan melakukan dekompresi terlebih dahulu dengan menggunakan algoritma Even Rodeh Code kemudian dilakukan proses dekripsi dengan menggunakan algoritma RSA yang nantinya menghasilkan plaintext yang dapat dibaca dan dipahami maknanya.

3.1 Analisis Proses Enkripsi dan Kompresi

Proses enkripsi dan kompresi dalam skema sistem kripto kompresi dimulai dengan mengenkripsi teks yang terdapat pada file dokumen menggunakan algoritma kriptografi asimetris RSA. Proses enkripsi merupakan tahap untuk mentransformasi data ke dalam bentuk yang tidak dimengerti oleh banyak orang. Gambar 2 merupakan diagram umum proses enkripsi dan kompresi pada sistem kripto kompresi yang dirancang pada penelitian ini.



Gambar 2. Skema Proses Enkripsi dan Kompresi

Proses enkripsi dan kompresi dalam skema sistem kripto kompresi seperti pada Gambar 2 dimulai dengan mengenkripsi teks berupa karakter atau string yang terdapat dalam file dokumen dengan menggunakan kunci publik (public key) algoritma RSA. Untuk mengenkripsi dan dekripsi dengan menggunakan algoritma RSA, maka terlebih dahulu membangkitkan sepasang kunci, yaitu kunci publik (public key) dan kunci privat (private key). Adapun algoritma untuk membangkitkan kunci publik (public key) dan kunci privat (private key) algoritma RSA yaitu sebagai berikut:

1. Pilih dua buah bilangan prima sembarang, p dan q, misalkan $p = 71$ dan $q = 83$
2. Hitung nilai p sehingga diperoleh hasilnya: $n = p * q = 71 * 83 = 5893$
3. Hitung nilai totient (φ) sehingga diperoleh hasilnya: $\varphi(n) = (p-1)(q-1) = 5740$
4. Pilih sembarang bilangan e sebagai kunci publik yang relatif prima terhadap $\varphi(n)$ yaitu $1 > e < \varphi(n)$ dan $\gcd(e, \varphi(n)) = 1$. Misalkan dipilih $e = 3107$ karena relatif prima dengan $\varphi(n) = 5740$.
   Pembuktian, $gcd(3107, 5740) = 1$ sehingga nlai e yang digunakan yaitu $e = 3107$
5. Hitung kunci privat, disebut namanya d sedemikian agar $d * e \bmod \varphi(n) = 1$
   Misalkan dipilih nilai $d = 3863$ yang memenuhi syarat $d * e \bmod \varphi(n) = 1$
   Pembuktian, $d * e \bmod \varphi(n) = 3853 * 3107 \bmod 5740 = 1$ sehingga nlai d yang digunakan yaitu $d = 3863$.

Berdasarkan hasil perhitungan diatas maka diperoleh pasangan kunci publik dan kunci privat algoritma RSA sebagai berikut:

ISSN: 2964-1950

1.Kunci enkripsi (public key) adalah pasangan $(n, e) = (5893,3107)$

2.Kunci dekripsi (private key) adalah pasangan $(n, d) = (5893,3863)$

Setelah kunci publik (public key) dan kunci privat (private key) algoritma RSA dibangkitkan, maka proses enkripsi karakter atau string yang terdapat pada file dokumen dapat dilakukan. Adapun sampel data file dokumen yang digunakan pada proses perhitungan ebkripsi algoritma RSA dapat disajikan pada Gambar 3.

(a) (b)

Gambar 3. Sampel Data (a) Karakter (b) Nilai Desimal

Gambar 3 menampilkan nilai desimal dari sampel data yang diperoleh dengan menggunakan alat bantu (tools) dari https://onlinetexttools.com/. Setelah nilai desimal dari pesan yang terdapat dalam file dokumen didapatkan, maka proses enkripsi dengan algoritma RSA dapat dilakukan. Dengan mengambil 4 huruf pertama dari file dokumen sebagai berikut:

Plaintext (m_i) : I m a m

Desimal : 73 109 97 109

Proses enkripsi dengan menggunakan kunci publik algoritma RSA dapat dilakukan dengan menggunakan persamaan (1) adalah sebagai berikut:

