IJCIT (Indonesian Journal on Computer and Information Technology)
Vol.2 No.1, Mei 2017, pp. 77~82
ISSN: 2527-449X
E-ISSN**:** 2549-7421

Pemanfaatan Kunjungan Pohon Biner Pada Kriptografi Hill Cipher Kunci Matriks Persegi Panjang

Tuti Alawiyah

AMIK BSI Tasikmalaya
tuti.tah@bsi.ac.id

Abstrak

Keamanan informasi dalam proses pertukaran data sangatlah penting. Untuk itu diperlukan kriptografi yang dapat menjaga kerahasiaan informasi yang ditukarkan. Dua proses penting pada kriptografi yaitu proses enkripsi dan proses deskripsi. Pada proses enkripsi dilakukan perubahan informasi/pesan/data asli (plainteks) menjadi bentuk yang tidak dimengerti / teks sandi (cipherteks), sedangkan deskripsi merubah informasi/pesan yang tidak dimengerti / teks sandi menjadi data aslinya yang dapat dimengerti. Hill cipher merupakan salah satu algoritma kriptografi yang memanfaatkan operasi modulo dengan menggunakan matriks sebagai kunci untuk merubah data asli menjadi teks sandi. Pada penelitian ini matriks kunci yang digunakan adalah matriks persegi panjang dengan menambahkan pemanfaatan kunjungan pohon biner di awal prosesnya. Setiap karakter data asli dikonversikan kedalam bilangan desimal sesuai dengan kode ASCII dikurangi 32 dengan modulo 95. Penggunaan matriks kunci persegi panjang menjadikan teks sandi lebih panjang dibandingkan data aslinya sehingga informasi/pesan yang disampaikan menjadi lebih tersamarkan, ditambahkan dengan operasi kunjungan pohon biner menjadikan teks sandi yang terbentuk menjadi lebih rumit untuk dipecahkan oleh kripnatalis.
Kata Kunci: hill cipher, kriptografi, kunjungan pohon biner, matriks persegi panjang, modulo

Abstract

Information security in the process of data exchange is very important. For that we need cryptography that can keep the confidentiality of the information exchanged. Two important processes in cryptography are the encryption process and the description process. In the encryption process the information / message (plaintext) changes into an unintelligible form (ciphertext), whereas the description changes the unintelligible information / message (ciphertext) to its original plain text. Hill cipher is one of the cryptographic algorithms that utilize modulo operation using matrix as key to convert plaintext into ciphertext. In this study the key matrix used is a rectangular matrix by adding the utilization of traverse binary tree at the beginning of the process. Each plaintext character is converted into a decimal number according to the ASCII code minus 32 by modulo 95. The use of a rectangular key matrix makes ciphertext longer than its plaintext so that the information / messages conveyed becomes more disguised, added by the binary visitation operation making the ciphertext more complicated to be solved by crypnatalis.
Keywords: ***hill cipher, cryptografy, traverse binary tree, rectangular matrix, modulo***

1. Pendahuluan

Kriptografi diperlukan untuk memastikan keamanan informasi yang dikirim. Hal ini diperlukan untuk kenyamanan bagi pengirim maupun penerima informasi. Salah satu keamanan informasi yang dimaksud adalah kerahasiaan. Pengirim dan penerima informasi harus yakin bahwa informasi yang dikirim atau diterima tidak diketahui atau tidak terbaca oleh pihak lain. “Prinsip kerckhoff mengatakan bahwa keamanan sebuah sistem kripto tidak bersandar pada kerahasiaan algoritma tetapi hanya pada kerahasiaan kunci” (Muis, 2013).

Berbagai teknik digunakan dalam kriptografi untuk menghasilkan sistem keamanan yang handal, salah satunya kriptografi hill cipher. Meskipun termasuk kedalam kategori kriptografi klasik, namun perkembangan teknik dalam kriptogtafi hill cipher mampu menghasilkan teks sandi yang rumit, sehingga sulit dipahami oleh pihak lain.

Diterima Maret 20, 2017; Revisi April 20, 2017; Disetujui April 25, 2017

Kriptografi hill cipher menggunakan operasi matriks dan modulo. Pada perkembangannya, penggunaan kunci matriks semakin beragam serta adanya beberapa penambahan proses yang menghasilkan teks sandi yang semakin rumit dan sulit untuk menemukan hubungan linier antara teks sandi dengan data aslinya.

