Implementasi Algoritma Triangle Chain Cipher dalam Penyandian Pesan

Yudi Irawan

STMIK Budi Darma Medan, Jl. Sisingamangaraja No. 338 Simpang Limun, Medan, Indonesia

ARTICLE INFORMATION

Received: September, 20, 2020
Revised: September, 29, 2020
Available online: Oktober,31 2020

KEYWORDS

Five words maximum, comma separated

KEAMANAN DATA, PESAN TEKS, TRIANGLE CHAIN CIPHER, KRIPTOGRAFI

Phone: +62 822-7576-6021

E-mail: yitheangel@gmail.com

A B S T R A K

Keamanan data telah menjadi bagian dari pengembangan teknologi informasi, mengingat bahwa berjuta-juta bit informasi telah dipertukarkan dalam jaringan komputer terutama di internet. Keamanan data yang menjamin bahwa data tidak terganggu selama proses transfer dari sumber ke tujuan melalui saluran-saluran komunikasi. Masalah integrity berkaitan dengan bagaimana melindungi data dari penyusup yang berusaha masuk ke sumber data, atau menyusup dalam jaringan data untuk mengubah dan merusak. Algoritma triangle chain cipher merupakan salah satu algoritma penyandian yang beroperasi berdasarkan penyandian (kriptografi) klasik khususnya dalam teknik subtitusi terhadap karakter. Setiap karakter akan disubtitusi berdasarkan kunci dan faktor pengali yang telah ditetapkan berdasarkan formula yang berlaku dalam algoritma ini. Algoritma ini melakukan penyandian pada pesan teks sebanyak dua kali dan selalu bergantung pada hasil proses sebelumnya. Hal inilah yang mendasari rumitnya pemecahan dari algoritma penyandian berantai ini.Penyandian pesan khususnya pada pesan teks merupakan salah satu hal yang sangat penting dilakukan dalam meningkatkan keamanan pesan teks tersebut dari berbagai tingkat kejahatan yang dilakukan oleh orang-orang yang tidak berhak dan tidak bertanggungjawab. Implementasi algoritma triangle chain cipher pada penyandian pesan teks dapat mempersulit siapa saja yang berhasil mengakses dan mendapatkan pesan teks untuk memahami dan mengerti, merusak, mendistribusikan, mencuri pesan teks atau tindakan lain yang dapat merugikan pihak-pihak pemilik pesan teks tersebut.

PENDAHULUAN

Keamanan data telah menjadi bagian dari pengembangan teknologi informasi, mengingat bahwa berjuta-juta bit informasi telah dipertukarkan dalam jaringan komputer terutama di internet. Keamanan data yang menjamin bahwa data tidak terganggu selama proses transfer dari sumber ke tujuan melalui saluran-saluran komunikasi. Masalah integrity berkaitan dengan bagaimana melindungi data dari penyusup yang berusaha masuk ke sumber data, atau menyusup dalam jaringan data untuk mengubah dan merusak. Kemajuan alat komunikasi telah membebaskan siapapun untuk berkomunikasi dengan orang yang ada di seluruh dunia jika masih terhubung dengan internet. Hal ini dikuatkan dengan bertambah banyaknya situs jejaring sosial yang dapat digunakan untuk mengirim pesan dengan sangat mudah dan gratis. Situs jejaring sosial menyediakan layanan berkomunikasi dengan bebas, layanan yang wajib adalah layanan mengirim pesan, baik pesan singkat, email ataupun metode baru yang dinamakan chatting [1], [2].

Media sosial tidak hanya populer dikalangan masyarakat saja, akan tetapi media sosial juga populer pada kalangan pemerintah, dan kelompok organisasi non formal lainnya. Hal ini terbukti banyaknya instansi pemerintah yang memanfaatkan jejaring sosial dalam kegiatan instansi mereka. Seperti mengirim dokumen atau pesan. Penggunaan media sosial sebagai perantara kegiatan komunikasi baik teks atau lainnya, tidak diketahui tingkat keamanannya. Maksudnya, pesan yang dikirim apakah dapat dilihat atau dipakai dan didapatkan oleh orang lain. Mengingat banyak sekali jenis pesan teks yang tidak boleh diketahui oleh orang lain, atau pesan rahasia. Karena media sosial yang gratis, maka pihak pengguna juga tidak dapat meminta fasilitas keamanan terhadap pesan [3], [4].

Algoritma triangle chain chiper atau umumnya dikenal dengan sebutan algoritma rantai segitiga merupakan cipher yang ide awalnya dari algoritma kriptografi one time pad, yaitu kunci yang dibangkitkan secara random dan panjang kunci sepanjang plaintext yang akan dienkripsi. Algoritma kriptografi triagle chain chiper membangkitkan kunci-kunci secara otomatis dilakukan dengan teknik berantai. Kekuatan cipher ini terletak pada kunci, yaitu nilai integer yang menunjukkan pergeseran karakter-karakter sesuai dengan operasi pada caesar cipher. Kekuatan kedua terletak pada barisan bilangan-bilangan yang berfungsi sebagai pengali dengan kunci [5]

Algoritma kriptografi triangle chain cipher atau umumnya dikenal dengan sebutan rantai segitiga merupakan cipher yang ide awalnya dari algoritma kriptografi One Time Pad, yaitu kunci yang dibangkitkan secara random dan panjang kunci sepanjang plainteks yang akan dienkripsi. Tetapi pada algoritma kriptogarfi rantai segitiga pembangkitan kunci-kunci tersebut secara otomatis dengan teknik berantai [6], [7].

