Sistem Kriptokompresi Menggunakan Algoritma Asimetris Rabin-P dan Algoritma Lossless Compression Golbach Code

Tri Joko Wardani[1], Imran Lubis[2], Kalvin Chiuloto[3]
[1,2,3]Program Studi Teknik Informatika Fakultas Teknik dan Komputer
Universitas Harapan Medan
Jl. H.M. Jhoni No 70 Medan, Indonesia
trijokowardani7@gmail.com, imran.loebis.medan@gmail.com,
kalvin.chiuloto@gmail.com

Abstract

The in line line with the development of information technology, information security must also be considered. Advances in information technology provide many advantages for the survival of human life, but the advantages offered by technology also cause losses such as data theft. Data originating from word processing documents is one form of document that is commonly used to store information, whether private or confidential. The security issues are one of the most important aspects in the world of information technology which is a human need in the era of globalization, for example document security. Therefore, document security is very necessary and must be protected from irresponsible parties, either when stored in storage or when distributed. In addition to the security aspect, the thing that needs to be considered is also about the storage media. Therefore, additional steps are needed to streamline the storage media by compressing the data that has been secured so that its size becomes smaller. In this study, we will combine cryptographic techniques and compression techniques called cryptocompression. The cryptocompression system in this research is applied by encrypting data in text files using the public key of the Rabin-p algorithm which will produce an encrypted text file. Then it is compressed using Goldbach Code lossless compression algorithm to reduce its size. From the results of this test, it is concluded that the cryptocompression system is able to maintain the confidentiality of information because it has been encrypted and can save more efficient storage space because it has been compressed first. Text files that have been encrypted and compressed can be returned to the original file through a decompression and decryption process according to the size and number of characters contained in the original text file.

Keywords : *Cryptography, Compression, Cryptocompression, Rabin-p, Goldbach Code*

Abstrak

Sejalan dengan perkembangan teknologi informasi maka keamanan informasi juga harus diperhatikan. Kemajuan teknologi informasi memberikan banyak keuntungan bagi keberlangsungan kehidupan manusia, tetapi keuntungan yang ditawarkan oleh teknologi juga menimbulkan kerugian seperti pencurian data. Data yang berasal dari dokumen pengolah kata menjadi salah satu bentuk dokumen yang umum digunakan untuk menyimpan informasi, baik yang sifatnya pribadi atau rahasia. Masalah keamanan merupakan salah satu aspek paling penting dalam dunia teknologi informasi yang menjadi kebutuhan manusia pada era globalisasi, misalnya keamanan dokumen. Oleh karena itu, keamanan dokumen sangat diperlukan dan harus dilindungi dari pihak-pihak yang tidak bertanggung jawab, baik saat disimpan pada ruang penyimpanan (storage) ataupun pada saat didistribusikan. Selain aspek keamanan, maka hal yang perlu diperhatikan juga adalah tentang media penyimpanan. Oleh karena itu, diperlukan langkah tambahan untuk mengefisiensikan media penyimpanan dengan melakukan kompresi pada data yang sudah diamankan supaya ukurannya menjadi lebih kecil. Dalam penelitian ini, akan menggabungkan teknik kriptografi dan teknik kompresi yang disebut dengan kriptokompresi.

Sistem kriptokompresi dalam penelitian diterapkan dengan cara mengenkripsi data pada file text menggunakan kunci publik dari algoritma Rabin-p yang akan menghasilkan file text terenkripsi. Selanjutnya dikompresi menggunakan algoritma lossless compression Goldbach Code untuk memperkecil ukurannya. Dari hasil pengujian disimpulkan bahwa sistem kriptokompresi mampu menjaga kerahasiaan sebuah informasi karena telah dienkripsi serta dapat menghemat ruang penyimpanan yang lebih efisien karena sudah dikompresi terlebih dahulu. File text yang telah dienkripsi dan dikompresi dapat dikembalikan lagi kedalam file aslinya melalui proses dekompresi dan dekripsi sesuai dengan size dan jumlah karater yang terdapat pada file text aslinya.

Kata kunci : *Kriptografi, Kompresi, Kriptokompresi, Rabin-p, Goldbach Code.*

1. PENDAHULUAN

Teknologi informasi meliputi segala hal yang berkaitan dengan proses, penggunaan sebagai alat bantu, manipulasi, dan pengelolaan informasi. Sedangkan teknologi komunikasi adalah segala sesuatu yang berkaitan dengan penggunaan alat bantu untuk memproses dan mentransfer data dari perangkat yang satu ke lainnya [1]. Sehingga teknologi informasi dan komunikasi merupakan dua buah konsep yang tidak terpisahkan yang mencakup semua peralatan teknis untuk memproses dan menyampaikan informasi. Teknologi komunikasi ditekankan pada bagaimana suatu hasil data dapat disampaikan ke tempat tujuan sedangkan teknologi informasi lebih ditekankan pada hasil data yang diperoleh. Teknologi informasi berkembang cepat seiring meningkatnya perkembangan komputer beserta perangkat-perangkat pendukung lainnya serta teknologi komunikasi berkembang cepat dengan meningkatnya perkembangan teknologi elektronika dan sistem transmisi sehingga suatu informasi dapat disampaikan dengan cepat dan tepat. Sejalan dengan perkembangan informasi maka keamanan informasi juga harus diperhatikan. Kemajuan teknologi informasi memberikan banyak keuntungan bagi keberlangsungan kehidupan manusia, tetapi keuntungan yang ditawarkan oleh teknologi juga menimbulkan kerugian seperti pencurian data. Sehingga perkembangan ilmu untuk mengamankan data semakin ditingkatkan agar pengguna teknologi selalu merasa aman. Berbagai cara dilakukan untuk menjaga keamanan data seperti menyembunyikan data (teknik steganografi) dan penyandian data menjadi suatu kode-kode yang tidak dimengerti (teknik kriptografi), sehingga apabila dicuri atau disadap oleh orang lain akan kesulitan untuk mengetahui dan memahami informasi yang sebenarnya.