Beberapa penelitian yang berkaitan dengan kriptografi hill cipher diantaranya menghasilkan proses deskripsi yang tidak menggunakan invers matriks karena proses enkripsi dan deskripsi menggunakan kunci matriks yang sama. Pada penelitian ini juga digunakan kunci matriks acak pada proses enkripsi dan deskripsinya (Rahman, Abidin, Yusof, & Usop, 2013)

Penelitian juga dilakukan oleh Alz Danny Wowor dimana pembangkit kunci menggunakan kunci tambahan determinan dari matriks polinomial berupa persamaan polinomial. Selain itu juga menggunakan *convert between base* (CBB) untuk mengkonversi *teks sandi* menjadi bilangan biner (Wowor, 2014).

Penelitii lain menggunakan 2 jenis kunci yaitu kunci umum (*public key)* dan kunci rahasia (*private key*). Kunci umum yang digunakan adalah matriks persegi yang memiliki nilai eigen irrational. Sedangkan kunci rahasianya terdiri dari sebuah bilangan bulat dan matriks persegi panjang (Viswanat & Kumar, 2015)

Pada penelitian yang lain Kombinasi 3 metode digunakan dalam proses enkripsi, dimana *data asli* diproses menggunakan metode Caesar cipher. Hasilnya diproses lagi menggunakan metode vernam cipher dan terakhir diproses menggunakan hill cipher (Puspita & Wayahdi, 2015)

Dalam penelitian ini, matriks kunci yang digunakan adalah matriks persegi panjang yang akan dikombinasikan dengan operasi kunjungan pohon biner. Setiap karakter pada data asli akan dikonversikan kedalam bilangan desimal ASCII dikurangi 32 dengan penggunaan operasi modulo 95. Dengan kombinasi ini diharapkan teks sandi yang dihasilkan akan semakin rumit dan semakin sulit untuk dipecahkan oleh kripnatalis.

Kunci matriks yang digunakan harus memiliki invers yang akan digunakan pada proses deskripsi. sebuah matriks persegi panjang dapat digunakan sebagai matriks kunci jika memiliki matriks *pseudo-invers*nya yang mana sebuah matriks B dikatakan *pseudo-invers* dari matriks A jika memenuhi syarat-syarat sebagai berikut:

1. $ABA = A$
2. $BAB = B$
3. $(AB)^H = AB$
4. $(BA)^H = BA$

Pseudo Invers matriks $A_{(mxn)}$ yang dinotasikan dengan $A^{\#}$ didapatkan dengan ketentuan:

1. Matriks $A_{(mxn)}$ dengan $m \geq n$ merupakan matriks yang memiliki *full coloumn rank* (rank matriks = kolom matriks), maka $A^{\#} = (A^H A)^{-1} A^H$
2. Matriks $A_{(mxn)}$ dengan $m < n$ merupakan matriks yang memiliki *full row rank* (rank matriks = baris matriks), maka $A^{\#} = A^H (A A^H)^{-1}$

Modulo merupakan operasi bilangan bulat yang menghasilkan sisa bagi bulat. Sebuah bilangan bulat x dibagi bilangan bulat y ($y > 0$) menghasilkan sisa bilangan bulat m, maka x modulo p menghasilkan m.

$x \bmod p = m$ sedemikian hingga:

$$x = p * q + m \quad \text{dengan } 0 \leq m < p$$

"Pohon biner adalah Pohon M-ary dimana M=2, yang artinya simpul paling banyak memiliki 2 simpul sub ordinat yang biasa disebut sub ordinat kiri (left-child) dan sub ordinat kanan (right-child)" (Sjukani, 2012)

Ada beberapa cara mengunjungi pohon biner (operasi pohon biner), diantaranya preorder, inorder dan postorder. Pada preoder kunjungan dimulai dari akar, subordinat kiri lalu subordinat kanan. Sedangkan inorder, kunjungan dimulai dari subordinat kiri, akar lalu subordinat kanan. Pada operasi kunjungan pohon biner postorder, kunjungan dimulai dari subordinat kiri, subordinat kanan lalu akar.

