Jurnal Sains Teknologi dan Sistem Informasi
journal.yp3a.org/index.php/satesi
Vol. 1 No. 1 (2021) 01 - 06 ISSN Media Elektronik: xxxx-xxxx

Penerapan Algoritma Hybrid Affine Cipher dan RSA Terhadap Sistem Keamanan

Togar Timoteus Gultom

Teknik Informatika, STMIK ITMI Medan
Email: timoteustogar@gmail.com

ABSTRACT

The higher the value of a data, the higher the risk of damage or loss of the data. This is understandable, because high-value data is like "life" for humans, so if a data has been damaged or even lost, then the data has no meaning anymore. There are many ways to protect data. Data protection uses a method called encryption process which encodes the code in the original text into arbitrary characters. So that the contents of the encrypted data can be read again, it is necessary to change the contents of the data to its original form, this process is called decryption. In this study, a File Security System Protection Application Design was made using a hybrid affine cipher and the RSA algorithm. This application is made using the Delphi 7.0 Programming Language.

Keywords: File Security System, Data Protection, affine hybrid cipher and RSA.

ABSTRAK

Semakin tinggi nilai suatu data, semakin tinggi pula resiko akan kerusakan atau kehilangan data tersebut. Hal ini dapat dimaklumi, karena data yang bernilai tinggi ibarat “nyawa” bagi manusia, sehingga bila suatu data sudah mengalami kerusakan atau bahkan hilang, maka data tersebut sudah tidak ada artinya lagi. Ada banyak cara untuk melakukan proteksi data. Proteksi data menggunakan suatu cara yang disebut dengan proses enkripsi yang menyandikan kode pada teks asli menjadi karakter sembarang. Supaya isi data yang dienkripsi dapat dibaca kembali, maka harus dilakukan perubahan isi data ke bentuk aslinya, proses ini disebut dengan dekripsi. Pada penelitian ini dibuat sebuah Perancangan Aplikasi Perlindungan Sistem Keamanan File dengan menggunakan algoritma hybrid affine cipher dan RSA. Aplikasi ini dibuat dengan menggunakan Bahasa Pemrograman Delphi 7.0.

Kata Kunci: Sistem Keamanan File, Perlindungan Data, affine hybrid cipher dan RSA.

1. Pendahuluan

Kerahasiaan sebuah data menjadi sangat penting jika informasi didalamnya terdapat hal-hal yang tidak boleh diketahui oleh publik. Upaya pembobolan data sering dilakukan pada saat melakukan pengiriman data. Data sering disimpan dalam bentuk *file* dokumen, PDF, ataupun disisipkan dalam sebuah gambar ataupun video. Masalah keamanan dan kerahasiaan data merupakan hal yang sangat penting dalam suatu organisasi maupun pribadi. Apalagi jika data tersebut berada dalam jaringan komputer yang terhubung/terkoneksi dengan jaringan lain. Hal tersebut tentu saja akan menimbulkan resiko jika informasi yang sensitif dan berharga tersebut diakses oleh orang-orang yang tidak berhak. Yang mana jika hal tersebut sampai terjadi, kemungkinan besar akan merugikan bahkan membahayakan orang yang mengirim pesan atau menerima pesan, maupun organisasinya. Informasi yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima pesan. Selain itu data yang dibajak tersebut akan memiliki kemungkinan rusak bahkan hilang dan akan menimbulkan kerugian material yang besar.

Oleh karena itu, untuk menghindari agar hal tersebut tidak terjadi, digunakanlah sebuah program khusus proteksi/enkripsi data. Saat ini banyak beredar program khusus proteksi data, pada umumnya program tersebut memiliki, beberapa jenis metode sehingga kita dapat memilih metode yang menurut kita paling aman.