Pada kenyataannya cipher substitusi segitiga tidak dibuat secara sederhana, tetapi dengan mengenkripsi ganda (mengenkripsi dua kali), jadi plainteks dienkripsi dengan cipher segitiga I, kemudian hasil enkripsi pertama dienkripsi kembali dengan cipher segitiga II yang arah segitiga II merupakan kebalikan arah segitiga I. Secara matematis pola enkripsi rantai segitiga dapat digambarkan dengan matriks N x N dengan N merupakan panjang plainteks yang akan dienkripsi dan operasi pada alfabet ASCII [8], [9].

Oleh karena itu perlu dipikirkan suatu cara untuk membuat keamanan sendiri disamping keamanan yang disediakan oleh media sosial tersebut. Keamanan yang digunakan untuk pesan teks adalah metode kriptografi. Metode kriptografi dapat merubah pesan menjadi kode-kode yang tidak mempunyai makna atau arti sehingga orang yang tidak mempunyai kepentingan tidak akan dapat membaca pesan, walaupun mempunyai pesan tersebut. Untuk itu, perlu dibuat sebuah aplikasi personal komputer yang mampu merubah pesan menjadi kode atau chipper. Salah satu metode kriptografi yang banyak digunakan atau metode triangle chain chiper. Metode ini mempunyai tingkat pengamanan yang dalam, hingga tiga rantai.

METODE PENELITIAN

Adapun metodologi penelitian penelitian dimana matriks dilambangkan dengan Mij [10], dengan $1 \leq i \leq N$ dan $1 \leq j \leq N$, nilai integer kunci dengan K, faktor pengali merupakan tabel integer R. Plainteks dengan P dimana P merupakan tabel plainteks dengan panjang N yaitu P[N].

Berikut operasi matriks untuk proses enkripsi (Mohamad Firda Fauzan.Studi dan Implementasi Cipher Subtitusi Rantai Segitiga [6], [11].

1. Matriks enkripsi segitiga pertama
   Untuk baris ke-1:
   M1j = P[j] + (K * R[1]) mod 256
   Untuk baris ke-2 dan selanjutnya untuk nilai $j \geq i$:
   Mij = M(i-1)j +(K * R [i]) mod 256
   Sehingga nilai cipherteks yang diperoleh adalah:
   Mij pada nilai j = (N+i)-N.
2. Matriks enkripsi segitiga kedua
   Nilai P diperoleh dari nilai Mij pada i = j
   Untuk baris ke-1:
   M1j = P[j] + (K*R[1])) mod 256
   Untuk baris ke 2 dan selanjutnya untuk nilai $j \leq (N+1) - i$:
   Mij = M(i-1)j+(K*R[i])) mod 256
   Sehingga nilai cipherteks yang diperoleh adalah:
   Mij pada nilai j = (N+1)-i.
   Keterangan:
   P = Plainteks
   N = Jumlah karakter plainteks
   M = Matriks penampung hasil penyandian
   K = Kunci
   R = Row (baris perkalian faktor pengali dengan kunci)
   i = Indeks faktor pengali
   j = Indek karakter plainteks

Sedangkan untuk proses dekripsi merupakan kebalikan dari proses enkripsi Berikut operasi matriks untuk proses dekripsi.

1. Matriks dekripsi segitiga pertama operasinya merupakan kebalikan dari matriks enkripsi, jadi operasi ini kebalikan operasi matriks enkripsi segitiga kedua. Nilai C merupakan tabel dari cipherteks dengan panjang N yaitu C[N].
   Untuk baris ke-1, berlaku formula:
   $j \leq (N + 1) - i$
   M1j = C [j] - (K * R[1])) mod 256
   sedangkan untuk baris kedua dan selanjutnya dimana nilai $j \geq i$, berlaku formula : Mij = (M(i-1) j - K * (R[i])) mod 256.
   Sehingga nilai plainteks yang diperoleh adalah:
   Mij pada nilai j = (N+i)-i.
2. Matriks dekripsi segitiga kedua
   Untuk baris pertama berlaku formula:
   M1j = C [j] – (K * R[1])) mod 256
   Sedangkan untuk baris kedua dan seterusnya nilai $j \geq i$, berlaku formula
   Mij = C[i-1]j – (K * R[i])) mod 256.
   nilai plainteks yang diperoleh adalah :

Mij pada nilai j = (N+1)-i.
Sehingga nilai plainteks yang diperoleh adalah:
Mij pada nilai j = (N+i)-N.
Keterangan:
C = Cipherteks
N = Jumlah karakter cipherteks
M = Matriks penampung hasil cipher yang dijadikan sebagai plaintext
K = Kunci
R = Row (baris perkalian faktor pengali dengan kunci)
i = Indeks faktor pengali
j = Indek karakter cipherteks

HASIL DAN PEMBAHASAN

Penggunaan media sosial sebagai alat komunikasi memang memberikan kemudahan dari pada menggunakan peralatan konvensional. Penggunaan media sosial memberikan kebebasan dalam kegiatan berkomunikasi. Selain memberikan kebebasan pemanfaatan media sosial juga tidak dipungut biaya apapun atau gratis. Satu-satunya aspek yang mendukung kemajuan media sosial, sehingga banyak dipakai di kalangan luas adalah karena gratis.