2. Metode Penelitian

A. Analisis Kriptografi Hill Cipher

Algoritma kriptografi hill cipher terdiri dari proses inisialisasi matriks kunci, enkripsi dan deskripsi dengan operasi modulo *p* pada setiap proses perhitungannya. Inisialisasi matriks kunci dilakukan untuk memeriksa apakah matriks kunci yang dipilih memiliki invers atau tidak? Karena hanya matriks yang memiliki *invers* yang dapat digunakan sebagai matriks kunci. pada penelitian ini matriks kunci yang digunakan adalah matriks persegi panjang yang memiliki *pseudo invers*. Penggunaan matriks persegi panjang menjadikan teks sandi lebih panjang dibandingkan data aslinya,

sehingga informasi menjadi lebih tersamarkan.

Proses enkripsi dilakukan melalui beberapa tahap dengan menambahkan operasi kunjungan pohon biner, yaitu:

1. Hitung panjang karakter data asli *l*, jika *l* mod 3 ≠ 0, maka tambahkan karakter bebas hingga didapatkan *l* mod *3* = 0
2. Partisi data asli kedalam blok-blok P_1, P_2, …, P_n dengan masing-masing blok terdiri dari 3 elemen
3. Setiap blok dibuatkan pohon biner dengan elemen pertama sebagai root, elemen kedua sebagai subordinat kiri dan elemen ketiga sebagai subordinat kanan.
4. Cetak simpul pohon biner menggunakan operasi kunjungan pohon inorder atau postorder
5. Gabungkan hasil proses 3
6. Hitung panjang elemen hasil proses 5 *l,* jika *l* mod *r* ≠ 0, maka tambahkan karakter bebas hingga didapatkan *l* mod *r* = 0. Dimana *r* adalah rank matriks kunci.
7. Konversi elemen hasil proses 6 kedalam bilangan desimal sesuai dengan kode ASCII dikurangi 32.
8. Partisi data asli kedalam blok-blok $P_1, P_2, \ldots, P_i$ dimana $i = 1 \ldots \frac{(l)}{r}$ dengan masing-masing blok terdiri dari *r* elemen.
9. Hitung $C_1, C_2, \ldots, C_i$ dengan ketentuan:
   a. $C_i = (A (P_i)^T)^T$ jika $m \geq n$, matriks kunci *full coloumn rank*
   b. $C_i = P_i \; A$ jika $m < n$, matriks kunci *full row rank*
10. Gabungkan $C_1, C_2, \ldots, C_i$
11. Tambahkan 32 semua bilangan pada $C_1, C_2, \ldots, C_i$ lalu konversi ke dalam bentuk karakter ASCII sehingga didapatkan sebuah *teks sandi*.

Proses deskripsi menggunakan *pseudo invers* matriks kunci. Proses ini terdiri dari beberapa tahap, yaitu:

1. Konversi teks sandi kedalam bilangan desimal sesuai karakter ASCII dikurangi 32.
2. Partisi bilangan desimal *teks sandi* kedalam blok-blok $C_1, C_2, \ldots, C_i$ dengan $i = 1 \ldots \frac{l}{j}$ dimana setiap blok terdiri dari *j* elemen.
   a. *j* = *m*, jika matriks kunci *full coloumn rank* ($m \geq n$)
   b. *j* = *n,* jika matriks kunci *full row rank* ($m < n$)
3. Hitung $P_1, P_2, \ldots, P_i$ dengan ketentuan:
   a. $P_i = (A^{\#} (C_i)^T)^T$ jika $m \geq n$, matriks kunci *full coloumn rank*
   b. $P_i = C_i \; A^{\#}$ jika $m < n$, matriks kunci *full row rank*
4. Tambahkan dengan angka 32 setiap elemen $P_1, P_2, \ldots, P_i$ dan gabungkan.
5. Konversi *P* kedalam karakter ASCII
6. Hitung elemen hasil proses 5 (*l*), jika *l* mod 3 ≠ 0, maka hapus elemen akhir hingga didapatkan *l* mod *3* = 0
7. Partisi hasil proses 6 kedalam blok-blok $P_1, P_2, \ldots, P_i$ dimana setiap blok terdiri dari 3 elemen
8. Setiap blok dibuatkan pohon biner sesuai dengan kunjungan pohon biner yang dipilih pada saat proses enkripsi.
   a. Jika inorder, maka elemen kedua menjadi root, elemen pertama sebagai subordinat kiri dan elemen ketiga sebagai subordinat kanan
   b. Jika postorder, maka elemen ketiga sebagai root, elemen pertama sebagai subordinat kiri dan elemen kedua sebagai subordinat kanan.
9. Cetak simpul pohon biner menggunakan operasi kunjungan pohon biner preorder
10. Gabungkan hasil proses 9 untuk mendapatkan data asli.

