Implementasi Algoritma Kunci Matriks Untuk Keamanan Data File Teks Menggunakan Borland Delphi

Joko Soebagyo[1], Imay Kurniawan[2]

[1,2]*Jurusan Teknik Informatika , Sekolah Tinggi Teknologi Wastukancana Purwakarta*
Jl. Cikopak 53, Sadang, Purwakarta, Jawa Barat

[1]joko@stt-wastukancana.ac.id

[2]imaykurniawan@stt-wastukancana.ac.id

***Abstrak*— Penelitian ini bertujuan untuk menerapkan algoritma kunci matriks berordo 3×3 untuk keamanan data file teks yang merupakan modifikasi dari Hill Chiper. Gagasan dalam memodifikasi Hill Chiper dengan Delphi, menjadi tantangan bagi para peneliti terkait efisiensi waktu enkripsi dan dekripsinya. Metode perangembangan perangkat lunak yang digunakan adalah model prototype 4 fase dengan siklus lingkaran tidak berujung. Hasil dari penelitian adalah enkripsi dan dekripsi data file teks yang meliputi docx, pdf, dan pptx. Berdasarkan hasil pengujian, maka dapat disimpulkan bahwa implementasi teknik kriptografi menggunakan algoritma kunci matriks menggunakan Delphi untuk pengamanan data file teks berhasil dilakukan, dengan durasi yang cukup efektif.**

***Kata-kata kunci*— Kriptografi, Algoritma Kunci Matriks, Enkripsi, Dekripsi, Delphi**

I. PENDAHULUAN

Hill cipher merupakan algoritma enkripsi-dekripsi yang menggunakan matriks transformasi. Kekuatan Hill Cipher terletak pada kemamampuannya untuk menyembunyikan frekuensi kemunculan huruf . Ini berarti Hill Cipher cukup aman dari kriptanalis dengan analisis frekuensi huruf. Semakin besar ukuran matriks kunci maka semakin menyembunyikan frekuensi kemunculan huruf. Meskipun demikian, Hill Cipher masih dapat diserang dengan *known-plaintext attack*. Dengan mengetahui potongan berkas plainteks dan potongan berkas cipherteks maka invers matriks kunci dapat dipecahkan. Dalam penelitian ini akan dibahas implementasi dan pengujian algoritma kunci matriks yang merupakan modifikasi algoritma Hill Cipher.

II. TINJAUAN PUSTAKA

2.1. Algoritma Kunci Matriks

Algoritma kunci matriks merupakan modifikasi dari Hill Cipher. Algoritma enkripsi dapat ditunjukan dengan persamaan 2.1.

$$C = KP \ mod\ 256 \quad \text{.....2.1}$$

Dimana C merupakan cipherteks (data yang disandikan), hasil dari pengubahan data asli menjadi data yang sudah disandikan dengan menggunakan matriks kunci K. Sedangkan algoritma dekripsi dapat ditunjukan dengan persamaan 2.2.

$$P = K^{-1}C \ mod\ 256 \quad \text{.....2.2}$$

Dimana P merupakan plainteks (data asli), hasil dari pengubahan data cipherteks menjadi data asli dengan menggunakan invers matriks kunci K.