Pada zaman dahulu tepatnya Romawi Kuno, Julius Caesar telah menggunakan teknik kriptografi yang kemudian dijuluki Caesar Cipher untuk mengirim pesan secara rahasia. Kejadian yang sama juga terjadi pada perang dunia kedua, yaitu pihak sekutu mampu memecahkan kode mesin kriptografi Jerman yang disebut dengan Enigma. Keberhasilan tersebut banyak membantu pihak sekutu dalam memenangkan pertempuran. Pada dasarnya kriptografi dipahami sebagai ilmu tentang menyembunyikan pesan, tetapi seiring perkembangan zaman hingga saat ini pengertian kriptografi berkembang menjadi ilmu tentang teknik matematis yang digunakan untuk menyelesaikan persoalan keamanan berupa privasi dan otentikasi. Teknik kriptografi merupakan salah satu alternatif solusi yang dapat diterapkan untuk menjaga keamanan data, yaitu dengan cara memanipulasi data ke dalam bentuk yang tidak dimengerti oleh banyak orang. Terdapat dua jenis algoritma kriptografi berdasarkan jenis kuncinya, yaitu algoritma simetris (konvensional) dan algoritma asimetris (kunci publik). Algoritma simetris adalah algoritma kriptografi yang menggunakan satu kunci untuk proses enkripsi dan dekripsi, sedangkan algoritma asimetris, kunci terbagi menjadi dua bagian yaitu kunci umum (public key) untuk proses enkripsi dan kunci pribadi (private key) untuk proses dekripsi. Dalam penelitian yang dilakukan oleh [2] menyimpulkan bahwa algoritma simetris AES mempunyai kinerja waktu enkripsi dan dekripsi yang lebih baik daripada algoritma asimetris RSA seiring dengan besarnya ukuran karakter yang

dienkripsi. Namun, kompresi algoritma Huffman terhadap ciphertext pada algoritma asimetris RSA lebih efisien daripada AES. Dari hasil penelitian tersebut dapat disimpulkan bahwa penerapan kompresi lebih efisien jika diterapkan pada algoritma kriptografi asimetris dibandingkan dengan algoritma simetris.

Penelitian terdahulu yang relevan dengan pokok permasalahan dalam penelitian ini dijadikan sebagai data pendukung. Penelitian terdahulu mengenai algoritma Rabin-p seperti yang dilakukan oleh [3], menyimpulkan bahwa algoritma Rabin-p bekerja lebih cepat dan menggunakan lebih sedikit penyimpanan dibandingkan dengan kriptosistem mirip Rabin lainnya. Penelitian lainnya juga dilakukan oleh [4], hasil penelitian dengan menggunakan lima sampel uji file text berformat .txt menyimpulkan bahwa algoritma Rabin-p lebih efisien dibandingkan algoritma RSA-CRT berdasarkan pengukuran waktu eksekusi komputasi dan kompleksitas algoritma. Penelitian terdahulu mengenai algoritma Goldbach Code seperti yang dilakukan oleh [5], hasil yang dicapai dalam penelitian tersebut djelaskan bahwa algoritma Goldbach Code merupakan algoritma pengkompresian file yang cukup handal serta dapat menghemat space penyimpanan. Penelitian lainnya juga dilakukan oleh [6], menyimpulkan bahwa hasil kompresi pada file text menggunakan Goldbach Code sangat bagus dan hasilnya tidak berkurang dari file asli atau tidak ada pengurangan.

File .txt merupakan dokumen teks standar yang berisi teks yang tidak diformat dan berguna untuk menyimpan informasi dalam teks biasa. File .txt dapat diproses oleh sebagian besar program perangkat lunak lain seperti Notepad. Sedangkan file .docx merupakan dokumen yang berisi teks yang di format dan dapat diproses menggunakan perangkat lunak Microsoft Word. Data yang berasal dari dokumen pengolah kata seperti Notepad dan Microsoft Word menjadi salah satu bentuk dokumen yang umum digunakan untuk menyimpan informasi, baik yang sifatnya tidak penting maupun yang sifatnya pribadi atau rahasia. Oleh karena itu, keamanan dokumen sangat diperlukan dan harus dilindungi dari gangguan maupun serangan yang terjadi dari pihak-pihak yang tidak bertanggung jawab, baik saat disimpan pada ruang penyimpanan (storage) ataupun pada saat didistribusikan. Banyak hal yang dapat dilakukan untuk meningkatkan keamanan dokumen, salah satunya dengan menyandikan isi pesan menjadi suatu kode-kode yang tidak dimengerti atau yang sering dikenal dengan teknik kriptografi. Masalah keamanan merupakan salah satu aspek paling penting dalam dunia teknologi informasi yang menjadi kebutuhan manusia pada era globalisasi, misalnya keamanan dokumen. Dalam penelitian ini, akan menggabungkan teknik kriptografi dan teknik kompresi yang disebut dengan kriptokompresi menggunakan algoritma asimetris Rabin-p dan algoritma lossless compression Goldbach Code. Kriptokompresi dalam penelitian ini merupakan teknik yang memanfaatkan dua metode kedalam satu proses, artinya data yang akan diamankan yaitu berupa file text akan dienkripsi terlebih dahulu dengan algoritma asimetris Rabin-p untuk tujuan mengamankan data, kemudian dikompresi menggunakan algoritma lossless compression Goldbach Code guna memperkecil ukuran data.

2. METODOLOGI PENELITIAN

Dalam mendukung jalannya penelitian ini agar lebih terarah dan sistematis maka dibutuhkan suatu tahapan desain penelitian yang disusun dengan terstruktur. Penelitian ini menerapkan beberapa metode penelitian yang dapat dijelaskan yaitu:

1. Identifkasi Masalah

Pada tahap ini penulis mengidentifikasi apa-apa saja yang menjadi permasalahan dan mengambil permasalahan tersebut menjadi topik penelitian sehingga peneliti dapat mencari solusi yang nantinya akan menjadi tujuan dari penelitian ini. Permasalahan yang diangkat dalam penelitian ini terkait dalam hal meningkatkan keamanan data pada file text serta bagaimana menghemat

kebutuhan akan ruang penyimpanan (storage) data menjadi lebih efisian. Dalam penelitian ini, akan menggabungkan teknik kriptografi dan teknik kompresi yang disebut dengan sistem kriptokompresi sebagai alternatif solusi dari permasalahan tersebut.

2. Studi Pustaka

Pada tahap ini bertujuan untuk mendapatkan landasan teori mengenai permasalahan yang akan diteliti sehingga dapat memahami permasalahan yang diteliti sesuai dengan pembahasan yang dilakukan. Sumber pustaka pada penelitian ini diperoleh dengan membaca berbagai literatur seperti buku dan jurnal atau hasil kajian dari penelitian terdahulu.

3. Analisis dan Perancangan Sistem

Pada tahap ini dilakukan analisis penerapan algoritma kriptografi asimetris Rabin-p dan kompresi Goldbach Code dalam skema sistem kriptokompresi. Terdapat dua proses utama yang dilakukan dalam tahap ini, yaitu enkripsi dan kompresi pada tahap pertama yang bertujuan untuk mentransformasi file text menjadi ciphertext dengan menggunakan kunci algoritma Rabin-p, selanjutnya akan dikompresi lagi dengan menggunakan kompresi Goldbach Code sehingga ukuran datanya menjadi lebih kecil. Proses kedua yaitu dekompresi dan dekripsi yang bertujuan untuk mengembalikan data ke bentuk aslinya. Sedangkan Perancangan sistem dilakukan dengan membuat flowchart sistem dan tampilan interface (antarmuka) sistem yang akan di integrasikan dengan aplikasi pada tahap implementasi sistem.