Kriptografi telah menjadi suatu bagian yang tidak dapat dipisahkan dari sistem keamanan jaringan. Ada

Diterima Redaksi: 30-08-2021 | Selesai Revisi: 01-09-2021 | Diterbitkan Online: 10-09-2021

berbagai algoritma kriptografi yang sekarang ini telah dan sedang dikembangkan, salah satunya adalahAlgoritma LUC. Algoritma LUC merupakan metode kriptografi dengan menggunakan dua kunci yang berbeda untuk pengamanan data. Algoritma LUC dilakukan dalam bentuk bilangan, oleh karena itu sebelum dilakukan proses enkripsi,teks akanterlebih dahulu dikonversikan kedalam bentuk angka.

Berdasarkan masalah diatas maka peneliti tertarik untuk melakukan penelitian dengan judul penerapan algoritma *hybrid affine cipher* dan rsa terhadap perlindungan sistem keamanan.

Beberapa penelitian yang pernah dilakukan oleh peneliti, dirangkum dalam tinjauan pustaka.

[1] Penelitian yang lakukan oleh Romindo dengan judul "Analisa Perbandingan Algoritma Mono alphabetic Cipher Dengan Algoritma One Time Pad Sebagai Pengamanan Pesan Teks". Penelitian ini membahas tentang perbandingan fungsi ciphering adalah satu ke satu, sementara algoritma One Time Pad memiliki sifat bahwa panjang plainteks (pesan) harus sama panjang dengan kunci. Algoritma monoalphabetic chiper memiliki kelemahan pada ciphertext -nya, yaitu beberapa huruf masih sama dengan plaintext, sedangkan algoritma One Time Pad berbeda dengan monoalphabetic chiper.

[2] Penelitian yang dilakukan oleh Jamaludin dan Romindo dengan judul "Implementation of Combination Vigenere Cipher and RSA in Hybrid Cryptosystem for Text Security" Tujuan dari ini penelitian adalah menghitung dan mengimplementasikan sehingga dapat diterapkan untuk mengatasi kelemahan yang terjadi dari kedua jenis kriptografi.

[3] Penelitian yang dilakukan oleh Lisda Juliana Pangaribuan dengan judul "Kriptografi Hybrida Agloritma Hill Cipher Dan Rivest Shamir Adleman (RSA) Sebagai Pengembangan Kriptografi Kunci Simetris (Studi Kasus : Nilai Mahasiswa Amik MBP)". Penelitian ini bertujuan untuk menganalisis kinerja algoritma kriptografi terhadap peningkatan keamanan nilai sehingga nilai tidak dapat dibajak oleh kriptanalis. Algoritma yang digunakan adalah modifikasi algoritma hill cipher dengan menggunakan hybrid cryptosystem untuk menyembunyikan kunci.

[4] Penelitian dilakukan oleh Iskandar Muda Siregar dengan judul "Penerapan Algoritma Affine Cipher Dan Algoritma Coloumnar Transposition Dalam Keamanan Teks". Dimana algoritma affine cipher mengalihkan plainteks dengan sebuah nilai dan menambahkannya dengan sebuah pergeseran dengan enkripsi dan dekripsi kemudian algortima coloumnar transposition berfungsi memutasi karakter diamana plainteks disusun kearah kanan sehingga keluaran cipher mengikuti urutan kunci menurun kebawah dan apabila jumlah karakternya kurang, maka dapat ditambahkan dengan katrakter yang unik untuk menutupi jejak dengan tujuan mempersulit analisa cipher dan susah dipecahkan oleh orang lain.

[5] Penelitian yang dilakukan oleh Sebastian Suhandinata dkk dengan judul "Analisis Performa Kriptografi Hybrid Algoritma Blowfish Dan Algoritma Rsa". Algoritma hybrid memiliki performa yang tidak jauh berbeda dari algoritma Blowfish dan membuat proses enkripsi dan dekripsi data lebih aman dengan keunggulan dari algoritma RSA. Rata-rata kecepatan enkripsi algoritma hybrid untuk dokumen 0,85 detik, gambar 1,06 detik, audio 3,38 detik, dan video 15,56 detik. Sedangkan rata-rata kecepatan dekripsi algoritma hybrid untuk dokumen 1,01 detik, gambar 1,38 detik, audio 4,3 detik, dan video 27,56 detik.