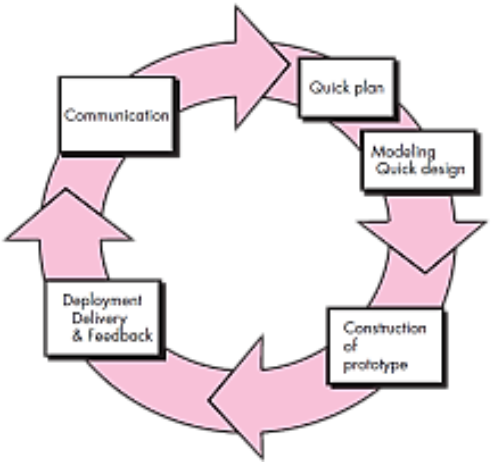
2.2. Borland Delphi

Dengan delphi kita dapat mengembangkan aplikasi berbentuk GUI, umumnya Delphi lebih banyak digunakan untuk pengembangan aplikasi desktop dan *enterprise* berbasis *database*, tapi sebagai perangkat pengembangan yang bersifat *general-purpose* ia juga mampu dan digunakan dalam berbagai jenis proyek pengembangan *software*. Bahan pertimbangan menggunakan Delphi karena penelitian ini membutuhkan eksekusi yang cepat untuk mengamankan data file teks dimana ukuran data file teks berbeda-beda ada yang kecil maupun yang besar, hal ini dapat dilakukan oleh Delphi salah satu kelebihan Delphi optimasi compiler sangat cepat.

III. Metode

3.1 Metode Pengembangan Perangkat Lunak

Pada metode pengembangan perangkat lunak, digunakan Model *Prototype.* Dengan Model *Prototype* ini pengembang dan pengguna dapat saling berinteraksi selama proses pembuatan sistem. Model ini sangat cocok untuk ruang lingkup perangkat lunak yang kecil dan memiliki waktu singkat untuk pembuatannya. Gambar Prototype dapat ditunjukan pada Gambar 3.1.



Gambar 3.1 Prototype
(Sumber : R. S Pressman, 2010)

Model *Prototype* tersebut terdiri dari fase-fase sebagai berikut:

1. C*ommunication*, pada fase ini dilakukan diskusi dengan pengguna untuk memahami prosedur yang sedang berjalan, kelemahan prosedur yang berjalan dan solusi yang dapat digunakan untuk menyelesaikan masalah.
2. *Quick Plan & Quick Design*, pada fase ini dibuat perencanaan dan desain dari perangkat lunak yang akan dikembangkan berdasarkan hasil dari fase pertama. Ciri khas dari Model Prototype adalah proses perencanaan dan desain yang cepat, karena pada model ini pengembangan aplikasi berfokus untuk menyajikan perangkat lunak secepat mungkin kepada pengguna.
3. *Construction of Prototype*, pada fase ini desain yang telah dibuat akan direalisasikan menjadi sebuah perangkat lunak.
4. *Deployment, Delivery & Feedback*, pada fase ini perangkat lunak yang telah dibuat akan dipresentasikan kepada pengguna untuk dilakukan uji coba dan diskusi kembali bila terdapat kekurangan dalam perangkat lunak tersebut.

Model Prototype digambarkan sebagai siklus lingkaran yang tidak berujung, sehingga bila pada fase *Deployment, Delivery & Feedback* ditemukan adanya ketidaksesuaian atau fungsi-fungsi yang tidak berjalan, maka proses pengembangan dapat kembali ke fase pertama untuk menganalisa kekurangan - kekurangan yang ada.

3.2 Proses Enkripsi dan Dekripsi Algoritma Kunci Matriks

Proses enkripsi merupakan proses mengubah data plainteks menjadi data cipherteks. Proses enkripsi algoritma kunci matriks menggunakan perkalian matrik.

$$C = K * P \ mod\ 256$$

$$K = \begin{vmatrix} k1 & k2 & k3 \\ k4 & k5 & k6 \\ k7 & k8 & k9 \end{vmatrix}$$

K merupakan matriks kunci.

$$P = \begin{vmatrix} p1 \\ p2 \\ p3 \end{vmatrix}$$

P merupakan data plainteks.

$$C = \begin{vmatrix} k1 & k2 & k3 \\ k4 & k5 & k6 \\ k7 & k8 & k9 \end{vmatrix} * \begin{vmatrix} p1 \\ p2 \\ p3 \end{vmatrix}$$

C merupakan data cipherteks.