4. Implementasi dan Pengujian Sistem

Pada tahap ini akan dilakukan pengkodean (coding) menggunakan bahasa pemrograman Microsoft Visual C#.NET yang mengacu pada perancangan sistem yang telah dibuat sebelumnya. Implementasi sistem dilakukan dengan menampilkan ke user (pengguna) mengenai hasil aplikasi berbasis desktop tentang sistem kriptokompresi yang telah dibuat. Sedangkan tahap pengujian dilakukan dengan tujuan untuk menjamin sistem yang dibuat sesuai dengan hasil analisis dan perancangan serta menghasilkan satu kesimpulan apakah sistem tersebut sesuai dengan yang diharapkan.

5. Kesimpulan Penelitian

Pada tahap ini diambil kesimpulan yang menjawab tujuan akhir dari penelitian berdasarkan hasil analisis sampai pengujian sistem yang telah dilakukan.

3. HASIL DAN PEMBAHASAN

3.1 Kriptografi

Kata kriptografi berasal dari bahasa Yunani, “cryptós” yang berarti tersembunyi dan “gráphein” yang berarti tulisan. Teknik penulisan pesan rahasia ini digunakan oleh bangsa mesir sekitar 3000 tahun sebelum masehi. Penulisan rahasia ini disebut hieroglyphcs dimana mereka (bangsa mesir kuno) menyembunyikan tulisan supaya tidak dapat diketahui oleh pihak yang tidak diharapkan. Kriptografi tidak hanya mengenai penyembunyian pesan atau tulisan tetapi lebih pada sekumpulan teknik matematika untuk keamanan informasi yang bersifat kerahasiaan [7]. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data 8]. Kriptografi bertujuan untuk menjaga kerahasiaan informasi yang tergantung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut dengan kriptologi (cryptology) [5].

Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Enkripsi adalah suatu proses yang dilakukan untuk mengubah pesan asli (plaintext) menjadi pesan tersembunyi (ciphertext). Sedangkan suatu proses yang dilakukan untuk mengubah pesan tersembunyi menjadi pesan biasa (yang mudah dibaca) disebut dekripsi [9]. Penerapan kriptografi saat ini telah menjadi salah satu syarat penting dalam keamanan teknologi informasi terutama dalam pengiriman pesan rahasia. Pengiriman pesan rahasia sangat rentan terhadap serangan yang dilakukan oleh pihak ketiga, seperti penyadapan, pemutusan komunikasi, dan pengubahan pesan yang dikirim. Kriptografi dapat meningkatkan keamanan dalam pengiriman pesan atau komunikasi data dengan cara menyandikan pesan tersebut berdasarkan algoritma dan kunci tertentu yang hanya diketahui oleh pihak-pihak yang berhak atas data, informasi dan dokumen tersebut.

3.2 Algoritma Rabin-p

Algoritma Rabin-p merupakan varian dari algoritma Rabin yang dirancang oleh M. A. Asbullah dan M. R. K. Ariffin. Salah satu keuntungan yang jelas dari algoritma ini adalah bahwa prosedur dekripsi dari algoritma Rabin-p hanya menghasilkan satu hasil yaitu pesan asli, sehingga tidak ada lagi kebingungan di pihak penerima tentang mengetahui pesan asli dari empat hasil dekripsi seperti pada algoritma Rabin [10]. Algoritma Rabin-p bekerja lebih cepat dan menggunakan lebih sedikit penyimpanan dibandingkan dengan dua kripto sistem mirip Rabin lainnya, yaitu Rabin Takagi [3].

Algoritma Rabin-p dinamai Rabin dengan tambahan p yang melambangkan bahwa skema yang diusulkan hanya menggunakan satu bilangan prima p sebagai kunci dekripsi. Berikut ini akan dijelaskan prosedur pembuatan kunci, enkripsi, dan dekripsi algoritma Rabin-p [10], yaitu sebagai berikut:

1. Pembangkitan Kunci Algoritma Rabin-p

Untuk mengenkripsi dan dekripsi pesan dengan menggunakan algoritma Rabin-p terlebih dahulu membangkitkan sepasang kunci, yaitu kunci publik (public key) dan kunci privat (private key). Berikut ini algoritma penyelesaiannya yaitu:

a) Tentukan k sebagai parameter keamanan.
b) Pilih dua buah bilangan acak prima untuk nilai p dan q dimana $2^k < p, q < 2^{k+1}$ memenuhi $p \equiv q \equiv 3 (\mathrm{mod}\ 4)$.
c) Hitung $N = p^2 q$
d) Publikasikan N sebagai kunci publik (public key) untuk enkripsi
e) Kunci dekripsi (private key) adalah nilai p dan sifatnya rahasia

2. Proses Enkripsi Algoritma Rabin-p

Pengamanan data berdasarkan algoritma teknik kriptografi dilakukan dengan merubah pesan yang akan dirahasiakan (plaintext) menjadi sandi (ciphertext). Proses untuk mengkonversi plaintext menjadi ciphertext disebut dengan proses enkripsi. Adapun proses enkripsi pesan (plaintext) dengan menggunakan algoritma Rabin-p dapat dijelaskan tahapannya sebagai berikut:

a) Langkah pertama ambil kunci publik (public key) algoritma Rabin-p yang telah dibangkitkan yaitu N dan parameter keamanan nilai k
b) Pilih plaintext $0 < m < 2^{2k-1}$ dimana $\mathrm{GCD}(m, N) = 1$
   Dalam hal ini m adalah plaintext yang terdapat dalam file text.
c) Hitung $c \equiv m^2 (\mathrm{mod}\ N)$
   Dalam hal ini c adalah hasil enkripsi file text (ciphertext).

3. Proses Dekripsi Algoritma Rabin-p

Proses untuk mengkonversi ciphertext menjadi plaintext disebut dengan proses dekripsi. Proses dekripsi algoritma Rabin-p dengan menggunakan kunci privat dapat dapat dijelaskan tahapannya sebagai berikut:

a) Langkah pertama ambil hasil enkripsi (ciphertext) yaitu c
b) Langkah kedua ambil kunci privat (private key) Rabin-p yaitu p

c) Hitung nilai $w \equiv c \bmod p$
d) Hitung nilai $m_p \equiv w^{\frac{p+1}{4}} \bmod p$
e) Hitung nilai $i = \frac{c - m_p^2}{p}$
f) Hitung nilai $j \equiv \frac{i}{2m_p} \bmod p$
g) Hitung nilai $m_1 = m_p + jp$
h) Jika $m_1 < 2^{2k-1}$ maka $m = m_1$
i) Selain itu, $m = p^2 - m_1$

Dasar penulis memilih algoritma ini adalah berdasarkan hasil penelitian terdahulu seperti yang dilakukan oleh [9], hasil penelitian dengan menggunakan lima sampel uji file teks berformat .txt menyimpulkan bahwa algoritma Rabin-p lebih efisien dibandingkan algoritma RSA-CRT berdasarkan pengukuran waktu eksekusi komputasi dan kompleksitas algoritma.