3. Hasil dan Pembahasan

A. Inisialisasi Matriks Kunci

Tentukan sebuah matriks kunci $A_{(4x5)}$ terdiri dari 4 baris, 5 kolom

$$A = \begin{bmatrix} 4 & 7 & 1 & 9 & 0 \\ 8 & 2 & 6 & 5 & 3 \\ 7 & 3 & 7 & 8 & 1 \\ 5 & 2 & 8 & 1 & 3 \end{bmatrix}$$

jumlah baris < jumlah kolom
Rank (*A*)=4
Matriks *A* *full row rank* karena rank (*A*) sama dengan jumlah baris yaitu 4, maka *pseudo invers* matriks A adalah $\boldsymbol{A^{\#} = A^T (A A^T)^{-1}}$

$$A^T = \begin{bmatrix} 4 & 8 & 7 & 5 \\ 7 & 2 & 3 & 2 \\ 1 & 6 & 7 & 8 \\ 9 & 5 & 8 & 1 \\ 0 & 3 & 1 & 3 \end{bmatrix}$$

$$A\,A^{\mathrm{T}} = \begin{bmatrix} 52 & 2 & 33 & 51 \\ 2 & 43 & 52 & 11 \\ 33 & 52 & 77 & 13 \\ 51 & 11 & 13 & 8 \end{bmatrix}$$

$$A^{\#} = \begin{bmatrix} 69 & 61 & 2 & 8 \\ 23 & 24 & 59 & 81 \\ 27 & 60 & 90 & 87 \\ 33 & 32 & 91 & 61 \\ 40 & 60 & 67 & 29 \end{bmatrix}$$

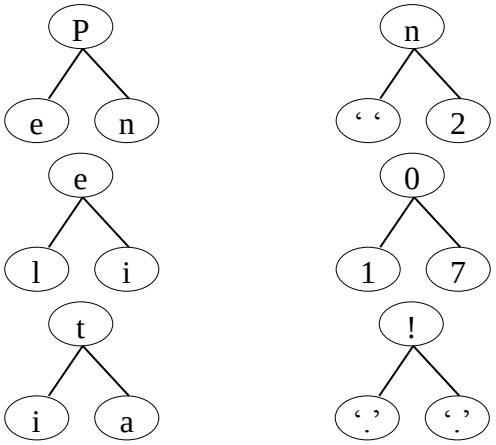
Syarat-syarat *p-invers*:
1. $A\,A^{\#}A = A$ dipenuhi
2. $A^{\#}A\,A^{\#} = A^{\#}$ dipenuhi
3. 3. $(A\,A^{\#})^* = A\,A^{\#}$ dipenuhi
4. $(A^{\#}A)^* = A^{\#}A$ dipenuhi

Dari hasil perhitungan menunjukkan matriks *A* memenuhi syarat untuk digunakan sebagai matriks kunci.

B. Enkripsi

Proses enkripsi menggunakan matriks *A* sebagai matriks kunci, serta pemilihan kunjungan pohon biner yang akan digunakan (inorder atau postorder).

Data asli: Penelitian 2017!
1. Panjang karakter data asli *l* = 16, karena *l* mod 3 ≠ 0, maka perlu ditambahkan 2 karakter, misalkan karakter titik "."
2. Partisi data asli sehingga didapatkan:
   P_1 = [P,e,n] P_4 = [n, spasi, 2]
   P_2 = [e,l,i] P_5 = [0, 1, 7]
   P_3 = [t,i,a] P_6 = [!, titik, titik]
3. Pembuatan pohon biner untuk setiap blok