Contoh proses enkripsi :
Diketahui data plainteks “Sistem”, matriks kunci

$$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 3 & 3 \end{vmatrix}$$

Kelompokan plainteks menjadi dua blok.

$$\begin{vmatrix} p1 \\ p2 \\ p3 \end{vmatrix} = \begin{vmatrix} S \\ i \\ s \end{vmatrix} \text{ dan } \begin{vmatrix} p1 \\ p2 \\ p3 \end{vmatrix} = \begin{vmatrix} t \\ e \\ m \end{vmatrix}$$

Kemudian konversi plainteks ke ascii desimal.

$$\begin{vmatrix} p1 \\ p2 \\ p3 \end{vmatrix} = \begin{vmatrix} S \\ i \\ s \end{vmatrix} = \begin{vmatrix} 83 \\ 105 \\ 115 \end{vmatrix} \quad \begin{vmatrix} p1 \\ p2 \\ p3 \end{vmatrix} = \begin{vmatrix} t \\ e \\ m \end{vmatrix} = \begin{vmatrix} 116 \\ 101 \\ 109 \end{vmatrix}$$

$$C_1 = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 3 & 3 \end{vmatrix} * \begin{vmatrix} 83 \\ 105 \\ 115 \end{vmatrix} = \begin{vmatrix} 188 \\ 408 \\ 115 \end{vmatrix} \text{ mod } 256 = \begin{vmatrix} 188 \\ 152 \\ 231 \end{vmatrix}$$

$$C_2 = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 3 & 3 \end{vmatrix} * \begin{vmatrix} 116 \\ 101 \\ 109 \end{vmatrix} = \begin{vmatrix} 217 \\ 427 \\ 746 \end{vmatrix} \text{ mod } 256 = \begin{vmatrix} 217 \\ 171 \\ 234 \end{vmatrix}$$

Gabungkan C_1 dan C_2.

$C = 188\ 152\ 231\ 217\ 171\ 234$
Konversi cipherteks ke ascii karakter
$C =$ ¼˜çÙ«ê
Dengan demikian enkripsi plainteks “Sistem” menjadi bentuk cipherteks “¼˜çÙ«ê”

Proses dekripsi merupakan proses mengubah data cipherteks menjadi data plainteks. Proses dekripsi algoritma kunci matriks menggunakan perkalian matriks.

$$P = K^{-1} * C \ mod\ 256$$

K^{-1} merupakan invers matriks kunci.

$$C = \begin{vmatrix} C1 \\ C2 \\ C3 \end{vmatrix}$$

C merupakan data cipherteks.

$$P = K^{-1} * \begin{vmatrix} p1 \\ p2 \\ p3 \end{vmatrix}$$

P merupakan data plainteks.

Contoh proses dekripsi :
Diketahui data cipherteks “¼˜çÙ«ê”, invers matriks kunci $\begin{vmatrix} 3 & -3 & 1 \\ -2 & 3 & -1 \\ 1 & -2 & 1 \end{vmatrix}$

Kelompokan data cipherteks menjadi dua blok dan konversi ke dalam bentuk ascii decimal.

$$\begin{vmatrix} c1 \\ c2 \\ c3 \end{vmatrix} = \begin{vmatrix} 188 \\ 152 \\ 231 \end{vmatrix} \text{ dan } \begin{vmatrix} c4 \\ c5 \\ c6 \end{vmatrix} = \begin{vmatrix} 217 \\ 171 \\ 234 \end{vmatrix}$$

$$P_1 = \begin{vmatrix} 3 & -3 & 1 \\ -2 & 3 & -1 \\ 1 & -2 & 1 \end{vmatrix} * \begin{vmatrix} 188 \\ 152 \\ 231 \end{vmatrix} = \begin{vmatrix} 339 \\ -151 \\ 115 \end{vmatrix} \text{ mod } 256 = \begin{vmatrix} 83 \\ 105 \\ 115 \end{vmatrix}$$

$$P_2 = \begin{vmatrix} 3 & -3 & 1 \\ -2 & 3 & -1 \\ 1 & -2 & 1 \end{vmatrix} * \begin{vmatrix} 217 \\ 171 \\ 234 \end{vmatrix} = \begin{vmatrix} 372 \\ -155 \\ 109 \end{vmatrix} \text{ mod } 256 = \begin{vmatrix} 116 \\ 101 \\ 109 \end{vmatrix}$$