1.Ambil kunci publik (public key) algoritma RSA yang telah dibangkitkan sebelumnya, yaitu pasangan $(n, e) = (5893,3107)$

2.Enkripsi plaintext dengan menggunakan persamaan (1), sehingga hasilnya:

$c_1 = 73^{3107} \ mod \ 5893 = 1529$

Lakukan hal yamg sama untuk semua plaintext (m_i) sehingga diperoleh hasil enkripsi (ciphertext) dalam bentuk bilangan desimal yaitu: 1529 5548 1745 5548.

Setelah hasil enkripsi (ciphertext) didapatkan maka tahap selanjutnya dilakukan proses kompresi untuk memperkecil ukuran data pada ciphertext. Metode kompresi yang digunakan dalam penelitian ini yaitu metode lossless compression dengan menggunakan algoritma Even Rodeh Code. Tahap awal kompresi adalah akan dilakukan pengelompokan nilai karakter hasil enkripsi (ciphertext) berdasarkan nilai frekuensi. Urutan karakter nilai yang hasilnya dapat ditunjukkan pada Tabel 2.

Tabel 2. Ciphertext Sebelum Dikompresi

Karakter	ASCII (Desimal)	ASCII (Biner)	Bit	Frekuensi	Bit x Frekuensi
5	53	00110101	8	6	48
spasi	32	00100000	8	3	24
4	52	00110100	8	3	24
1	49	00110001	8	2	16
8	56	00111000	8	2	16
9	57	00111001	8	1	8
2	50	00110010	8	1	8
7	55	00110111	8	1	8
Total Bit					**152**

ISSN: 2964-1950

Berdasarkan Tabel 2, satu karakter bernilai 8 bit bilangan biner dalam kode ASCII. Sehingga ciphertext (1529 5548 1745 5548) berjumlah 19 karakter pada string ciphertext mempunyai nilai biner sebanyak 152 bit, artinya string ciphertext sebelum dikompresi adalah 19 byte, dimana 1 karakter adalah 1 byte, dan 1 byte adalah 8 bit maka total ukuran ciphertext yang akan di kompresi = 152 bit.

Setelah menghitung jumlah frekuensi yang terdapat dalam ciphertext, maka tahap selanjutnya adalah dilakukan proses pengkodean algoritma Even Rodeh Code yang dimulai dengan jumlah frekuensi karakter terbesar hingga yang paling kecil sehingga didapat n (panjang bit dalam karakter yang telah diurutkan). Adapun hasil kompresinya dapat dilihat pada Tabel 3.

Tabel 3. Ciphertext Setelah Dikompresi

n	Karakter	Even-Rodeh Code	Frekuensi	Bit	Bit x Frekuensi
0	5	000	6	3	18
1	spasi	001	3	3	9
2	4	010	3	3	9
3	1	011	2	3	6
4	8	1000	2	4	8
5	9	1010	1	4	4
6	2	1100	1	4	4
7	7	1110	1	4	4
Total Bit					**62**

Setelah proses kompresi selesai, berdasarkan kode-kode Even Rodeh Code string pada Tabel 3, maka dilakukan penyusunan kode-kode tersebut sesuai dengan posisi karakter string asli pada ciphertext, sehingga akan menjadi string bit: 011000110010100010000000101000001011111001000000100000000101000.

Dalam tabel ASCII satu karakter dipresentasikan sebanyak 8 bit dengan bilangan biner. Namun, jika bit tersebut bukan kelipatan 8, maka dilakukan penambahan padding bit dan flag bit. Padding yaitu menambahkan bit 0 pada hasil kompresi yang bukan merupakan kelipatan 8. Sedangkan flag yaitu untuk menjelaskan berapa jumlah bit yang ditambahkan dalam melakukan padding. Terdapat 62 bit dari string bit di atas, maka dilakukan penambahan bit 0 sebanyak 2 kali agar jumlah bit data tersebut dapat habis bila dibagi dengan 8. Sehingga bit-bit data tersebut setelah diberikan padding menjadi:

01100011001010001000000010100000101111100100000010000000101000000

Karena terdapat 2 bit penambahan padding maka flag bits-nya adalah bilangan biner dari 2 dengan panjang 8 bit yaitu 00000010, sehingga bit-bit datanya setelah diberikan flag bits menjadi:

0110001100101000100000001010000010111110010000001000000010100000000000010

Setelah dilakukan penambahan bit padding dan bit flagging maka diperoleh total string bit hasil kompresi yaitu 62 (string bit) + 2 (bit padding) + 8 (bit flagging) = 72 bit.