Salah satu teknik kriptografi yang digunakan adalah menggunakan enkripsi, dekripsi dan kunci. Enkripsi merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirim agar terjaga kerahasiannya. Pesan asli disebut *plain*teks, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan sebagai *cipher*. Dekripsi merupakan kebalikan dari proses enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (teks asli), disebut dengan dekripsi pesan. Kunci yang dipakai untuk melakukan enkripsi dan dekripsi.

Salah satu teknik penyandian yang dapat digunakan dalam penyandian pesan adalah menerapkan algoritma *triangle chain cipher* (rantai segitiga) yang mengadopsi teknik penyadian caesar, dimana dapat melakukan subtitusi setiap karakter yang akan disandikan secara berantai berdasarkan nilai kunci dan faktor-faktor pengali yang terbentuk. Algoritma kriptografi rantai segitiga juga mengadopsi prinsip menyebar karena proses enkripsi dan dekripsi dilakukan berulang-ulang. Hasil dari proses pertama akan menjadi dasar untuk proses ke dua, hasil proses ke dua akan menjadi dasar proses ke tiga dan berlaku untuk proses selanjutnya. Kunci-kunci yang dibangkitkan secara acak dalam proses penyandian pesan dapat membentuk karakter pesan yang telah diacak atau selanjutnya disebut *cipher* yang memiliki kunci berupa bilangan *integer*. Kunci tersebut tidak dapat berjalan dengan sendirinya, kunci tersebut butuh faktor pengali. Faktor pengali itu dapat berupa deretan bilangan asli, deret bilangan ganjil, deret bilangan genap, deret bilangan prima ataupun faktor pengali yang ditetapkan sesuai dengan kemauan pengenkripsi. Hasil dari perkalian kunci dengan faktor pengali menghasilkan kunci pergeseran yang nilainya acak, sehingga menghasilkan cipherteks yang acak pula. Pembangkitan kunci secara acak, menyebabkan hasil karakter sandi pesan pun menjadi acak dan tidak ada hubungannya dengan pesan aslinya.

Berikut operasi matriks untuk proses enkripsi:

1. Matriks enkripsi segitiga pertama
   Untuk baris ke-1 :
   $M_{1j} = P[\ j\ ] + (K * R[\ 1]) \bmod 256$
   untuk baris ke-2 dan selanjutnya untuk nilai $j \geq i$:
   $M_{ij} = M_{(i-1)j} + (K * R\ [\ i\ ]) \bmod 256$

   sehingga nilai *cipher*teks yang diperoleh adalah :
   M_{ij} pada nilai j = (N+i)-N.

2. Matriks enkripsi segitiga ke dua
   Nilai P diperoleh dari nilai M_{ij} pada i = j

   Untuk baris ke-1 :
   $M_{1j} = P[\ j\ ] + (K*R[1])) \bmod 256$

   untuk baris ke 2 dan selanjutnya untuk nilai $j \leq (N+1) - i$:
   $M_{ij} = M_{(i-1)j} + (K*R[i])) \bmod 256$

   sehingga nilai *cipher*teks yang diperoleh adalah :
   M_{ij} pada nilai j = (N+1)-i.

Keterangan :
P = Plainteks
N = Jumlah karakter *plain*teks
M = Matriks penampung hasil penyandian
K = Kunci
R = *Row* (baris perkalian faktor pengali dengan kunci)
i = Indeks faktor pengali
j = Indek karakter p*laint*eks

Sedangkan untuk proses dekripsi merupakan kebalikan dari proses enkripsi Berikut operasi matriks untuk proses dekripsi :

1. Matriks dekripsi segitiga pertama operasinya merupakan kebalikan dari matriks enkripsi, jadi operasi ini kebalikan operasi matriks enkripsi segitiga ke dua. Nilai C merupakan tabel dari *cipher*teks dengan panjang N yaitu C[N].
   Untuk baris ke-1, berlaku formula :
   $j \leq (N + 1) - i$
   $M_{1j} = C [j] - (K * R[1])) \bmod 256$
   sedangkan untuk baris ke dua dan selanjutnya dimana nilai $j \geq i$, berlaku formula : $M_{ij} = (M_{(i-1)\,j} - K * (R[i])) \bmod 256$.
   sehingga nilai *plain*teks yang diperoleh adalah :
   M_{ij} pada nilai j = (N+i)-i.

2. Matriks dekripsi segitiga ke dua
   Untuk baris pertama berlaku formula :
   $M_{1j} = C [j] – (K * R[1])) \bmod 256$
   sedangkan untuk baris ke dua dan seterusnya nilai $j \geq i$, berlaku formula :
   $M_{ij} = C_{[i-1]j} – (K * R[i])) \bmod 256$.
   nilai *plain*teks yang diperoleh adalah :
   M_{ij} pada nilai j = (N+1)-i.

   sehingga nilai *plain*teks yang diperoleh adalah :
   M_{ij} pada nilai j = (N+i)-N.