Gabungkan P_1 dan P_2.
$P = 83\ 105\ 115\ 116\ 101\ 109$
Konversi plainteks ke dalam bentuk ascii karakter.
$P =$ Sistem

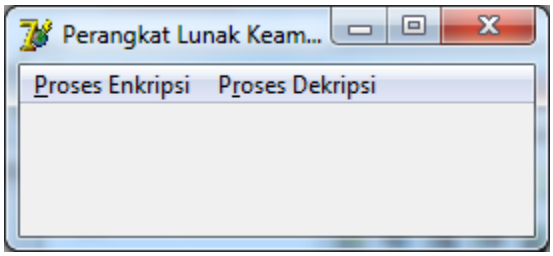
Dengan dengan hasil proses dekripsi cipherteks ““¼˜çÙ«ê” menjadi bentuk plainteks “Sistem”.

IV. Pembahasan

4.1 Implementasi

Perangkat lunak keamanan data file teks dibangun dengan menggunakan algoritma kunci matriks dan bahasa pemrograman Borland Delphi. Perangkat lunak dibagi dalam tiga bagian yaitu interface menu, interface proses enkripsi, dan proses dekripsi. File teks yang akan diuji adalah file dengan ekstensi docx,pptx, dan xlsx.

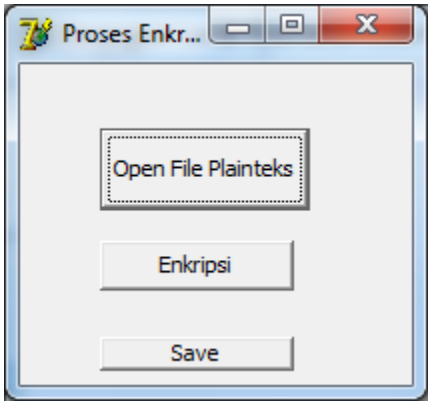
1. *Interface* Menu



Gambar 4. 1. *Interface Menu*

Untuk melakukan proses enkripsi klik tombol proses enkripsi dan untuk melakukan proses dekripsi klik tombol proses dekripsi.

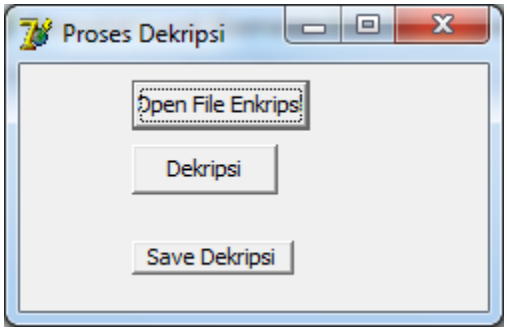
2. *Interface* Proses Enkripsi



Gambar 4. 2. *Interface* Proses Enkripsi

Tahapan untuk melakukan proses enkripsi, pertama pilih file yang akan dienkripsi dengan mengkilk tombol open file selanjutnya klik tombol enkripsi untuk memperoleh hasl enkripsi. Hasil file enkripsi dapat disimpan dengan mengklik tombol save.

3. *Interface* Proses Dekripsi



Gambar 4. 3. *Interface* Proses Dekripsi

Tahapan untuk melakukan proses dekripsi, pertama pilih file yang akan didekripsi dengan mengkilk tombol open file enkripsi selanjutnya klik tombol dekripsi untuk memperoleh hasl dekripsi. Hasil file enkripsi dapat disimpan dengan mengklik tombol save dekripsi.