3.3 Algoritma Goldbach Code

Algoritma Goldbach Code adalah algoritma yang dibuat oleh Peter Fenwick yang dibuat menggunakan conjecture Goldbach. Conjecture Goldbach ini diciptakan oleh Christian Goldbach yaitu salah satu matematikawan terkenal pada abad 17. Conjecture Goldbach berisi setiap bilangan genap lebih besar dari 2 merupakan hasil penjumlahan dari 2 buah bilangan prima [11]. Algoritma Goldbach Code adalah algoritma yang diasumsikan menggunakan teori Goldbach Conjecture yaitu semua bilangan genap positif yang lebih besar dari 2 merupakan penjumlahan dari dua bilangan prima [12]. Bilangan prima merupakan bilangan bulat yang hanya memiliki dua faktor yaitu 1 dan bilangan itu sendiri. Misalnya, bilangan bulat 5 adalah bilangan yang hanya dapat membagi bilangan 1 dan bilangan itu sendiri. Konsep kerja algoritma Goldbach Code yaitu dengan menghitung jumlah frequensi kemunculan tiap karakter dari yang terbesar sampai yang terkecil, dan dilanjutkan dengan mencari codeword dengan cara mengkodekan bilangan bulat positif n dengan menghubahnya menjadi bilangan bulat positif genap dengan rumus $(2n + 3)$ dan kemudian menuliskan pasangan penjumlahan bilangan prima dalam keadaan terbalik [13]. Algoritma Goldbach Codes memiliki tiga kode, Goldbach Codes yang pertama dinamakan "G0" dan Goldbach Codes kedua dinamakan "G1" serta Goldbach Codes yang ketiga dinamakan "G2" adalah perluasan dari G1 Codes [5]. Adapun tahapan yang dilakukan dalam proses kompresi data pada file text dengan menggunakan algoritma Goldbach Code [11] dapat dijelaskan sebagai berikut:

1. Langkah pertama mencari frekuensi kemunculan setiap karakter.
2. Kemudian urutkan karakter berdasarkan frekuensi kemunculan dari yang terbesar hingga terkecil.
3. Selanjutnya cari nilai $(2n + 3)$.
4. Setelah nilai dari n2 ditemukan, langkah berikutnya menentukan bilangan prima dan codeword G0 algoritma Goldbach Code dari nilai n2.
5. Langkah selanjutnya adalah mencari nilai bit tiap karakter, dimana jumlah digit codeword mewakili nilai bit karakter.
6. Hasil dari algoritma Goldbach Code adalah teks sebelum dikompresi kemudian dikompresi menggunakan codeword yang mewakili setiap karakter.

Sedangkan tahapan yang dilakukan dalam proses dekompresi data pada file text dengan menggunakan algoritma Goldbach Code [11] dapat dijelaskan sebagai berikut:

1. Mengambil nilai biner 1 dengan urutan ke-2 diikuti dengan nilai biner dibelakangnya sebagai identitas setiap karakter, begitu selanjutnya. Misalkan biner 1001101, ambil nilai biner 1 diurutan ke 2 diikuti dengan nilai biner sebelumnya maka didapat 1001. Nilai biner "1001" tersebut mewakili identitas sebuah karakter.
2. Kemudian setelah identitas karakter didapat, pembacaan karakter tersebut melalui proses header atau pembacaan ulang karakter yang menjadi kunci dalam proses dekompresi.

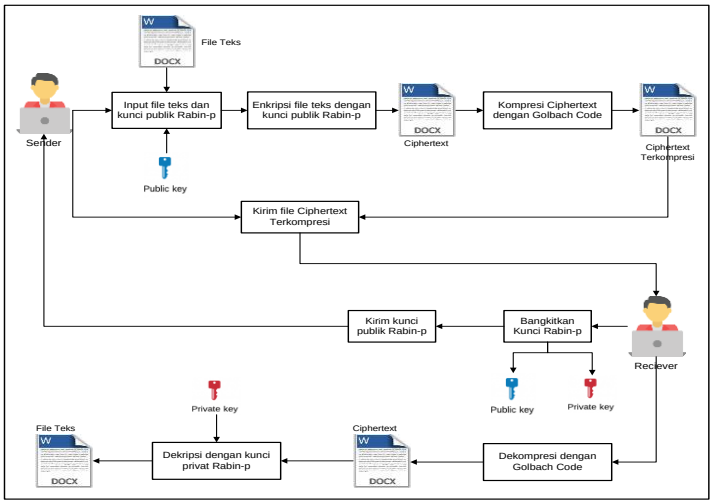
Algoritma Goldbach Code G0 adalah dasar dari algoritma Goldbach G1 Code. Goldbach G1 Code memiliki prinsip untuk menentukan dua bilangan prima Pi dan Pj (di mana $i \leq j$) yang jumlahnya menghasilkan bilangan bulat yang diberikan n, dan mengkodekan pair $(i, j - i + 1)$ dengan dua kode gamma [14].

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. Parameter yang digunakan dalam proses kompresi pada penelitian ini adalah Ratio of Compression (RC), Compression Ratio (CR), Space Savings (SS) serta Time Compression (TM) yang dapat dijelaskan yaitu:

1. Ratio of Compression (RC)
   Ratio of Compression atau rasio kompresi adalah menghitung kinerja dari repersentase data yang sudah dikompresi dan sebelum dikompresi [15]. Secara matematis rasio kompresi dapat dituliskan dengan persamaan (2.1).
   $$RC = 100\% - \left(\frac{Compressed\ bits}{Uncompressed\ bits}\right) * 100\% \quad (2.1)$$
   Misalkan rasio kompresi adalah 10% artinya 10% dari data semula telah berhasil dimampatkan.
2. Compression Ratio (CR)
   Compression Ratio atau kompresi rasio adalah persentase perbandingan antara data yang sudah dikompresi dan sebelum dikompresi [16]. Secara matematis kompresi rasio dapat dituliskan dengan persamaan (2.2).
   $$CR = \frac{Compressed\ bits}{Uncompressed\ bits} \quad (2.2)$$
3. Space Saving (SS)
   Space Savings adalah perbedaan antara data sebelum dikompresi dan data terkompresi [16]. Space savings merupakan persentase penghematan ruang (memori) setelah file dikompresi dengan mencari persentase selisih antara data awal sebelum dikompresi dengan hasil data yang telah dikompresi. Secara matematis space savings dapat dituliskan dengan persamaan (2.3).
   $$SS = \left(1 - \frac{Compressed\ bits}{Uncompressed\ bits}\right) x\ 100 \quad (2.3)$$
4. Time Compression (TM)
   Time Compression adalah perhitungan waktu yang diperoleh ketika algoritma melakukan kompresi atau dekompresi [16]. Time Compression pada pengujian sistem akan diukur dalam satuan millisecond (ms). Semakin sedikit waktu yang diperlukan sistem untuk melakukan kompresi, maka semakin efektif metode kompresi yang digunakan.