4. Cetak simpul setiap pohon biner menggunakan kunjungan pohon biner postorder sehingga didapatkan:
   P_1 = [e,n, P] P_4 = [spasi, 2, n]
   P_2 = [l,i, e] P_5 = [1, 7, 0]
   P_3 = [i,a, t] P_6 = [titik, titik, !]
5. Gabungkan hasil proses 4 sehingga didapakan P = [e,n,P,l,i,e,i,a,t,spasi,2,n, 1,7,0, titik, titik, !]
6. Panjang elemen hasil proses 5 *l* = 18, karena *l* mod *r* ≠ 0, maka perlu ditambahkan 2 karakter bebas hingga didapatkan *l* mod *r* = 0. Dimana *r* adalah rank matriks kunci (4) hingga didapatkan P = [e,n,P,l,i,e,i,a,t,spasi,2,n, 1,7,0, titik, titik, !, koma, koma]
7. Konversi elemen hasil proses 6 kedalam bilangan desimal sesuai dengan kode ASCII dikurangi 32.
   P=[69, 78, 48, 76, 73, 69, 73, 65, 84, 0, 18, 78, 17, 23, 16, 14, 14, 1, 12,12]
8. Partisi data asli kedalam blok-blok $P_1, P_2, \ldots, P_i$ dimana *i* = 1 ... $\frac{(l)}{r}$ dengan masing-masing blok terdiri dari *r* elemen.
   P_1 = [69,78,48,76] P_4 = [17,23,16,14]
   P_2 = [73,69,73,65] P_5 = [14,1,12,12]
   P_3 = [84, 0,18, 78]
9. Karena matriks kunci merupakan matriks full row rank, maka $C_1, C_2, \ldots, C_i$ didapat menggunakan persamaan $C_i = P_i\ A$

$$C_1 = P_1\ A\ = \begin{bmatrix} 69 & 78 & 48 & 76 \end{bmatrix} * \begin{bmatrix} 4 & 7 & 1 & 9 & 0 \\ 8 & 2 & 6 & 5 & 3 \\ 7 & 3 & 7 & 8 & 1 \\ 5 & 2 & 8 & 1 & 3 \end{bmatrix} mod\ 95$$

$$= \begin{bmatrix} 1 & 80 & 56 & 46 & 35 \end{bmatrix}$$

$$C_2 = P_2\ A\ = \begin{bmatrix} 73 & 69 & 73 & 65 \end{bmatrix} * \begin{bmatrix} 4 & 7 & 1 & 9 & 0 \\ 8 & 2 & 6 & 5 & 3 \\ 7 & 3 & 7 & 8 & 1 \\ 5 & 2 & 8 & 1 & 3 \end{bmatrix} mod\ 95$$

$$= \begin{bmatrix} 65 & 48 & 93 & 36 & 0 \end{bmatrix}$$

$$C_3 = P_3\ A\ = \begin{bmatrix} 84 & 0 & 18 & 78 \end{bmatrix} * \begin{bmatrix} 4 & 7 & 1 & 9 & 0 \\ 8 & 2 & 6 & 5 & 3 \\ 7 & 3 & 7 & 8 & 1 \\ 5 & 2 & 8 & 1 & 3 \end{bmatrix} mod\ 95$$

$$= \begin{bmatrix} 92 & 38 & 74 & 28 & 62 \end{bmatrix}$$

$$C_4 = P_4\ A\ = \begin{bmatrix} 17 & 23 & 16 & 14 \end{bmatrix} * \begin{bmatrix} 4 & 7 & 1 & 9 & 0 \\ 8 & 2 & 6 & 5 & 3 \\ 7 & 3 & 7 & 8 & 1 \\ 5 & 2 & 8 & 1 & 3 \end{bmatrix} mod\ 95$$

$$= \begin{bmatrix} 54 & 51 & 94 & 30 & 32 \end{bmatrix}$$

$C_5 = P_5\ A\ =$

$$[14 \quad 1 \quad 12 \quad 12] * \begin{bmatrix} 4 & 7 & 1 & 9 & 0 \\ 8 & 2 & 6 & 5 & 3 \\ 7 & 3 & 7 & 8 & 1 \\ 5 & 2 & 8 & 1 & 3 \end{bmatrix} mod\ 95$$

$= [18 \quad 65 \quad 10 \quad 49 \quad 51]$

10. Gabungkan $C_1, C_2, \ldots, C_i$
    C = [1,80,56,46,35,65,48,93,36,0,92,38, 74,28,62,54,51,94,30,32,18,65,10,49,51]