Hasil dari proses enkripsi dan kompresi kemudian akan dianalisis menggunakan parameter Compression Ratio (CR) dan Space Savings (SS) untuk tujuan menguji kinerja dari sistem kripto kompresi.

1.Compression Ratio (CR) atau rasio kompresi adalah menghitung kinerja dari repersentase data yang sudah dikompresi dan sebelum dikompresi. Dengan menggunakan persamaan (3) maka dapat diperoleh Compression Ratio (CR) yaitu sebagai berikut:

$$CR = \frac{Compressed\ bits}{Uncompressed\ bits} = \frac{72}{152} = 0.47$$

2.Space Savings (SS) adalah persentase penghematan rang (memori) setelah file dikompresi dengan mencari persentase selisih antara data awal sebelum dikompresi dengan hasil data yang telah dikompresi. Dengan menggunakan persamaan (4) maka dapat diperoleh Space Saving (SS) yaitu:

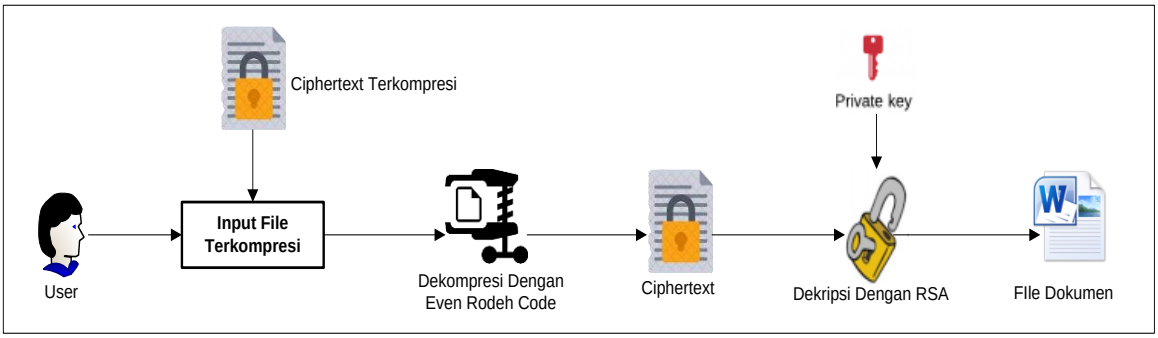
$$SS = \left(1 - \frac{Compressed\ bits}{Uncompressed\ bits}\right) x\ 100 = \left(1 - \frac{72}{152}\right) x\ 100 = 52.63$$

Berdasarkan hasil perhitungan kinerja dari sistem kripto kompresi menggunakan algoritma kriptografi RSA dan algoritma kompresi Even Rodeh Code, nilai dari Compression Ratio (CR) mencapai 0.47 atau 47% dan Space Savings (SS) yang dihasilkan cukup besar mencapai 52.63%, membuktikan bahwa kinerja dari sistem kripto kompresi cukup baik untuk kompresi file dokumen.

3.2 Analisis Proses Dekompresi dan Dekripsi

Proses dekompresi dan dekripsi dalam skema sistem kripto kompresi dimulai dengan melakukan proses dekompresi terlebih dahulu untuk mendapatkan kembali data ciphertext, kemudian dilanjutkan dengan melakukan dekripsi untuk mendapatkan kembali data aslinya. Gambar 4 merupakan diagram umum proses dekompresi dan dekripsi pada sistem kripto kompresi yang dirancang pada penelitian ini.