3.4 Analisis Proses Kriptokompresi

Kriptokompresi dalam penelitian ini merupakan teknik yang memanfaatkan dua metode kedalam satu proses, artinya data yang akan diamankan yaitu berupa file text akan dienkripsi terlebih dahulu dengan algoritma asimetris Rabin-p untuk tujuan mengamankan data, kemudian dikompresi menggunakan algoritma lossless compression Goldbach Code guna memperkecil ukuran data.

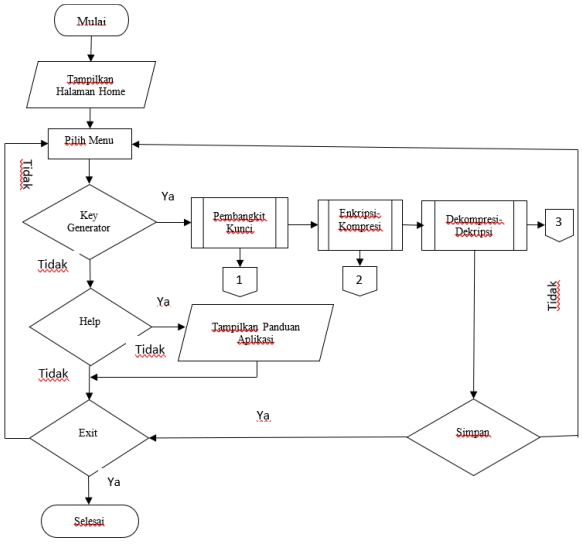


Gambar 1. Skema Sistem Kriptokompresi

Skema sistem kriptokompresi pada gambar 1 dapat dijelaskan untuk proses pertama kali penerima pesan (reciever) membangkitkan kunci Rabin-p yang akan menghasilkan sepasang kunci, yaitu kunci publik (public key) dan kunci privat (private key). Langkah selanjutnya reciever akan mengirimkan public key ke pengirim pesan (sender). Sender melakukan proses enkripsi file teks mnggunakan kunci publik Rabin-p dan menghasilkan ciphertext yang selanjutnya akan dikompresi menggunakan algoritma Goldbach Code yang akan menghasilkan file ciphertext yang terkompresi. Setelah itu akan dikirmkan ke reciever. Setelah reciever menerima file ciphertext yang terkompresi, maka tahap pertama yang dilakukan yaitu proses dekompresi menggunakan algoritma Goldbach Code yang akan menghasilkan file ciphertext. Selanjutnya akan di dekripsi menggunakan kunci privat algoritma Rabin-p dan menghasilkan file teks yang asli.

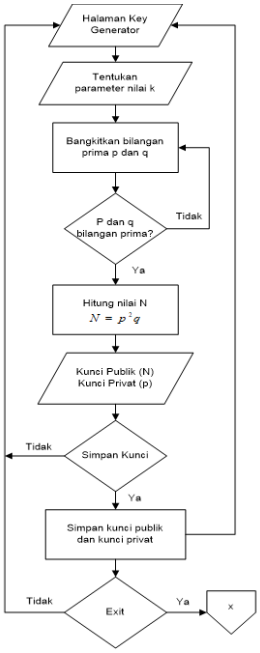
3.5 Flowchart Sistem

Flowchart atau diagram alir merupakan gambar atau bagan yang pada penelitian ini akan menggambarkan urutan dan hubungan antar proses yang terdapat dalam sistem dengan menggunakan simbol-simbol tertentu. Dalam penelitian ini, *flowchart* sistem terdiri dari *flowchart* pembangkitan kunci algoritma *Rabin-p*, *flowchart* proses enkripsi dan kompresi, dan *flowchart* proses dekompresi dan dekripsi. Adapun untuk *flowchart* sistem dapat dilihat pada gambar 2.



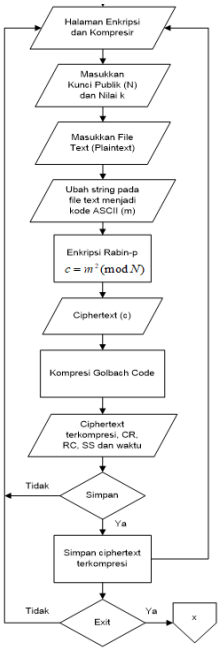
Gambar 2. Flowchart Sistem

Proses pembangkitan kunci (*key generator*) dengan algoritma *Rabin-p* dilakukan dengan menentukan parameter keamanan *k* lalu mengambil bilangan secara acak untuk *p* dan *q* kemudian dicek apakah bilangan tersebut termasuk dalam persayaratan bilangan prima. Selanjutnya menghitung nilai *N* sehingga didapatkan pasangan kunci, yaitu kunci publik (*public key*) dan kunci privat (*private key*). Adapun *flowchart* pembangkitan kunci algoritma *Rabin-p* dapat dilihat pada gambar 3.



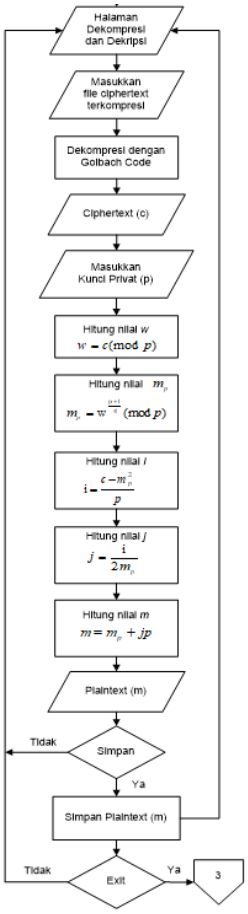
Gambar 3. Flowchart Pembangkitan Kunci Algoritma *Rabin-p*

Flowchart enkripsi dan kompresi dimulai dengan menginputkan kunci publik dan *plaintext* (file teks) yang akan di enkripsi. Kemudian plaintext tersebut akan di enkripsi dengan algoritma *Rabin-p* sehingga diperoleh hasil enkripsi (*ciphertext*) yang selanjutnya dilakukan proses kompresi dengan menggunakan algoritma *Goldbach Code* sehinnga ukuran *ciphertext* menjadi lebih kecil dan akan ditampilkan file terkompresi, RC (*Ratio Compression*), CR (*Compression Ratio*), SS (*Space Savings*) dan waktu (*running time*), kemudian disimpan. Adapun *flowchart* proses enkripsi dan kompresi dengan algoritma *Rabin-p* dan algoritma *Goldbach Code* dapat dilihat pada gambar 4.