Keterangan :
C = *Cipherteks*
N = Jumlah karakter *cipher*teks
M = Matriks penampung hasil *cipher* yang dijadikan sebagai *plaintext*
K = Kunci
R = *Row* (baris perkalian faktor pengali dengan kunci)
i = Indeks faktor pengali
j = Indek karakter cipherteks

a. Matrik enkripsi segitiga pertama
   Faktor pengali dengan kunci adalah deret bilangan asli (1, 2, 3,…, n).
   Langkah pertama yang dilakukan untuk proses enkripsi pertama ini adalah menentukan nilai desimal masing-masing karakter plainteks dalam ASCII :

Y	U	D	I
89	85	68	73

   Langkah ke dua adalah membentuk tabel faktor pengali :
   Jumlah deret bilangan akan disesuaikan dengan jumlah banyaknya karakter dari plainteks.
   Jadi, jumlah karakter plainteks (N) adalah 3. Deret bilangan asli (R) yang menjadi faktor pengali adalah 1, 2, 3, 4.
   Langkah ke tiga adalah melakukan proses enkripsi segitiga pertama sesuai dengan formulanya.
   Plainteks (P) = Y U D I
   N = 4
   K = 3
   R = 1,2,3,4
   Nilai desimal masing-masing karakter plainteks dalam ASCII

Y	U	D	I
89	85	68	73

   Untuk baris pertama (i = 1), maka :
   $M_{11} = (P[1] + 3 * R[1]) \bmod 256$
   $= (Y + 3 * (1)) \bmod 256$
   $= (89 + 3) \bmod 256$
   = 92 (huruf “ \ ” dalam karakter ASCII 256)

   $M_{12} = (P[2] + 3 * R[1]) \bmod 256$
   $= (U + 3 * (1)) \bmod 256$
   $= (85 + 3) \bmod 256$
   = 88 (huruf “ X ” dalam karakter ASCII 256)
   $M_{13} = (P[3] + 3 * R[1]) \bmod 256$
   $= (D + 3 * (1)) \bmod 256$
   $= (68 + 3) \bmod 256$
   = 71 (huruf “ G ” dalam karakter ASCII 256)
   $M_{14} = (P[3] + 3 * R[1]) \bmod 256$
   $= (I + 3 * (1)) \bmod 256$
   $= (73 + 3) \bmod 256$
   = 76 (huruf “ L ” dalam karakter ASCII 256)
   hasil sandi pada tahap i = 1 (baris pertama) adalah \ X G L.
   Sampai pada tahap ini hasil penyandian dapat ditunjukkan di bawah ini :
   Y U D I (nilai desimal dalam ASCII : 89 85 68 73) → i = 0

\ X G L (nilai desimal dalam ASCII : **92** 88 71 76) → i = 1

Hasil penyandian baris pertama (i = 1) akan digunakan sebagai plainteks pada proses enkripsi baris ke dua (i = 2), dimana nilai j ≥ i, sehingga :

i = 2, j = 2

M_{22}= ($M_{(2-1)2}$ + 3 * (2)) mod 256
= ($M_{(1)2}$ + 3 * (2)) mod 256
= (X + 6) mod 256
= (88 + 6) mod 256
= 94 (huruf " ^ " dalam karakter ASCII 256)

M_{23}= ($M_{(2-1)3}$ + 3 * (2)) mod 256
= ($M_{(1)3}$ + 3 * (2)) mod 256
= (G + 6) mod 256
= (71 + 6) mod 256
= 77 (huruf " M " dalam karakter ASCII 256)

M_{24}= ($M_{(2-1)4}$ + 3 * (2)) mod 256
= ($M_{(1)3}$ + 3 * (2)) mod 256
= (L + 6) mod 256
= (76 + 6) mod 256
= 82 (huruf " R " dalam karakter ASCII 256)

hasil dari enkripsi baris ke dua ini adalah ^ M R .

Hasil enkripsi sampai pada tahap ini (i = 2) dapat dilihat di bawah di bawah ini :

Y U D I (nilai desimal dalam ASCII : 89 85 68 73) → i = 0
\ X G L (nilai desimal dalam ASCII : **92** 88 71 76) → i = 1
^ M R (nilai desimal dalam ASCII : **94** 77 82) → i = 2

Hasil enkripsi pada baris ke dua (i=2) akan digunakan sebagai plainteks pada proses enkripsi baris ke tiga (i=3), sehingga :

i = 3, j = 3

M_{33}= ($M_{(3-1)3}$ + 3 * (3)) Mod 256
= ($M_{(2)3}$ + 3 * (3)) Mod 256
= (M + 9) mod 256
= (77 + 9) mod 256
= 86 (huruf " V " dalam karakter ASCII 256)

M_{34}= ($M_{(3-1)4}$ + 3 * (3)) Mod 256
= ($M_{(2)4}$ + 3 * (3)) Mod 256
= (R + 9) mod 256
= (82 + 9) mod 256
= 91 (huruf " [" dalam karakter ASCII 256)

hasil dari enkripsi baris ke tiga ini adalah V [

Hasil enkripsi sampai pada tahap ini (i = 3) dapat dilihat di bawah di bawah ini :

Y U D I (nilai desimal dalam ASCII : 89 85 68 73) → i = 0
\ X G L (nilai desimal dalam ASCII : **92** 88 71 76) → i = 1
^ M R (nilai desimal dalam ASCII : **94** 77 82) → i = 2
V [(nilai desimal dalam ASCII : **86** 91) → i = 3

Hasil enkripsi pada baris ke dua (i=3) akan digunakan sebagai plainteks pada proses enkripsi baris ke empat (i=4), sehingga :

i = 4, j = 4

M_{44}= ($M_{(4-1)4}$ + 3 * (4)) Mod 256
= ($M_{(3)4}$ + 3 * (4)) Mod 256
= ([+ 12) mod 256
= (91 + 12) mod 256
= 103 (huruf " g " dalam karakter ASCII 256)

Hasil enkripsi sampai pada tahap baris ke tiga (i = 4) dapat dilihat di bawah ini:

Y U D I (nilai desimal dalam ASCII : 89 85 68 73) → i = 0
\ X G L (nilai desimal dalam ASCII : **92** 88 71 76) → i = 1
^ M R (nilai desimal dalam ASCII : **94** 77 82) → i = 2
V [(nilai desimal dalam ASCII : **86** 91) → i = 3
g (nilai desimal dalam ASCII : **103**) → i = 4

Maka yang menjadi *cipher* pada proses enkripsi segitiga ke dua adalah \ ^ **V g** dimana dapat dilihat bahwa susunan dari baris dan kolomnya berbentuk segitiga yang mengerucut ke kiri.

b. Matrik enkripsi segitiga ke dua

Pada proses ini yang menjadi plainteks adalah *cipher* yang dihasilkan dari proses enkripsi segitiga pertama (\ ^ **V g**) kemudian dienkrip lagi sesuai dengan formula yang berlaku pada proses enkripsi segitiga ke dua.

Plainteks = \ ^ **V g** (cipher hasil enkripsi segitiga pertama)
92 94 86 103 (nilai desimal dalam ASCII)

Untuk baris pertama (i = 1) :

M_{11} = (P[1] + (3 * 1)) mod 256
= (\ + (3 * 1)) mod 256
= (92 + 3) mod 256

= 95 (huruf " _ " dalam karakter ASCII 256)

M_{12} = (P[2] + (3 * 1)) mod 256
= (^ + (3 * 1)) mod 256
= (94 + 3) mod 256
= 97 (huruf " a " dalam karakter ASCII 256)

M_{13} = (P[3] + (3 * 1)) mod 256
= (V + (3 * 1)) mod 256
= (86 + 3) mod 256
= 89 (huruf " Y " dalam karakter ASCII 256)

M_{14} = (P[3] + (3 * 1)) mod 256
= (9 + (3 * 1)) mod 256
= (103 + 3) mod 256
= 106 (huruf " j " dalam karakter ASCII 256)

hasil dari enkripsi baris pertama (i = 1) adalah _ a Y j .

Hasil enkripsi sampai pada tahap baris pertama (i = 1) dapat dilihat di bawah ini:

\ ^ **V g** (dalam nilai ASCII 92 94 86 103 → i = 0
_ a Y j (dalam nilai ASCII 95 97 89 **106** → i = 1

Hasil enkripsi baris pertama (i = 1) akan digunakan sebagai plainteks pada proses enkripsi baris ke dua dimana nilai $j \leq (N + 1) - i$, sehingga :

$i = 2;\ j \leq (4 + 1) - 2 \rightarrow j \leq 3$

M_{21} = $(M_{(2-1)1} + (K * R[i]))$ mod 256
= $(M_{(1)1} + (K * R[i]))$ mod 256
= (_ + (3 * 2)) mod 256
= (95 + 6) mod 256
= 101 (huruf " e " dalam karakter ASCII 256)

M_{22} = $(M_{(2-1)2} + (K * R[i]))$ mod 256
= $(M_{(1)2} + (K * R[i]))$ mod 256
= (a+ (3 * 2)) mod 256
= (97 + 6) mod 256
= 103 (huruf " g " dalam karakter ASCII 256)

M_{23} = $(M_{(2-1)3} + (K * R[i]))$ mod 256
= $(M_{(1)3} + (K * R[i]))$ mod 256
= (Y+ (3 * 2)) mod 256
= (89 + 6) mod 256
= 95 (huruf " _ " dalam karakter ASCII 256)

hasil dari enkripsi baris ke dua (i = 2) adalah "e g _" .

Hasil enkripsi sampai pada tahap baris ke dua (i=2) dapat dilihat di bawah ini:

\ ^ V g (dalam nilai ASCII 92 94 86 103 → i = 0
_ a Y **j** (dalam nilai ASCII 95 97 89 **106** → i = 1
e g _ (dalam nilai ASCII 101 103 **95** → i = 2

Hasil enkripsi baris ke dua (i=2) akan digunakan sebagai plainteks pada proses enkripsi baris ke tiga, sehingga :

$i = 3 ; j \leq (4 + 1) - 3 \rightarrow j \leq 2$

M_{31} = $(M_{(3-1)1} + (K * R[i]))$ mod 256
= $(M_{(2)1} + (K * R[i]))$ mod 256
= (e + (3 * 3)) mod 256
= (101 + 9) mod 256
= 110 (huruf " n " dalam karakter ASCII 256)

M_{32} = $(M_{(3-1)2} + (K * R[i]))$ mod 256
= $(M_{(2)2} + (K * R[i]))$ mod 256
= (g + (3 * 3)) mod 256
= (103 + 9) mod 256
= 112 (huruf " p " dalam karakter ASCII 256)

hasil dari enkripsi baris ke tiga (i = 3) adalah " n p " .