[6] Penelitian yang dilakukan oleh Fifit Alfiah dkk dengan judul "Aplikasi Kriptografi Dengan Menggunakan Algoritma Elgamal Berbasis Java Desktop Pada Pt. Wahana Indo Trada Nissan Jatake". Tujuan dari penelitian ini untuk menemukan cara mengimplementasikan algoritma kriptografi ElGamal ke dalam bentuk aplikasi yang nyata serta menghasilkan aplikasi pengamanan data berbasis desktop yang mudah dimengerti dan digunakan oleh user.

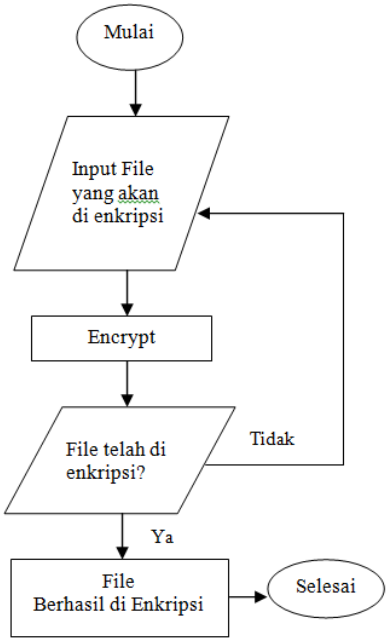
2. Metode Penelitian

2.1. *Flowchart* Sistem

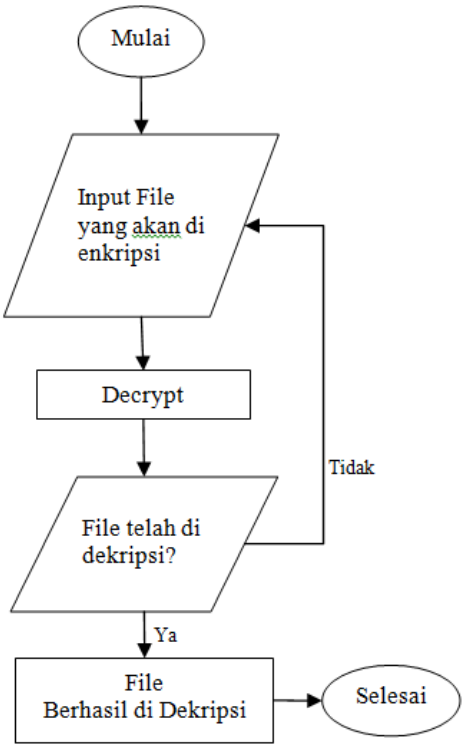
Berikut ini adalah bentuk-bentuk *flowchart* atau diagram alur yang menjelaskan alur proses aplikasi pengamanan data dari awal proses hingga akhir, termasuk proses pengenkripsian data dan mendekripsikan data itu kembali.

Pada gambar 1 dan 2 berikut, proses enkripsi dapat dijelaskan bahwa aplikasi memulai dengan menampilkan form utama dan masuk pada menu enkripsi dimana *user* dapat menginput *file* yang ingin di enkripsi. Setelah itu *file* akan dienkripsi sehingga tidak dapat dibaca oleh orang-orang yang tidak berhak. Apabila *file* tersebut memang sudah terenkripsi sebelumnya, maka *file* tersebut tidak dapat di enkripsi lagi.