Gambar 4. Flowchart Proses Enkripsi dan Kompresi

Flowchart proses dekompresi dan dekripsi dengan algoritma *Goldbach Code* dan algoritma *Rabin-p* dapat dilihat pada gambar 5.



Gambar 5. Flowchart Proses Dekompresi dan Dekripsi

Flowchart dekompresi dan dekripsi dimulai dengan menginputkan file terkompresi kemudian dilakukan dekompresi menggunakan algoritma *Goldbach Code* sehingga menghasilkan file hasil

dekompresi yang merupakan file *ciphertext*. Selanjutnya dilakukan proses dekripsi dengan algoritma *Rabin-p* dengan menggunakan kunci privat sehingga menjadi *plaintext* yang sama dengan *plaintext* sebelum dilakukan proses enkripsi.

3.6 Implementasi Sistem

3.6.1. Implementasi Form Key Generator

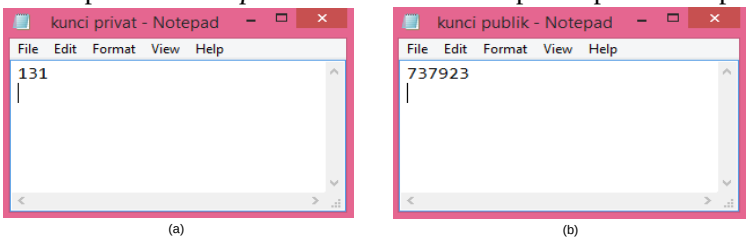
Form ini berfungsi untuk melakukan proses pembangkitan kunci (key generator) dari algoritma Rabin-p. Kunci yang dibangkitkan terdiri dari sepasang kunci, yakni kunci publik (public key) yang digunakan untuk melakukan proses enkripsi pada file teks dan kunci privat (private key) yang digunakan untuk melakukan proses dekripsi. Gambar 6 merupakan tampilan dari form key generator.

Gambar 6. Tampilan Form Key Generator

Proses membangkitkan kunci publik (*public key*) dan kunci privat (*private key*) algoritma asimetris *Rabin-p* dimulai dengan memilih tombol "Bangkitkan Kunci", sistem akan mengacak bilangan prima p dan q yang digunakan untuk menghasilkan pasangan kunci publik dan kunci privat. Adapun hasil dari pasangan kunci yang dibangkitkan seperti terlihat pada gambar 7.

Gambar 7. Hasil Pengujian Pasangan Kunci yang Dibangkitkan

Sesuai gambar 7, nilai bilangan prima yang dihasilkan setelah diacak untuk $p = 131$ dan $q = 43$. Sedangkan untuk nilai $N = 737923$ yang merupakan hasil dari p^2q. Kunci publik $N = 737923$ dan kunci privat merupakan $p = 131$. Pasangan kunci yang berhasil dibangkitkan selanjutnya akan disimpan untuk keperluan proses enkripsi dan dekripsi. Untuk menyimpan pasangan kunci dapat dilakukan dengan memilih tombol "Simpan Kunci" dan selanjutnya sistem akan menampilkan *diaog box* untuk proses menyimpan kunci. Adapun tampilan dari pasangan kunci publik dan kunci privat *Rabin-p* setelah berhasil disimpan dapat dilihat pada gambar 8.

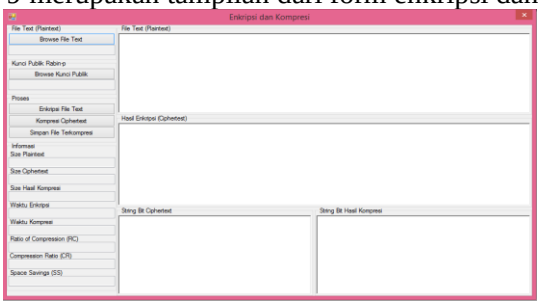


Gambar 8. Hasil Kunci Algoritma Rabin-p (a) Kunci Publik (b) Kunci Privat

3.6.2. Implementasi Form Enkripsi dan Kompresi

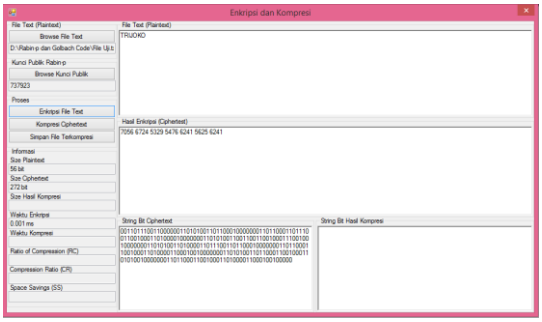
Form ini berfungsi untuk melakukan proses enkripsi pada file teks yang diinputkan oleh pengguna dengan menggunakan kunci publik dari algoritma asimetris Rabin-p yang telah dibangkitkan

sebelumnya. Hasil enkripsi (ciphertext) kemudian akan dikompresi yang bertujuan untuk memperkecil ukuran data pada ciphertext dengan menggunakan algoritma kompresi Goldbach Code. Setelah ciphertext berhasil dikompresi, sistem akan menampilkan hasil kompresi beserta informasi mengenai jumlah bit, Ratio of Compression, Compression of Ratio, Space Savings, waktu kompresi. Gambar 9 merupakan tampilan dari form enkripsi dan kompresi.



Gambar 9. Tampilan Form Enkripsi dan Kompresi

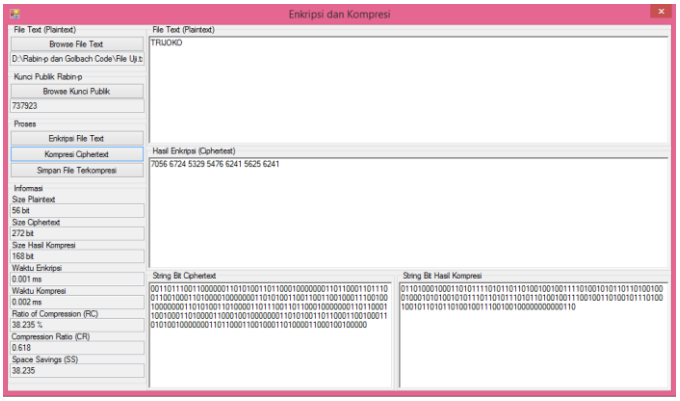
Setelah memilih tombol "Enkripsi" pada halaman utama, maka sistem akan menampilkan halaman enkripsi yang digunakan untuk proses enkripsi dan kompresi. Proses enkripsi dimulai dengan memilih tombol "*Browse File Text*" untuk mengimport *file* teks dengan memilih *format file .txt* atau *.docx* yang akan dienkripsi yang selanjutnya sistem akan menampilkan preview isi pada *file* teks. Setelah itu dilanjutkan dengan memasukkan kunci publik *Rabin-p* dengan memilih tombol "Browse Kunci Publik" dan sistem akan menampilkan kunci yang di inputkan. Setelah memasukkan *file* teks dan kunci publik, maka langkah selanjutnya adalah memilih tombol "Enkripsi File Text" dan sistem akan menampilkan hasil enkripsi (*ciphertext*) seperti terlihat pada gambar 10.