11. Tambahkan 32 semua bilangan lalu konversi ke dalam bentuk karakter ASCII sehingga didapatkan sebuah *teks sandi*.
    C = [33,112,88,78,67,97,80,125,68,32, 124,70,106,60,94,86,83,126,62,64,50,97, 42,81,83]

    Konversikan C kedalam karakter ASCII sehingga didapat teks sandi sebagai berikut:
    !pXNCaP}D |Fj<^VS~>@2a*QS

C. Deskripsi

Matriks kuci merupakan matriks full row rank, maka proses deskripsi menggunakan persamaan $P_i = C_i\ A$.Proses deskripsi dilakukan menggunakan algoritma berikut ini:

1. Konversi *teks sandi* **!pXNCaP}D |Fj<^VS~>@2a*QS** kedalam bilangan desimal sesuai karakter ASCII dikurangi 32 sehingga didapatkan
   C = [1,80,56,46,35,65,48,93,36,0,92,38, 74,28,62,54,51,94,30,32,18,65,10,49,51]

2. Partisi bilangan desimal *teks sandi* kedalam blok-blok $C_1, C_2, \ldots, C_i$ dengan $i = 1 \ldots \frac{l}{j}$ dimana setiap blok terdiri dari j elemen. Karena matriks kunci merupakan matriks full row rank, maka *j adalah jumlah kolom matriks kunci yaitu 5 sehingga didapatkan:*
   C_1 = [1,80,56,46,35]
   C_2 = [65,48,93,36,0]
   C_3 = [92,38,74,28,62]
   C_4 = [54,51,94,30,32]
   C_5 = [18,65,10,49,51]

3. Hitung $P_1, P_2, \ldots, P_i$ menggunakan persamaan $P_i = C_i\ A^{\#}$

   $P_1 = C_1\ A^{\#}\ =$

$$[1 \quad 80 \quad 56 \quad 46 \quad 35] * \begin{bmatrix} 69 & 61 & 2 & 8 \\ 23 & 24 & 59 & 81 \\ 27 & 60 & 90 & 87 \\ 33 & 32 & 91 & 61 \\ 40 & 60 & 67 & 29 \end{bmatrix} mod\ 95$$

$= [69 \quad 78 \quad 48 \quad 76]$

$P_2 = C_2\ A^{\#}\ =$

$$[65 \quad 48 \quad 93 \quad 36 \quad 0] * \begin{bmatrix} 69 & 61 & 2 & 8 \\ 23 & 24 & 59 & 81 \\ 27 & 60 & 90 & 87 \\ 33 & 32 & 91 & 61 \\ 40 & 60 & 67 & 29 \end{bmatrix} mod\ 95$$

$= [73 \quad 69 \quad 73 \quad 65]$

$P_3 = C_3\ A^{\#}\ =$

$$[92 \quad 38 \quad 74 \quad 28 \quad 62] * \begin{bmatrix} 69 & 61 & 2 & 8 \\ 23 & 24 & 59 & 81 \\ 27 & 60 & 90 & 87 \\ 33 & 32 & 91 & 61 \\ 40 & 60 & 67 & 29 \end{bmatrix} mod\ 95$$

$= [84 \quad 0 \quad 18 \quad 78]$

$P_4 = C_4\ A^{\#}\ =$

$$[54 \quad 51 \quad 94 \quad 30 \quad 32] * \begin{bmatrix} 69 & 61 & 2 & 8 \\ 23 & 24 & 59 & 81 \\ 27 & 60 & 90 & 87 \\ 33 & 32 & 91 & 61 \\ 40 & 60 & 67 & 29 \end{bmatrix} mod\ 95$$

$= [17 \quad 23 \quad 16 \quad 14]$

$P_5 = C_5\ A^{\#}\ =$

$$[18 \quad 65 \quad 10 \quad 49 \quad 51] * \begin{bmatrix} 69 & 61 & 2 & 8 \\ 23 & 24 & 59 & 81 \\ 27 & 60 & 90 & 87 \\ 33 & 32 & 91 & 61 \\ 40 & 60 & 67 & 29 \end{bmatrix} mod\ 95$$