DOI: xxxx



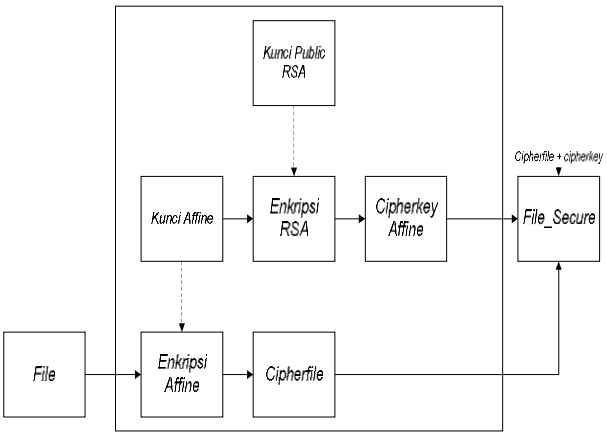
Gambar 1. Proses Enkripsi



Gambar 2. Proses Dekripsi

2.2. Arsitektur Umum Proses Enkripsi

Arsitektur Umum pada Proses Enkripsi dapat di lihat pada gambar 3:

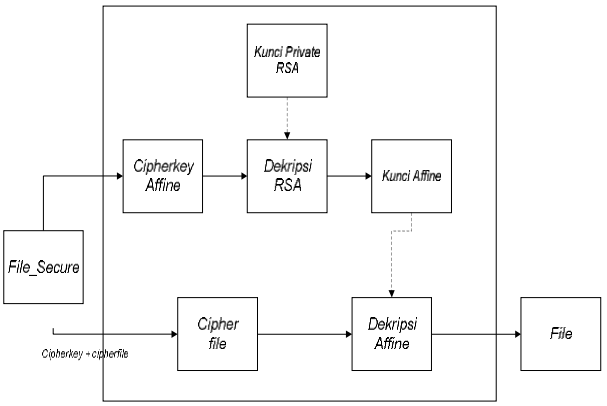


Gambar 3. Arsitektur Umum Proses Enkripsi

Pada Proses enkripsi, pertama-tama input *file* yang akan di enkripsi, kemudian data tersebut akan di enkripsi menggunakan kunci *affine* menghasilkan *cipherfile*, Kunci *affine* akan di enkripsi menggunakan kunci *public* yang menghasilkan *cipherkey affine*, sehingga *cipherfile* dan *cipherkey* ini menghasilkan *file secure* yang dikunci dengan aman.

2.3. Arsitektur Umum Proses Dekripsi

Arsitektur Umum pada Proses Dekripsi dapat di lihat pada gambar 4:



Gambar 4. Arsitektur Umum Proses Dekripsi

Pada proses dekripsi, *file_secure* yang telah di enkripsi tadi akan kita buka kembali. *Cipherkey* yang telah diterima akan di dekripsi dengan menggunakan kunci privat. Hasil dekripsi tersebut menghasilkan kunci *affine*. Kemudian *Cipherfile* dapat di dekripsi kembali oleh kunci *affine* tersebut menghasilkan data semula.

3. Hasil dan Pembahasan

3.1. Proses Enkripsi dan Dekripsi *Hybrid Affine Cipher* dan RSA.

DOI: xxxx

Proses enkripsi dan dekripsi kombinasi *Affine Cipher* dan RSA secara manual adalah sebagai berikut :

a. Proses Enkripsi

1. Diberikan sebuah *plaintext* "AB" kemudian *plaintext* diubah dalam bentuk ASCII yaitu 65 dan 66.

2. *Plaintext* "AB" di enkripsi untuk mendapatkan *ciphertext*-nya yaitu dengan cara sebagai berikut:
   Menentukan kunci *affine* (a,b) Ditetapkan a = 7 , b = 10, dan q =256
   Enkripsi : $C(P) \equiv (a.P + b)(\text{mod } q)$
   $\Leftrightarrow C(P1) \equiv (7.65 + 10)(\text{mod } 256)$
   $\Leftrightarrow C(P1) \equiv (455 + 10)(\text{mod } 256)$
   $\Leftrightarrow C(P1) \equiv 465 \text{ (mod } 256)$
   $\Leftrightarrow C(P1) \equiv 209$
   Enkripsi : $C(P) \equiv (a.P + b)(\text{mod } q)$
   $\Leftrightarrow C(P2) \equiv (7.66 + 10)(\text{mod } 256)$
   $\Leftrightarrow C(P2) \equiv (462 + 10)(\text{mod } 256)$
   $\Leftrightarrow C(P2) \equiv 472 \text{ (mod } 256)$
   $\Leftrightarrow C(P2) \equiv 216$
   Didapat *ciphertext* enkripsi *affine* adalah **(209,216)**