ISSN: 2964-1950



Gambar 4. Skema Proses Dekompresi dan Dekripsi

Proses dekompresi dan dekripsi dalam skema sistem kripto kompresi seperti pada Gambar 4 dimulai dengan membaca karakter yang terdapat pada file ciphertext terkompresi. Selanjutnya menentukan panjang string bit yang harus dibaca dengan menghitung padding dan flagging. Untuk proses dekompresi terhadap string bit yang telah dikompresi adalah dengan menentukan indeks terkahir untuk proses pembacaan string, yaitu total panjang string bit seluruhnya dikurang dengan flag ditambah padding atau dapat ditulis:

$$n = panjang\ string\ bit - (flagging + padding)$$

$$n = 48 - (flagging + padding)$$

Flagging pada string bit tersebut adalah 00000110 (8 bit terakhir pada string bit) yang bila dikonversi ke dalam desimal akan bernilai 2, maka padding 00. Sehingga panjang string bit yang harus dibaca adalah panjang string bit (72 bit) terkompresi dikurang total jumlah panjang padding dan flagging (2+8). Maka diperoleh nilai panjang string bit yang harus dibaca adalah $72 - (2 + 8) = 62$ sehingga akan menghasilkan string bit berikut:

011000110010100010000000101000001011111001000000100000000101000

Untuk melakukan dekompresi, maka pembacaan string bit dimulai dari indeks paling kecil (dilakukan dari indeks pertama sampai indeks terakhir) yang dapat diuraikan sebagai berikut:

011000110010100010000000101000001011111001000000100000000101000

Indeks ke-0 adalah **0**, pada Tabel 2 kode **0** tidak mewakilkan karakter apapun.

011000110010100010000000101000001011111001000000100000000101000

Maka dikuti oleh indeks ke-1 yaitu **1**, pada Tabel 3 kode **01** tidak mewakilkan karakter apapun.

0110001100101000100000001010000010111110010000001000000000101000

Maka dikuti oleh indeks ke-2 yaitu **1**, pada Tabel 3 kode **011** mewakili karakter angka "1". Pembacaan selanjutnya dimulai pada indeks ke-3 yaitu **0**

01100011001010001000000010100000101111100100000010000000101000

Maka dikuti oleh indeks ke-3 yaitu **0**, pada Tabel 3 kode **0** tidak mewakilkan karakter apapun.

01100011001010001000000010100000101111100100000010000000101000

Maka dikuti oleh indeks ke-4 yaitu **0**, pada Tabel 3 kode **00** tidak mewakilkan karakter apapun.

0110001100101000100000001010000010111110010000001000000000101000

Maka dikuti oleh indeks ke-5 yaitu **0**, pada Tabel 3 kode **000** mewakili karakter angka "5". Pembacaan selanjutnya dimulai pada indeks ke-5 yaitu **0**

0110001100101000100000001010000010111110010000001000000000101000

Maka dikuti oleh indeks ke-6 yaitu **1**, pada Tabel 3 kode **1** tidak mewakilkan karakter apapun.

0110001100101000100000001010000010111110010000001000000000101000

Maka dikuti oleh indeks ke-7 yaitu **1**, pada Tabel 2 kode **11** tidak mewakilkan karakter apapun.

0110001100101000100000001010000010111110010000001000000000101000

Maka dikuti oleh indeks ke-8 yaitu **1**, pada Tabel 3 kode **110** mewakili karakter angka "2".

Langkah tersebut akan dilakukan terus berlangsung sampai semua *string bit* habis sehingga didapatkan hasil dekompresi yaitu "1529 5548 1745 5548" yang merupakan hasil enkripsi (*ciphertext*) sebelumnya.