Gambar 10. Hasil Pengujian Enkripsi

Gambar 4.8 merupakan hasil pengujian enkripsi pada *file* teks dengan menggunakan kunci publik algoritma *Rabin-p*. Pada contoh pengujian ini menggunakan *plaintext* "TRIJOKO" dan kunci publik $N = 737923$. *File* teks yang di uji menggunakan sampel data dengan *size* 56 *bit*. Karakter (*string*) yang terdapat pada *file* teks yang telah di enkripsi akan menghasilkan *ciphertext* dalam bentuk bilangan desimal "7056 6724 5329 5476 6241 5625 6241" dengan *size* 272 *bit* serta estimasi waktu enkripsi selama 0.001 ms (*millisecond*). Dapat dilihat bahwa panjang karakter dari proses enkripsi dengan menggunakan algoritma *Rabin-p* akan menghasilkan *ciphertext* yang dua kali lebih besar dari *file* aslinya (*plaintext*). Oleh karena itu algoritma kompresi diperlukan guna memperkecil ukuran data dari hasil proses enkripsi sehingga dapat menghemat ruang penyimpanan data (*storage*) jika akan disimpan. Pada penelitian ini, hasil enkripsi berupa *ciphertext* kemudian akan dikompresi dengan menggunakan algoritma *lossless compression Goldbach Code* untuk tujuan memperkecil ukuran dari *ciphertext*.

Proses kompresi merupakan lanjutan dari proses enkripsi sebelumnya yang berfungsi untuk memperkecil ukuran dari hasil enkripsi (*ciphertext*). Proses kompresi dapat dilakukan dengan terlebih dahulu memilih tombol "Kompresi Ciphertext" dan sistem akan menampilkan hasil kompresi seperti terlihat pada gambar 11.

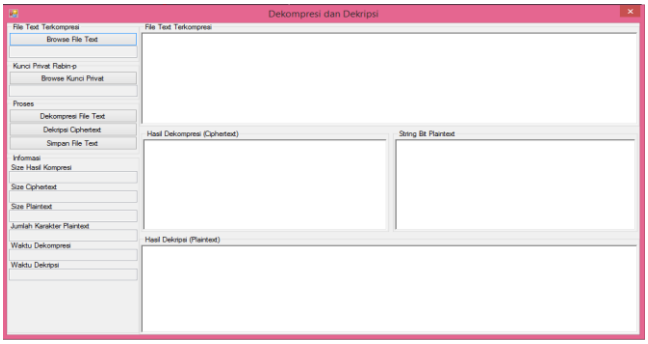


Gambar 11. Hasil Pengujian Kompresi

Pada gambar 11, setelah *ciphertext* dikompresi dengan menggunakan algoritma *Goldbach Code* maka ukuran datanya dapat diperkecil, yang mana ukuran *ciphertext* sebelum dikompresi adalah 272 *bit* dan setelah dikompresi menjadi 168 *bit*. Berdasarkan parameter yang digunakan dalam proses kompresi pada penelitian ini, diperoleh *Ratio of Compression* (RC) sebesar 38,235, *Compression of Ratio* (CR) sebesar 0,618, *Space Savings* (SS) sebesar 38,235% dan waktu kompresi selama 0.002 ms (*millisecond*). Hasil dari kompresi kemudian akan disimpan untuk tujuan pengjian pada proses dekompresi. Untuk menyimpan hasil kompresi dapat dilakukan dengan memilih tombol "Simpan File Terkompresi" dan selanjutnya sistem akan menampilkan *dialog box* untuk proses menyimpan hasil kompresi. Hasil proses kompresi berbentuk *string bit* dan akan disimpan dengan format *file .gc* (akronim dari algoritma *Goldbach Code*).

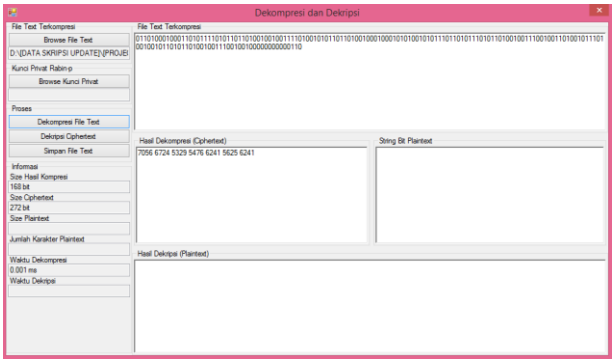
3.6.3. Implementasi Form Dekompresi dan Dekripsi

Form ini berfungsi untuk melakukan proses dekompresi pada file teks hasil kompresi sebelumnya dengan menggunakan algoritma Goldbach Code. Hasil dekompresi selanjutnya akan di dekripsi untuk mendapatkan kembali file teks aslinya dengan menggunakan kunci privat algoritma Rabin-p yang telah dibangkitkan sebelumnya. Gambar 12 merupakan tampilan dari form dekompresi dan dekripsi.



Gambar 12. Tampilan Form Dekompresi dan Dekripsi

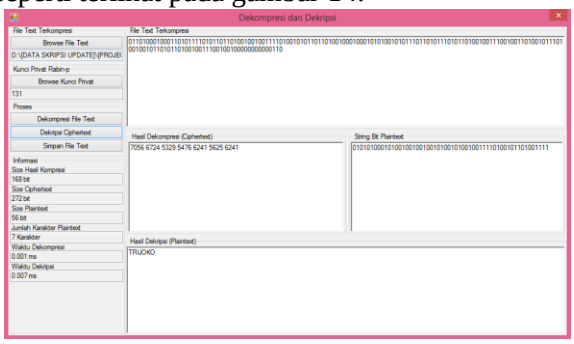
Proses dekompresi merupakan tahapan yang dilakuan untuk mengembalikan data dari hasil kompresi. Proses dekompresi dimulai dengan memilih tombol "*Browse File Text*" untuk mengimport *file* hasil kpmpresi sebelumnya dan sistem akan menampilkan preview dari *file* teks terkompresi. Selanjutnya pilih tombol "Dekompresi File Text" untuk memulai proses dan sistem akan menampilkan hasil dekompresi seperti terlihat pada gambar 13.