Hasil enkripsi sampai pada tahap baris ke tiga (i=3) dapat dilihat di bawah ini:

\ ^ V g (dalam nilai ASCII 92 94 86 103 → i = 0
_ a Y **j** (dalam nilai ASCII 95 97 89 **106** → i = 1
e g _ (dalam nilai ASCII 101 103 **95** → i = 2
n **p** (dalam nilai ASCII 110 **112** → i = 3

Hasil enkripsi baris ke tiga (i=3) akan digunakan sebagai plainteks pada proses enkripsi baris ke empat, sehingga :

$i = 4 ; j \leq (4 + 1) - 4 \rightarrow j \leq 1$

M_{41} = $(M_{(4-1)1} + (K * R[i]))$ mod 256
= $(M_{(3)1} + (K * R[i]))$ mod 256
= (n + (3 * 4)) mod 256
= (110 + 12) mod 256
= 122 (huruf " z " dalam karakter ASCII 256)

Hasil enkripsi sampai pada tahap baris ke empat (i = 4) dapat dilihat di bawah ini:

\ ^ V g (dalam nilai ASCII 92 94 86 103 → i = 0

_ a Y **j** (dalam nilai ASCII 95 97 89 **106** → i = 1
e g _ (dalam nilai ASCII 101 103 **95** → i = 2
n **p** (dalam nilai ASCII 110 **112** → i = 3
z (dalam nilai ASCII **122** → i = 4

a. Matrik dekripsi segitiga pertama

Pada proses ini yang menjadi *cipher*teks adalah plainteks yang dihasilkan dari proses enkripsi segitiga ke dua (**z p _ j**) kemudian didekrip lagi sesuai dengan formula yang berlaku pada proses dekripsi segitiga ke pertama.

*cipher*teks = **z p _ j** (cipher hasil enkripsi segitiga ke dua)
122 112 95 106 (nilai desimal dalam ASCII)

Untuk baris pertama (i = 1) :

M_{11} = (C[1] - (3 * 1)) mod 256
= (z - (3 * 1)) mod 256
= (122 - 3) mod 256
= 119 (huruf " w " dalam karakter ASCII 256)

M_{12} = (C[2] - (3 * 1)) mod 256
= (p - (3 * 1)) mod 256
= (112 - 3) mod 256
= 109 (huruf " m " dalam karakter ASCII 256)

M_{13} = (C[3] - (3 * 1)) mod 256
= (_ - (3 * 1)) mod 256
= (95 - 3) mod 256
= 92 (huruf " \ " dalam karakter ASCII 256)

M_{14} = (C[4] - (3 * 1)) mod 256
= (j - (3 * 1)) mod 256
= (106 - 3) mod 256
= 103 (huruf " g " dalam karakter ASCII 256)

hasil sandi pada tahap i = 1 (baris pertama) adalah w m \ g.

Sampai pada tahap ini hasil penyandian dapat ditunjukkan di bawah ini :

z p _ j (nilai desimal dalam ASCII : 122 112 95 106) → i = 0
w m \ g (nilai desimal dalam ASCII : **119** 109 92 103) → i = 1

Hasil penyandian baris pertama (i = 1) akan digunakan sebagai *cipher*teks pada proses dekripsi baris ke dua (i = 2), dimana nilai j ≤ i, sehingga :

i = 2, j = 2

M_{22}= ($M_{(2-1)2}$ - 3 * (2)) mod 256
= ($M_{(1)2}$ - 3 * (2)) mod 256
= (m - 6) mod 256
= (109 - 6) mod 256
= 103 (huruf " g " dalam karakter ASCII 256)

M_{23}= ($M_{(2-1)3}$ - 3 * (2)) mod 256
= ($M_{(1)3}$ - 3 * (2)) mod 256
= (\ - 6) mod 256
= (92 - 6) mod 256
= 86 (huruf " V " dalam karakter ASCII 256)

M_{24}= ($M_{(2-1)4}$ - 3 * (2)) mod 256
= ($M_{(1)4}$ - 3 * (2)) mod 256
= (g - 6) mod 256
= (103 - 6) mod 256
= 97 (huruf " a " dalam karakter ASCII 256)

hasil dari dekripsi baris ke dua ini adalah g V a.

Sampai pada tahap ini hasil penyandian dapat ditunjukkan di bawah ini :

z p _ j (nilai desimal dalam ASCII : 122 112 95 106) → i = 0
w m \ g (nilai desimal dalam ASCII : **119** 109 92 103) → i = 1
g V a (nilai desimal dalam ASCII : **103** 86 97) → i = 2

Hasil penyandian baris ke dua (i = 2) akan digunakan sebagai *cipher*teks pada proses dekripsi baris ke tiga (i = 3), dimana nilai j ≤ i, sehingga :

i = 3, j = 3

M_{33}= ($M_{(3-1)3}$ - 3 * (3)) Mod 256
= ($M_{(2)3}$ - 3 * (3)) Mod 256
= (V - 9) mod 256
= (86 - 9) mod 256
= 77 (huruf " M " dalam karakter ASCII 256)

M_{34}= ($M_{(3-1)4}$ - 3 * (3)) Mod 256
= ($M_{(2)4}$ - 3 * (3)) Mod 256
= (a - 9) mod 256
= (97 - 9) mod 256
= 88 (huruf " X " dalam karakter ASCII 256)

hasil dari dekripsi baris ke tiga ini adalah M X.
Sampai pada tahap ini hasil penyandian dapat ditunjukkan di bawah ini :
z p _ j (nilai desimal dalam ASCII : 122 112 95 106) → i = 0
w m \ g (nilai desimal dalam ASCII : **119** 109 92 103) → i = 1
g V a (nilai desimal dalam ASCII : **103** 86 97) → i = 2
M X (nilai desimal dalam ASCII : **77** 88) → i = 3
Hasil penyandian baris ke tiga (i = 3) akan digunakan sebagai *cipher*teks pada proses dekripsi baris ke empat (i = 4), dimana nilai j ≤ i, sehingga :
i = 4, j = 4
M_{44}= ($M_{(4-1)4}$ - 3 * (4)) Mod 256
= ($M_{(3)4}$ - 3 * (4)) Mod 256
= (X - 12) mod 256
= (88 - 12) mod 256
= 76 (huruf " L " dalam karakter ASCII 256)
Hasil dekripsi sampai pada tahap baris ke empat (i = 4) dapat dilihat di bawah ini:
z p _ j (nilai desimal dalam ASCII : 122 112 95 106) → i = 0
w m \ g (nilai desimal dalam ASCII : **119** 109 92 103) → i = 1
g V a (nilai desimal dalam ASCII : **103** 86 97) → i = 2
M X (nilai desimal dalam ASCII : **77** 88) → i = 3
L (nilai desimal dalam ASCII : **76**) → i = 4