Setelah ciphertext diperoleh, maka selanjutnya akan dilakukan proses dekripsi yang bertujuan untuk mengembalikan pesan sudah di enkripsi pada langkah sebelumnya. Proses dekripsi dengan menggunakan kunci privat algoritma RSA dapat dilakukan dengan menggunakan persamaan (2) yaitu:

1.Ambil kunci privat (private key) algoritma RSA yaitu $(n, d) = (5893,3863)$

2.Dekripsi ciphertext dengan menggunakan persamaan (2), sehingga hasilnya:

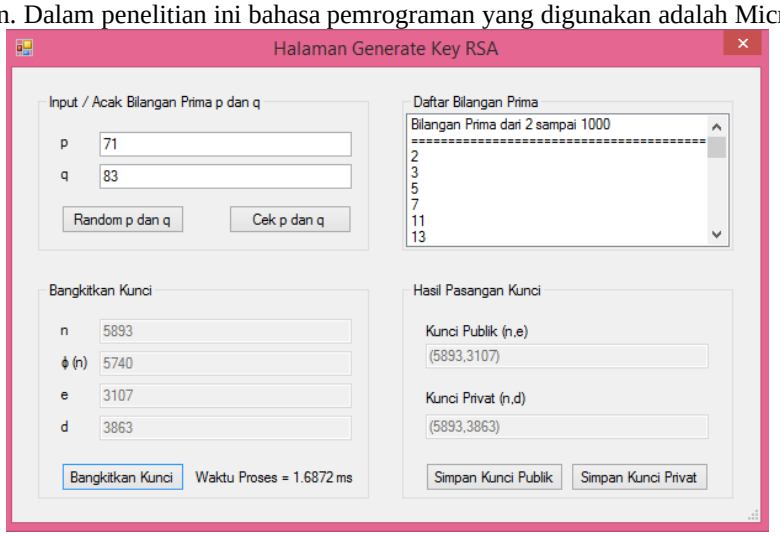
$m_1 = 1529^{3863}\ mod\ 5893 = 73$

Lakukan hal yamg sama untuk semua ciphertext (c_i) sehingga diperoleh hasil dekripsi (plaintext) yaitu "Imam" yang merupakan pesan asli yang sebenarnya.

ISSN: 2964-1950

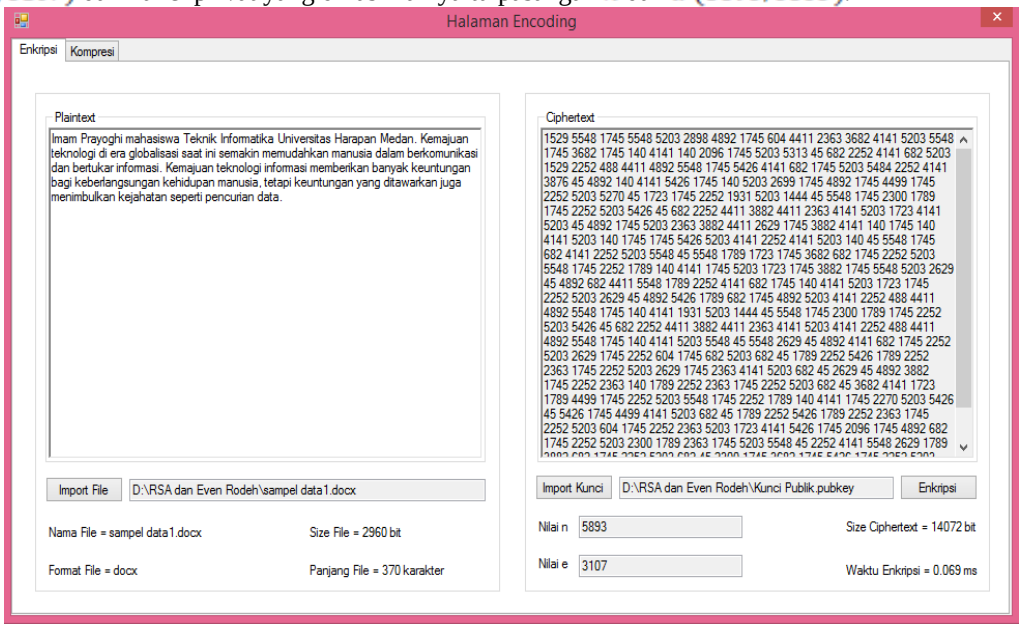
3.3 Implementasi Sistem

Setelah selesai menganalisis dan membuat rancangan dari sistem yang akan dibangun, selanjutnya adalah mengimplementasikan hasil analisis dan perancangan ke dalam bentuk perangkat lunak dengan menggunakan bahasa pemrograman. Dalam penelitian ini bahasa pemrograman yang digunakan adalah Microsoft Visual C#.