Gambar 13. Hasil Pengujian Dekompresi

Gambar 13 merupakan hasil pengujian dari proses dekompresi yang bertujuan untuk mengembalikan hasil kompresi sebelumnya yang merupakan *file* hasil enkripsi (*ciphertext*). Pengujian dekompresi menunjukkan bahwa *size* kompresi sebelumnya adalah 168 *bit* dan setelah di dekompresi akan menghasilkan size 272 *bit* yang sama dengan *size ciphertext* yang telah dikompresi sebelumnya dengan waktu dekompresi selama 0.001 ms (*millisecond*). Hal ini menunjukkan bahwa fungsional sistem berjalan sesuai dengan yang diharapkan, yang mana setelah proses dekompresi, jumlah *bit* dalam keseluruhan *file* teks hasil dekompresi sama persis dengan jumlah *bit* pada *file* teks aslinya.

Setelah melakukan dekompresi dengan algoritma *Goldbach Code* maka akan menghasilkan *ciphertext* yang selanjutnya akan di dekripsi dengan menggunakan kunci privat algoritma *Rabin-p*. Proses dekripsi dimulai dengan memasukkan kunci privat algoritma *Rabin-p* dengan memilih tombol "Browse Kunci Privat". Setelah itu pilih tombol "Dekripsi Ciphertext" dan sistem akan menampilkan hasilnya seperti terlihat pada gambar 14.



Gambar 14. Hasil Pengujian Dekripsi

Gambar 14. merupakan hasil pengujian dari proses dekripsi setelah dilakukan dekompresi sebelumnya. Hasil dekompresi akan menghasilkan *ciphertext*, sehingga perlu didekripsi untuk mendapatkan isi dari *file* teks aslinya. Setelah proses dekripsi jumlah *bit* dan panjang karakter hasil dekripsi (*plaintext*) sama persis dengan *file* teks aslinya yaitu 56 *bit* dengan waktu dekripsi selama 0.007 ms (*millisecond*).

4. KESIMPULAN

Berdasarkan hasil analisis, implementasi, dan pengujian sistem, maka kesimpulan yang dapat diperoleh dari penelitian sistem kriptokompresi menggunakan algoritma asimetris *Rabin-p* dan algoritma *lossless compression Goldbach Code*, yaitu sistem kriptokompresi dalam penelitian diterapkan dengan cara mengenkripsi data pada *file text* menggunakan kunci publik dari algoritma *Rabin-p* yang akan menghasilkan *file text* terenkripsi. Selanjutnya dikompresi menggunakan algoritma *lossless compression Goldbach Code* untuk memperkecil ukuran *file text* terenkripsi. Dengan demikian sistem kriptokompresi dapat meningkatkan keamanan data serta memperkecil

ukurannya. Data pada dokumen *file text* dari hasil sistem kriptokompresi mampu menjaga kerahasiaan sebuah informasi karena telah dienkripsi serta dapat menghemat ruang penyimpanan yang lebih efisien karena sudah dikompresi terlebih dahulu. *File text* yang telah dienkripsi dan dikompresi dapat dikembalikan lagi kedalam *file* aslinya melalui proses dekompresi dan dekripsi sesuai dengan *size* dan jumlah karater yang terdapat pada *file text* aslinya.

DAFTAR PUSTAKA

Volume, J., Tahun, N., Pendidikan, J., & Huda, I. A. (2020). *Perkembangan Teknologi Informasi dan Komunikasi (TIK) Terhadap Kulaitas Pembelajaran Di Sekolah Dasar*. 1–6.

Pradana, H. A., Sylfania, D. Y., & Juniawan, F. P. (2020). *Perbandingan kinerja RSA dan AES terhadap kompresi pesan SMS menggunakan algoritme Huffman Performance comparison of RSA and AES to SMS messages compression using*. 1–8. https://doi.org/10.14710/jtsiskom.2020.13468

Asbullah, M. A., & Kamel, M. R. (2016). *Design of Rabin-Like Cryptosystem without Decryption Failure i*. 1–18.

Saputro, T. H., Hidayati, N., Studi, P., Informasi, T., Magister, P., & Yogyakarta, U. T. (2018). *Survei tentang algoritma kriptografi asimetris*. 1–7.

Yogie, M. (2018). *PENERAPAN ALGORITMA GOLDBACH CODES PADA KOMPRESI FILE GAMBAR TERENKRIPSI VIGENERE CIHPER*. *7*(1), 1–6.

Tanjung, A. S., & Nasution, S. D. (2020). *Comparison Analysis with Huffman Algorithm and Goldbach Codes Algorithm in File Compression Text Using the Method Exponential Comparison*. 5–11.

Gifshuffle, K., Darwis, D., Prabowo, R., Hotimah, N., Indonesia, U. T., Komputer, J. I., & Lampung, U. (2018). *KOMBINASI GIFSHUFFLE , ENKRIPSI AES DAN KOMPRESI DATA HUFFMAN*. 1–7.

Buyung, O., & Hasugian, S. (2017). *Page 1*. 1–17.

Saputro, T. H., Hidayati, N., Studi, P., Informasi, T., Magister, P., & Yogyakarta, U. T. (2018). *Survei tentang algoritma kriptografi asimetris*. 1–7.

Budiman, M. A., & Saputra, M. Y. (2020). *Data science*. *June*, 1–12.

Apriyanto, M., Teknik, F., Komputer, I., & Darma, U. B. (n.d.). *ANALISA PENERAPAN ALGORTIMA GOLDBACH CODES DAN METODE SHANNON-FANO PADA KOMPRESI FILE TEKS*. *5*, 1–13.

Almurtada, I., Syahrizal, M., Pendahuluan, I., & Cipher, A. A. H. (2018). *PENERAPAN ALGORITMA GOLDBACH CODES PADA KOMPRESI FILE TEKS TERENKRIPSI HILL CIHPER*. 1–7.

Irliansyah, M. R., Nasution, S. D., & Ulfa, K. (2017). *PENERAPAN METODE DEFLATE DAN ALGORITMA GOLDBACH CODES*. 1–5.

Andri, M., & Muisa, E. (2019). *Analisis Perbandingan Kinerja Algoritma Start / Stop Code dan Algoritma Goldbach G1 Code pada Kompresi File Teks*. 5–6.

Pramadi, A. A., Nasution, S. D., Purba, B., Event, A., & Code, R. (2019). *PENERAPAN ALGORITMA EVEN-RODEH PADA APLIKASI KOMPRESI FILE*. 1–13. https://doi.org/10.30865/komik.v3i1.1570

Tanjung, A. S., & Nasution, S. D. (2020). *Comparison Analysis with Huffman Algorithm and Goldbach Codes Algorithm in File Compression Text Using the Method Exponential Comparison*. 5–11.