3. Enkripsi kunci *affine* (7,10) menggunakan kunci publik RSA
   - ditetapkan p dan q adalah 3 dan 11 (11 mod 3=1)
   - $\phi(n) = (p - 1)(q - 1) = (2)(10) = 20$
   - $n = p.q = 3 . 11 = 33$
   - Ambil secara acak kunci p dengan syarat $1 < e < n$
   - Sehinga didapat kunci publik RSA (e,n) = (3,33)

   Proses enkripsinya :
   $Ci = Pi^e \text{ mod } n$
   $C1 = 7^3 \text{ mod } 33 = 343 \text{ mod } 33 = 13$
   $C2 = 10^3 \text{ mod } 33 = 1000 \text{ mod } 33 = 10$
   Sehingga menghasilkan *cipherkey affine* yaitu **(13,10)**

b. Proses Dekripsi

Cipherkey affine yang diketahui (13,10)
Cipherkey affine ini di dekripsi menggunakan kunci privat RSA :

- Mencari kunci privat RSA
- Kunci d
  $d\ e = 1 \text{ (mod } \phi(n))$

Tabel 1. Perhitungan kunci privat "d" RSA

d	d e (mod ϕ (n))
1	3
2	6
3	9
4	12
5	15
6	18
7	1

- Didapat kunci privatnya (d,n) = (7,33)
- Dekripsi *cipherkey affine*
- $Pi = Ci^d \text{ Mod } n$
  $P1 = 13^7 \text{ mod } 33 = (13^4 \text{ mod } 33)\ (13^3 \text{ mod } 33) \text{ mod } 33 = (16)(19) \text{ mod } 33 = 304 \text{ mod } 33 = 7$
  $P2 = 10^7 \text{ mod } 33 = (10^4 \text{ mod } 33)\ (10^3 \text{ mod } 33) \text{ mod } 33$
  $= (1)(10) \text{ mod } 33 = 10$
  Didapat kunci *affine* **(7,10)**
- kunci *affine* tersebut di dekripsi dengan *affine cipher*
- $P(C) \equiv m^{-1}(C - b)(\text{mod } q)$

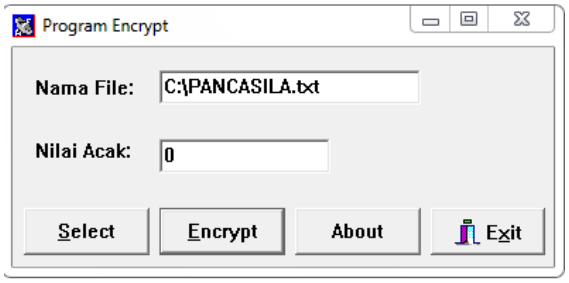
Tabel 2. Perhitungan m^{-1} *affine cipher* pada contoh *hybrid*

m-1	$m^{-1}\ m \text{ (mod q)} = m^{-1}\ 7 \text{ (mod 256)}$
1	7
2	14
3	21
4	28
5	35
. . .	. . .
183	1

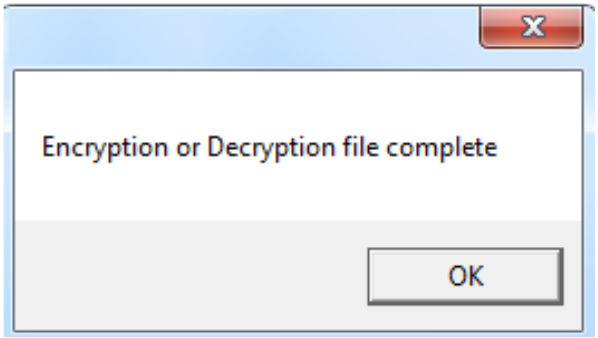
Sehingga didapat $m^{-1} = 183$
P1= 183 (C-10) mod 256
Ciphertext **(209,216)**
C1=209,
P1 = 183(209-10) mod 256
P1 = 183(199) mod 256
P1 = 36417 mod 256 P1 = 65 → A
C2= 216
P2 = 183(216-10) mod 256
P2 = 183(206) mod 256 P2 = 37698 mod 256
P2 = 66 → B
Sehingga didapat *plaintext*-nya : **A B**