4.2. Pengujian

Fle yang akan diuji ada 8 file terdiri dari 3 file docx, 3 file pdf , dan 3 file pptx. Hasil pengujian dapat ditunjukan pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Hasil Pengujian Proses Enkripsi

No	Nama File Sumber	Ukuran File	Durasi Proses Enkripsi	Nama File Enkripsi
1.	Revisi Sidang2	8.34 MB (8,755,868 bytes)	1 detik	cipherteks1
2.	STT-SKRIPSI	2.58 MB (2,715,244 bytes)	0,5 detik	Cipherteks2
3.	Rancang bagun bahasa Turki5	1.52 MB (1,596,853 bytes)	0,3 detik	Cipherteks3
4.	pdf1	1.39 MB (1,463,906 bytes)	0,1 detik	Cipherteks4
5.	pdf2	1.88 MB (1,979,253 bytes)	0,4 detik	cipherteks5
6.	pdf3	2.55 MB (2,682,134 bytes)	0,5 detik	cipherteks6
7.	DAA-III-Brute-Force	1.58 MB (1,662,392 bytes)	0,3 detik	cipherteks7
8.	modul kuliah orkom	4.03 MB (4,229,632 bytes)	0,7 detik	cipherteks8

Tabel 4.2 Hasil Pengujian Proses Dekripsi

No	Nama File Enkripsi	Ukuran File	Durasi Proses Enkripsi	Nama File Dekripsi
1.	cipherteks1	8.34 MB (8,755,868 bytes)	1 detik	plainteks1
2.	cipherteks2	2.58 MB (2,715,244 bytes)	0,5 detik	plainteks2
3.	cipherteks3	1.52 MB (1,596,853 bytes)	0,3 detik	lainteks3
4.	cipherteks4	1.39 MB (1,463,906 bytes)	0,1 detik	plainteks4
5.	cipherteks5	1.88 MB (1,979,253 bytes)	0,4 detik	plainteks5
6.	cipherteks6	2.55 MB (2,682,134 bytes)	0,5 detik	plainteks6
7.	cipherteks7	1.58 MB (1,662,392 bytes)	0,3 detik	plainteks7
8.	cipherteks8	4.03 MB (4,229,632 bytes)	0,7 detik	plainteks8

Dari hasil pengujian proses enkripsi dan dekripsi dapat ditunjukan bahwa file dekripsi dapat dikembalikan ke file data asli.

V. Kesimpulan dan Saran

5.1. Kesimpulan

Dari hasil penelitian dapat disimpulkan sebagai berikut :

1. Perangkat lunak keamanan data file teks menggunakan algoritma kunci matriks dapat diterapkan untuk mengamankan file teks ekstensi docx,pdf, dan pptx dengan durasi waktu yang cukup efektif.
2. Cipherteks hasil enkripsi algoritma kunci matriks sangat sulit dipecahkan dengan hanya mengetahui berkas potongan cipherteks. Namun dapat dengan mudah dipecahkan jika diketahui potongan plainteks dan potongan cipherteks.

5.2. Saran

Untuk pengembangan penelitian algoritma kunci matriks perlu dikombinasi dengan algoritma kriptografi yang lain untuk mengantisipasi jika diketahui potongan plainteks dan cipherteks.

Daftar Pustaka

[1] A. Edelman and G. Strang, “Pascal matrices,” *Am. Math. Mon.*, vol. 111, no. 3, pp. 189–197, 2004.

[2] Abdul Kadir, 2004, ”Dasar Aplikasi Database MySQL – Delphi”, Andi, Yogyakarta.

[3] H. Anton and C. Rorres, *Elementary Linear Algebra*, Ninth edit. John Wiley & Sons, Inc., 2010.

[4] Munir Rinaldi, 2019, Kriptografi, Informatika, Bandung.

[5] Pardede, A. M. H. (2017). Algoritma Vigenere Cipher Dan Hill Cipher Dalam Aplikasi Keamanan Data Pada File Dokumen. *Jurnal Teknik Informatika Kaputama (JTIK)*, *1*(1), 26–33 (Vol, 2008)

[6] Prabowo, H. E. and Hangga, A. (2015) ‘Enkripsi Data Berupa Teks Menggunakan Metode Modifikasi Vigenere Cipher’, *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, pp. 1–4.

[7] Pressman, Roger S. 2010. Rekayasa Perangkat Lunak. Yogyakarta: Andi

[8] Zarlis, M. (2017) ‘Implementasi Algoritma Vigenere Subtitusi dengan Shift Indeks Prima’, *Universitas Sumatera Utara*, pp. 149–154.