Gambar 5. Pengujian Generate Key RSA

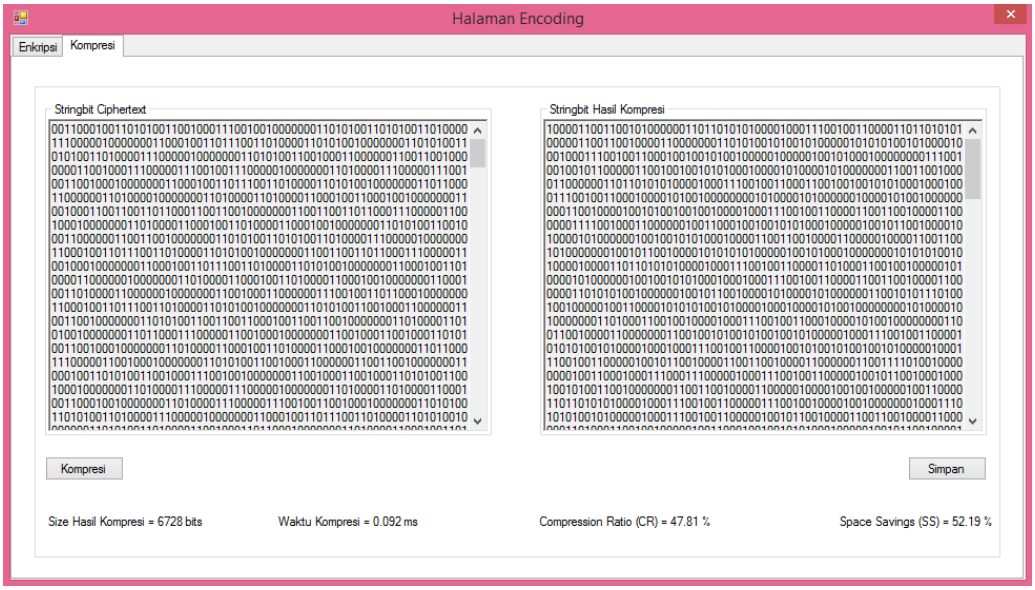
Sesuai Gambar 5, nilai bilangan prima yang dihasilkan untuk $p = 71$ dan $q = 83$. Sedangkan untuk nilai $n = 5893$ yang merupakan hasil dari $p * q$. Adapun kunci publik yang dihasilkan yaitu pasangan n dan e $(5893, 3107)$ dan kunci privat yang dihasilkan yaitu pasangan n dan d $(5893, 3863)$.



Gambar 6. Hasil Pengujian Enkripsi

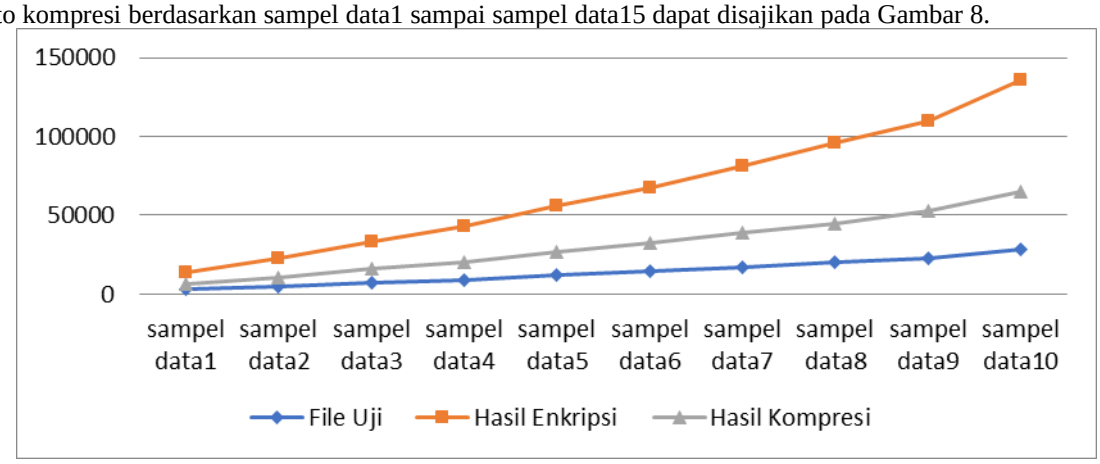
Gambar 6 merupakan hasil pengujian enkripsi pada file dokumen dengan format .docx menggunakan kunci publik algoritma RSA. File dokumen yang di uji menggunakan sampel data1 dengan size 2960 bit (370 karakter). Karakter (string) yang terdapat pada file dokumen yang telah di enkripsi akan menghasilkan ciphertext dalam bentuk bilangan desimal dengan size 14072 bit serta estimasi waktu enkripsi selama 0.069 ms (millisecond). Setelah file dokumen di enkripsi, selanjutnya akan dilakukan kompresi dengan menggunakan algoritma Even Rodeh Code. Hal ini dilakukan dengan tujuan untuk memperkecil size dari ciphertext.