$= [14 \quad 1 \quad 12 \quad 12]$

4. Tambahkan dengan angka 32 setiap elemen $P_1, P_2, \ldots, P_i$ dan gabungkan sehingga didapat

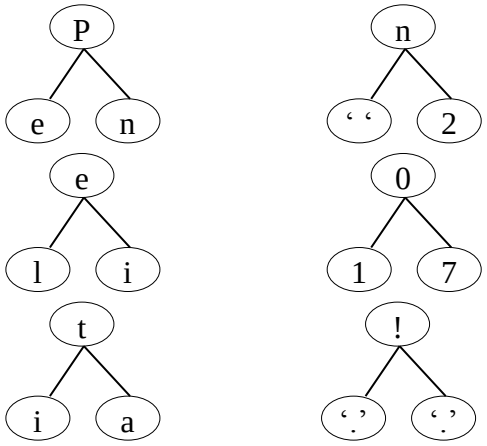
   P = [101, 110, 80, 108, 105, 101, 105, 97, 116, 32, 50, 110, 49, 55, 48, 46, 46, 33, 44,44]

5. Konversi P kedalam karakter ASCII sehingga didapat

P = [e,n,P,l,i,e,i,a,t,spasi,2,n,1,7,0,titik, titik,!,koma,koma]

6. Hitung elemen hasil proses 5 (*l*), jika *l* mod 3 ≠ 0, maka hapus elemen akhir hingga didapatkan *l* mod *3* = 0
   l(P) = 20, maka hapus 2 karakter akhir hingga l(P) = 18
7. Partisi hasil proses 6 kedalam blok-blok P_1, P_2, ... ,P_i dimana setiap blok terdiri dari 3 elemen sehingga didapat
   P_1 = [e,n, P] P_4 = [spasi, 2, n]
   P_2 = [l,i, e] P_5 = [1, 7, 0]
   P_3 = [i,a, t] P_6 = [titik, titik, !]

8. Setiap blok dibuatkan pohon biner sesuai dengan kunjungan pohon biner yang dipilih pada saat proses enkripsi yaitu postorder, maka elemen ketiga sebagai root, elemen pertama sebagai subordinat kiri dan elemen kedua sebagai subordinat kanan.



9. Cetak simpul pohon biner menggunakan operasi kunjungan pohon biner preorder sehingga didapatkan:
   P_1 = [P,e,n] P_4 = [n,spasi, 2]
   P_2 = [e,l,i] P_5 = [0,1, 7]
   P_3 = [t,i,a] P_6 = [!,titik, titik]

10. Gabungkan hasil proses 9 untuk mendapatkan data asli.
    P = Penelitian 2017!

4. Kesimpulan

Operasi kunjungan pohon biner yang dimanfaatkan pada kriptografi hill cipher kunci matriks persegi panjang menghasilkan teks sandi yang cukup rumit. Hal ini akan mempersulit penyerang dalam menemukan kunci matriks yang digunakan karena sulitnya menemukan persamaan liniernya, sehingga keamanan data lebih terjamin kerahasiaannya. Dengan terjaminnya kerahasiaan data menjadikan algoritma ini dapat disisipkan pada aplikasi sebagai pengaman data.

Referensi

Muis, S. (2013). *Pengantar Kriptografik Kuantum Teknik Enkripsi Masa Depan.* Yogyakarta: Graha Ilmu.

Puspita, K., & Wayahdi, M. R. (2015). Analisis Kombinasi Metode Caesar Cipher, Vernam Cipher, dan Hill Cipher Dalam Proses Kriptografi. *Seminar Nasional Teknologi Informasi dan Multimedia 2015* (pp. 1-6). Yogyakarta: STMIK AMIKOM Yogyakarta.

Rahman, M. N., Abidin, A. F., Yusof, M. K., & Usop, N. S. (2013). Cryptography: A New Approach of Classical Hill Cipher. *International Journal of Security and Its Applications*, 179-190.

Sjukani, M. (2012). *Struktur Data (Algoritma & struktur Data 2) dengan C, C++.* Jakarta: Mitra Wacana Media.

Viswanat, M., & Kumar, R. M. (2015). A Public Key Cryptosystem Using Hiil's Cipher. *Journal of Discrete Mathematical Sciences and Cryptography*, 129-138.

Wowor, A. D. (2014). Penggunaan Determinan Polinomial Matriks Dalam Modifikasi Kriptografi Hill Cipher.