3.2. Tampilan Aplikasi Keamanan File Metode Sistem Bilangan

DOI: xxxx

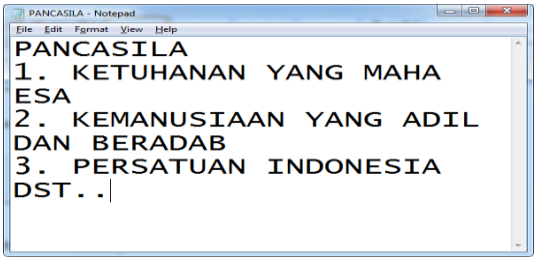


Gambar 5. Tampilan Input Enkripsi

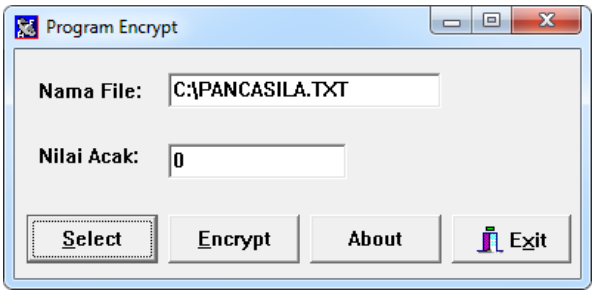


Gambar 6. Tampilan Hasil Enkripsi yang Sukses

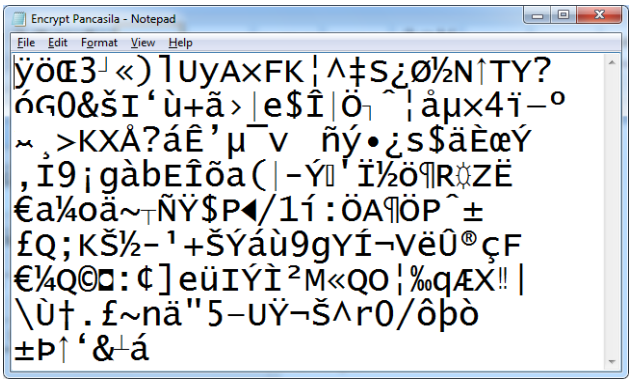
3.3. Tampilan Hasil File Enkripsi



Gambar 7. Tampilan File Pancasila.Txt sebelum di Enkripsi.