ISSN: 2964-1950



Gambar 7. Hasil Pengujian Kompresi

Gambar 7 merupakan hasil pengujian dari proses kompresi pada hasil enkripsi (ciphertext) dengan menggunakan algoritma kompresi Even Rodeh Code. Dapat dilihat bawwa size dari ciphertext sebelum dikompresi adalah 14072 bit dan setelah dikompresi menjadi 6728 bit. Berdasarkan parameter yang digunakan dalam proses kompresi pada penelitian ini, diperoleh Compression Ratio (CR) sebesar 47.81, Space Savings (SS) sebesar 52.19 dan waktu kompresi selama 0.092 ms (millisecond). Adapun perbandingan hasil pengujian sistem kripto kompresi berdasarkan sampel data1 sampai sampel data15 dapat disajikan pada Gambar 8.



Gambar 8. Grafik Perbandingan Size File Uji, Hasil Enkripsi dan Hasil Kompresi

Gambar 8 memperlihatkan perbandingan size untuk setiap file uji, dimana size hasil enkripsi (ciphertext) mengalami peningkatan size secara konstan seiring dengan perubahan pada size file uji yang digunakan. Oleh karena itu metode kompresi berperan untuk memperkecil size dari hasil enkripsi.

3. KESIMPULAN

Kesimpulan yang dapat diambil setelah melakukan implementasi dan pengujian sistem kripto kompresi pada file dokumen dengan menerapkan algoritma kriptografi simetris RSA dan algoritma kompresi Even Rodeh Code adalah sebagai berikut:

1. Sistem kripto kompresi dapat di implementasikan kedalam aplikasi berbasis desktop dengan menggunakan bahasa pemrograman Visual C# yang bisa digunakan untuk meningkatkan kerahasiaan file dokumen serta memperkecil ukuran data pada file dokumen sehingga dapat menghemat kebutuhan akan ruang penyimpanan (storage) data menjadi lebih efisian.

2. Kinerja dari sistem kripto kompresi dapat dianalisis dengan menggunakan parameter Compression Ratio (CR), Space Saving (SS), dan Running Time (RT). Dari hasil pengujian menunjukkan bahwa size file uji yang berbeda menghasilkan nilai perhitungan Compression Ratio (CR), Space Savings (SS) dan RT (Running Time) yang berbeda juga. Semakin besar size yang akan dikompresi maka semakin besar juga size yang dapat diperkecil ukurannya atau semakin besar ruang penyimpanan yang dapat di hemat. Sementara hasil RT mengalami peningkatan nilai secara konstan seiring dengan perubahan size file uji yang digunakan. Artinya semakin besar size file uji yang akan dikompresi maka semakin lama juga waktu yang dibutuhkan untuk mengkompresinya.