b. Matrik dekripsi segitiga ke dua
Pada proses ini yang menjadi *cipher*teks adalah *cipher* yang dihasilkan dari proses dekripsi segitiga pertama (**w g M L**) kemudian didekrip lagi sesuai dengan formula yang berlaku pada proses dekripsi segitiga ke dua.
Plainteks = **w g M L** (cipher hasil enkripsi segitiga pertama)
119 103 77 76 (nilai desimal dalam ASCII)
M_{11} = (C[1] - (3 * 1)) mod 256
= (w - (3 * 1)) mod 256
= (119 - 3) mod 256
= 116 (huruf " t " dalam karakter ASCII 256)
M_{12} = (C[2] - (3 * 1)) mod 256
= (g - (3 * 1)) mod 256
= (103 - 3) mod 256
= 100 (huruf " d " dalam karakter ASCII 256)
M_{13} = (C[3] - (3 * 1)) mod 256
= (M - (3 * 1)) mod 256
= (77 - 3) mod 256
= 74 (huruf " J " dalam karakter ASCII 256)
M_{14} = (C[3] - (3 * 1)) mod 256
= (L - (3 * 1)) mod 256
= (76 - 3) mod 256
= 73 (huruf " I " dalam karakter ASCII 256)
hasil dari dekripsi baris pertama (i = 1) adalah t d J I
Hasil dekripsi sampai pada tahap baris pertama (i = 1) dapat dilihat di bawah ini:
w g M L (dalam nilai ASCII 119 103 77 76 → i = 0
t d J **I** (dalam nilai ASCII 116 100 74 **73** → i = 1
Hasil dekripsi baris pertama (i = 1) akan digunakan sebagai *cipher*teks pada proses dekripsi baris ke dua dimana nilai j ≤ (N + 1) – i, sehingga :
i = 2; j ≤ (4 + 1) – 2 → j ≤ 3
M_{21} = ($M_{(2-1)1}$ - (K * R[i])) mod 256
= ($M_{(1)1}$ - (K * R[i])) mod 256
= (t - (3 * 2)) mod 256
= (116 - 6) mod 256
= 110 (huruf " n " dalam karakter ASCII 256)
M_{22} = ($M_{(2-1)2}$ - (K * R[i])) mod 256
= ($M_{(1)2}$ - (K * R[i])) mod 256
= (d - (3 * 2)) mod 256
= (100 - 6) mod 256
= 94 (huruf " ^ " dalam karakter ASCII 256)
M_{23} = ($M_{(2-1)3}$ - (K * R[i])) mod 256
= ($M_{(1)3}$ - (K * R[i])) mod 256
= (j - (3 * 2)) mod 256
= (74 - 6) mod 256
= 68 (huruf " D " dalam karakter ASCII 256)
hasil dari dekripsi baris ke dua (i = 2) adalah n ^ D
Hasil dekripsi sampai pada tahap baris ke dua (i = 2) dapat dilihat di bawah ini:
w g M L (dalam nilai ASCII 119 103 77 76 → i = 0

t d J **I** (dalam nilai ASCII 116 100 74 **73** → i = 1
n ^ **D** (dalam nilai ASCII 110 94 **68** → i = 2

Hasil dekripsi baris ke dua (i = 2) akan digunakan sebagai *cipher*teks pada proses dekripsi baris ke tiga dimana nilai j ≤ (N + 1) – i, sehingga :

i = 3; j ≤ (4 + 1) – 3 → j ≤ 2

M_{31} = $(M_{(3-1)1}$ - (K * R[i])) mod 256
= $(M_{(2)1}$ - (K * R[i])) mod 256
= (n - (3 * 3)) mod 256
= (110 - 9) mod 256
= 101 (huruf " e " dalam karakter ASCII 256)

M_{32} = $(M_{(3-1)2}$ - (K * R[i])) mod 256
= $(M_{(2)2}$ - (K * R[i])) mod 256
= (^ - (3 * 3)) mod 256
= (94 - 9) mod 256
= 85 (huruf " U " dalam karakter ASCII 256)

hasil dari dekripsi baris ke tiga (i = 3) adalah e U

Hasil dekripsi sampai pada tahap baris ke tiga (i = 3) dapat dilihat di bawah ini:

w g M L (dalam nilai ASCII 119 103 77 76 → i = 0
t d J **I** (dalam nilai ASCII 116 100 74 **73** → i = 1
n ^ **D** (dalam nilai ASCII 110 94 **68** → i = 2
e **U** (dalam nilai ASCII 101 **85** → i = 3