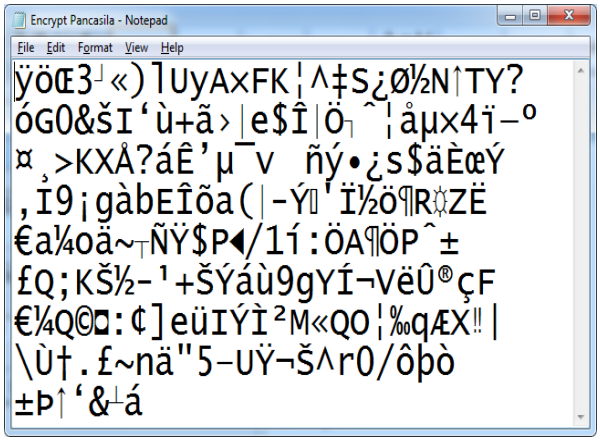


Gambar 8. Tampilan Input File Pancasila yang akan diEnkripsi

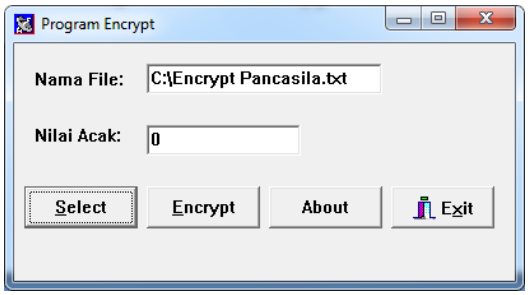


Gambar 9. Tampilan Hasil Enkripsi File Pancasila.Txt

3.4. Tampilan Hasil File Dekripsi

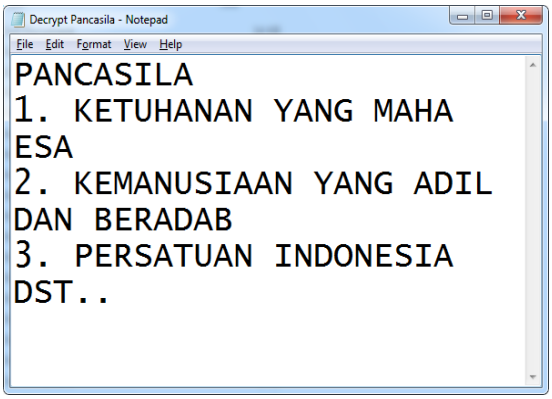


Gambar 10. Tampilan File Encrypt Pancasila sebelum di Decrypt



Gambar 11. Tampilan Input File Encrypt Pancasila.Txt yang akan di Decrypt

DOI: xxxx



Gambar 11. Tampilan Hasil Decrypt dari File Encrypt Pancasila.Txt

4. Kesimpulan

Berdasarkan analisis dari sistem dan pengujian sistem secara menyeluruh yang dilakukan pada bab-bab sebelumnya, maka ada beberapa hal yang dapat dijadikan kesimpulan. Ukuran *file* mempengaruhi lamanya proses enkripsi. Semakin besar ukuran *file*, maka akan semakin lama pula proses enkripsinya. Enkripsi hanya dapat dilakukan pada *Text File* saja. Hasil Enkripsi maupun Dekripsi dapat disimpan pada file Ms-Word selain Text File yang di NotePad.

Referensi

[1] R. Romindo, "Analisa Perbandingan Algoritma Monoalphabetic Cipher Dengan Algoritma One Time Pad Sebagai Pengamanan Pesan Teks," *Sink. J. dan Penelit. Tek. Inform.*, vol. 2, no. 2, pp. 62–66, 2018.

[2] J. Jamaludin and R. Romindo, "Implementation of Combination Vigenere Cipher and RSA in Hybrid Cryptosystem for Text Security," *Int. J. Inf. Syst. Technol.*, vol. 4, no. 1, pp. 471–481, 2020.

[3] L. J. Pangaribuan, "Kriptografi Hybrida Agloritma Hill Cipher Dan Rivest Shamir Adleman (RSA) Sebagai Pengembangan Kriptografi Kunci Simetris (Studi Kasus : Nilai Mahasiswa Amik MBP)," *J. Teknol. dan Komun.*, vol. 7, no. 1, pp. 11–26, 2018.

[4] I. Muda Siregar, "Penerapan Algoritma Affine Cipher Dan Algoritma Coloumnar Transposition Dalam Keamanan Teks," *J. Inform. Kaputama*, vol. 3, no. 1, pp. 6–12, 2019.

[5] S. Suhandinata, R. A. Rizal, D. O. Wijaya, P. Warren, and S. Srinjiwi, "Analisis Performa Kriptografi Hybrid Algoritma Blowfish Dan Algoritma Rsa," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 6, no. 1, pp. 1–10, 2019, doi: 10.33330/jurteksi.v6i1.395.

[6] F. Alfiah, R. Sudarji, and D. T. Al Fatah, "Aplikasi Kriptografi Dengan Menggunakan Algoritma Elgamal Berbasis Java Desktop Pada Pt. Wahana Indo Trada Nissan Jatake," *ADI Bisnis Digit. Interdisiplin J.*, vol. 1, no. 1, pp. 22–34, 2020, doi: 10.34306/abdi.v1i1.114.

DOI: xxxx