DAFTAR PUSTAKA

[1] Rizki, M., & Ariyani, P. F. (2021). Penerapan Kriptografi Dengan Menggunakan Algoritma RSA Untuk Pengamanan Data Berbasis Desktop Pada PT Trias Mitra Jaya Manunggal. *Skanika*, *4*(2), 1–6. https://doi.org/10.36080/skanika.v4i2.1991

[2] Fatonah, Dadang Iskandar Mulyana, Heryani, A. P., & Khoirunnisa, V. (2022). Implementasi Metode Rivest Shamir Adleman untuk Enkripsi dan Dekripsi Text. *Jurnal Informatika Dan Teknologi Komputer (J-ICOM)*, *3*(1), 32–39. https://doi.org/10.33059/j-icom.v3i1.4990

[3] Saputro, T. H., Hidayati, N., & Ujianto, E. I. H. (2020). Survei Tentang Algoritma Kriptografi Asimetris. *Jurnal Informatika Polinema*, *6*(2), 67–72. https://doi.org/10.33795/jip.v6i2.345

[4] Hariska, E., Yuliani, E., & Nasution, S. D. (2021). Performance Comparison Analysis of the Elias Delta Code Algorithm with the Even Rodeh Code Algorithm for Compressing Image Files. *The IJICS (International Journal of Informatics and Computer Science)*, *5*(1), 29–36. https://doi.org/10.30865/ijics.v5i1.2888

[5] Hardi, S. M., Zarlis, M., Lubis, D. R. P., Sihombing, P., & Elveny, M. (2021). Text File Compression Using Hybrid Run Length Encoding (Rle) Algorithm with even Rodeh Code (ERC) and Variable Length Binary Encoding (VLBE) to Save Storage Space. *Journal of Physics: Conference Series*, *1830*(1). https://doi.org/10.1088/1742-6596/1830/1/012022

[6] Syahputra, R. (2020). Analysis of Even-Rodeh Code For Text Compression. *The IJICS (International Journal of Informatics and Computer Science)*, *4*(1), 1–4. https://doi.org/10.30865/ijics.v4i1.1348

[7] Ihsan, I., & Utomo, D. P. (2020). Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, *4*(1), 223–227. https://doi.org/10.30865/komik.v4i1.2684

[8] Auliyah, A. I. (2020). Implementasi Kombinasi Algoritma Enkripsi Rivest Shamir Adleman (RSA) dan Algoritma Kompresi Huffman Pada File Document. *Indonesian Journal of Data and Science*, *1*(1), 23–28. https://doi.org/10.33096/ijodas.v1i1.6

[9] Muzaki, A. A. (2021). *Studi Kompleksitas Algoritma Enkripsi Teks Simetri dan Asimetri*. https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Makalah/Makalah-Matdis-2020 (64).pdf

[10] Muchlis, B. S., Budiman, M. A., & Rachmawati, D. (2017). Teknik Pemecahan Kunci Algoritma Rivest Shamir Adleman (RSA) dengan Metode Kraitchik. *Jurnal & Penelitian Teknik Informatika*, *e-ISSN : 2541-2019, p-ISSN : 2541-044X*, *2*(2), 49–64. http://jurnal.polgan.ac.id/index.php/sinkron/article/view/75

[11] Diana, M., & Zebua, T. (2018). Optimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *2*(1), 12–22. https://doi.org/10.30645/j-sakti.v2i1.52

[12] Hasugian, B. S. (2017). Peranan Kriptografi Sebagai Keamanan Sistem Informasi Pada Usaha Kecil Dan Menengah. *Journal Warta*, *53*(9), 2. http://ejournal.ust.ac.id/index.php/JTIUST/article/view/190

[13] Pujianto, Mujito, Prasetyo, B. H., & Prabowo, D. (2020). Perbandingan Metode Huffman dan Run Length Encoding Pada Kompresi Document. *InfoTekJar: Jurnal Nasional Informatika Dan Teknologi Jaringan*, *5*(1), 216–223. https://doi.org/10.30743/infotekjar.v5i1.2892

[14] Pramadi, A. A., Nasution, S. D., & Purba, B. (2019). Penerapan Algoritma Even-Rodeh Pada Aplikasi Kompresi File Gambar. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, *3*(1), 73–84. https://doi.org/10.30865/komik.v3i1.1570