Hasil dekripsi baris ke tiga (i = 3) akan digunakan sebagai *cipher*teks pada proses dekripsi baris ke empat dimana nilai j ≤ (N + 1) – i, sehingga :

i = 4; j ≤ (4 + 1) – 4 → j ≤ 1

M_{41} = $(M_{(4-1)1}$ - (K * R[i])) mod 256
= $(M_{(3)1}$ - (K * R[i])) mod 256
= (e - (3 * 4)) mod 256
= (101 - 12) mod 256
= 89 (huruf " Y " dalam karakter ASCII 256)

Hasil dekripsi sampai pada tahap baris ke empat (i = 4) dapat dilihat di bawah ini :

w g M L (dalam nilai ASCII 119 103 77 76 → i = 0
t d J **I** (dalam nilai ASCII 116 100 74 **73** → i = 1
n ^ **D** (dalam nilai ASCII 110 94 **68** → i = 2
e **U** (dalam nilai ASCII 101 **85** → i = 3
Y (dalam nilai ASCII **89** → i = 4

Setelah melakukan proses dekripsi segitiga yang kedua sudah terlihat hasil dekripsi, yaitu : **Y, U, D, I**

KESIMPULAN

Setelah membahas dan sajian sebelumnya, dapat dibuat kesimpulan sebagai berikut:

1. Prosedur pengamanan teks menggunakan teknik kriptografi adalah dengan memiliki komponen seperti teks atau pesan yang akan dikodekan, kemudian memiliki metode kriptografi yang akan digunakan setelah itu menentukan kunci dari metode kriptografi yang digunakan. Teks yang diperlukan adalah teks atau informasi yang tidak boleh diketahui oleh orang lain, dalam hal ini disebut dengan pesan rahasia. Kemudian kunci yang digunakan harus berjenis angka integer diatas nilai satu.
2. Cara untuk mengimplementasikan algoritma triangle chain chipper adalah dengan melakukan pengujian manual menggunakan data yang sederhana. Dalam kasus ini penulis menggunakan nama penulis sendiri sebagai data untuk proses enkripsi tersebut. Pengujian dilakukan menggunakan data Ascii 255. Dalam kasus ini data yang digunakan adalah "YUDI" dengan hasil chipper adalah "zp_j" dengan menggunakan kunci "3". Setelah melihat hasil perhitungan manual maka dapat dilakukan pembuatak perhitungan otomatis menggunakan aplikasi Visual Basic 2008.
3. Cara untuk merancang dan membuat aplikasi enkripsi pesan teks menggunakan algoritma triangle chain chipper adalah dengan menerapkan logika dan algoritma triangle chain chipper menggunakan IDE Visual Basic 2008. Semua variabel dan perhitungan disesuaikan dengan yang ada dalam perhitungan manual. Kemudian, hasil perhitungan menggunakan aplikasi akan disesuaikan lagi dengan perhitungan manual yang telah dilakukan

DAFTAR PUSTAKA

[1] D. Ariyus, "Kriptografi keamanan data dan komunikasi," *Yogyakarta Graha Ilmu*, 2006.
[2] M. D. Herminingtyas and R. Sholihah, "Implementasi algoritma rc4 untuk enkripsi keamanan data," pp. 1–11, 1987.
[3] I. K. Sudarsana *et al.*, "Paradigma Pedidikan Bermutu Berbasis Teknologi Pendidikan," *Jayapangus Press Books*, vol. 0, no. 0, Mar. 2018, Accessed: Sep. 12, 2018. [Online]. Available: http://books.jayapanguspress.org/index.php/publisher/article/view/19.
[4] R. Holmes, "6 Prediksi Tren Media Sosial yang Akan Mengubah Bisnis pada 2019," 2019. https://id.techinasia.com/prediksi-tren-media-sosial-2019 (accessed Sep. 17, 2019).
[5] T. Zebua, "ANALISA DAN IMPELEMENTASI ALGORITMA TRIANGLE CHAIN PADA PENYANDIAN RECORD DATABASE," *Pelita Inform.*, vol. III, pp. 37–49, Apr. 2013.
[6] T. Limbong, "Pengujian Kriptografi Klasik Caesar Chipper Menggunakan Matlab," *no. Sept.*, vol. 2017, 2015.

[7] H. Sahara, "Implementasi Pengamanan Pesan Chatting menggunakan Metode Vigenere Cipher dan Cipher Block Chainning," *MEANS (Media Inf. Anal. dan Sist.*, vol. 3, no. 2, pp. 173–178, 2018.

[8] B. Silaban and T. Limbong, "Aplikasi Pembelajaran Pengenalan Kriptografi Algoritma Affine Cipher Dan Vigenere Cipher Menggunakan Metode Computer Assisted Instruction," *Media Inf. Anal. dan Sist.*, vol. 2, no. 2, pp. 14–20, 2017.

[9] A. P. Windarto and A. Sudrajat, "Enhancing Database Transmission Security by Implementing Triangle Chain Cipher Algorithm," *Int. Conf. Res. Soc. Sci. Humanit.*, 2019.

[10] M. Simanjuntak, T. Pasaribu, and S. Rahmadilla, "Implementasi Algoritma Merkle Hellman untuk Keamanan Database," *MEANS (Media Inf. Anal. dan Sist.*, vol. 4, no. 1, pp. 46–50, 2019, [Online]. Available: http://ejournal.ust.ac.id/index.php/Jurnal_Means/.

[11] F. Anita, "Implementasi Algoritma Modular Multiplication Based Block Cipher Dalam Mengamankan Data Teks," *MEANS (Media Inf. Anal. dan Sist.*, vol. 3, no. 2, pp. 121–125, 2018.