PRISMA 4 (2021): 553-559
PRISMA, Prosiding Seminar Nasional Matematika
https://journal.unnes.ac.id/sju/index.php/prisma/
ISSN 2613-9189

Pengkodean *Monoalphabetic* Menggunakan *Affine Cipher* dengan Kunci *Diffie-Hellman*

Tika Khairani[a,*], Kiswara Agung Santoso[a], Ahmad Kamsyakawuni[a]

[a] *Universitas Jember, Jl. Kalimantan 37, Jember 68121, Indonesia*

* *Alamat Surel: tikakhairaniii@gmail.com*

Abstrak

Kriptografi adalah ilmu yang digunakan untuk menjaga kerahasiaan pesan agar tetap aman dan tidak dapat diketahui oleh pihak yang tidak berkepentingan. Algoritma dalam kriptografi cukup banyak, diantaranya *Affine Cipher* dan *Diffie-Hellman*. *Affine Cipher* merupakan perluasan dari *Caesar Cipher* dan termasuk *monoalphabetic substitution cipher* yang setiap karakternya dapat dikonversi menjadi bilangan desimal. *Diffie-Hellman* merupakan algoritma kunci publik pertama yang memungkinkan dua pengguna saling bertukar kunci. Pada artikel ini akan dibahas mengenai penggunaan *Affine Cipher* dengan kunci *Diffie-Hellman* untuk meningkatkan keamanan pada pengkodean pesan *text*, serta hasil perbandingannya dengan *Affine Cipher* dengan kunci biasa. Rumus *Affine Cipher* mengalami modifikasi dengan adanya pengurangan dan penambahan 32, agar karakter yang muncul merupakan ASCII *printable characters*. Nilai modulo yang digunakan juga mengalami perubahan dari 26 menjadi 95. Metode yang digunakan yaitu proses pertukaran dan pembangkitan kunci, proses enkripsi dan proses dekripsi. Hasil dari penelitian ini menunjukkan bahwa *Affine Cipher* dengan kunci *Diffie-Hellman* lebih aman dibandingkan dengan *Affine Cipher* dengan kunci biasa, karena kunci penggeser bersifat rahasia yang hanya diketahui oleh pihak-pihak yang bertukar pesan.

Kata kunci:

Kriptografi, *Monoalphabetic*, *Affine Cipher*, *Diffie-Hellman*.

1. Pendahuluan

Kriptografi merupakan ilmu yang mempelajari penyandian pesan. Seiring berkembangnya zaman keamanan penyandian pesan menjadi sangat sensitif, karena banyak individu yang menyalahgunakan informasi untuk hal yang merugikan orang lain. Sehingga dibutuhkan ilmu penyandian pesan yang baik agar kerahasiaan pesan terjaga dengan aman.

Salah satu algoritma yang dapat digunakan untuk menyandikan pesan adalah algoritma substitusi. Metode substitusi terdiri *polyalphabetic substitution* dan *monoalphabetic substitution*. *Polyalphabetic substitution* adalah teknik kriptografi substitusi yang mengganti setiap karakter pada *plaintext* menjadi karakter lain pada *ciphertext*, dan karakter yang sama pada *plaintext* diganti dengan karakter yang berbeda pada *ciphertext* sesuai dengan kunci yang digunakan. Contoh *polyalphabetic substitution* yang dikenal adalah *Vigenere Cipher*. M*onoalphabetic substitution* adalah teknik kriptografi substitusi yang mengganti setiap karakter pada *plaintext* menjadi karakter lain pada *ciphertext*, dan karakter yang sama pada *plaintext* diganti dengan karakter yang sama pula pada *ciphertext*. Contoh *monoalphabetic substitution* yang dikenal adalah *Affine Cipher* (Permanasari dan Harahap, 2018). *Affine Cipher* adalah algoritma yang memiliki kunci dua bilangan bulat *R* dan *G*. Bilangan *R* yang digunakan merupakan bilangan bulat yang harus relatif prima dengan jumlah alfabet yaitu 26, dengan kata lain GCD(*R*,26)=1 (Sadikin, 2012). Rumus *Affine Cipher* pada proses enkripsi dan dekripsi dapat dilihat pada Persamaan (1) dan Persamaan (2).

$$C_i = \left((R \times P_i) + G\right) mod\ 26 \quad (1)$$

To cite this article:
Khairani, T., Santoso, K. A., & Kamsyakawuni, A. (2021). Pengkodean *Monoalphabetic* Menggunakan *Affine Cipher* dengan Kunci *Diffie-Hellman*. *PRISMA, Prosiding Seminar Nasional Matematika 4*, 553-559

$$P_i = \left(R^{-1}(C_i - G)\right) mod\ 26 \quad (2)$$

Berdasarkan penjelasan diatas dapat diketahui bahwa *monoalphabetic substitution* akan lebih mudah dipecahkan dibandingkan dengan *polyalphabetic substitution*. Oleh karena itu dibutuhkan algoritma tambahan untuk keamanan penyandian pesan dengan *monoalphabetic substitution* agar pesan tidak mudah dipecahkan. Algoritma yang dapat ditambahkan untuk membantu mengamankan pesan dengan *monoalphabetic substitution* adalah *Diffie-Hellman*.

Diffie-Hellman merupakan sebuah algoritma yang memungkinkan kedua pihak saling bertukar kunci melalui jaringan publik. Hasil dari pertukaran kunci yang dilakukan adalah sebuah kunci privat yang sama pada kedua pihak. Kunci privat tersebut secara matematis sulit dipecahkan oleh pihak lain yang tidak bertukar kunci (Ahirwal dan Ahke, 2013).

Affine Cipher (*monoalphabetic substitution*) dan *Diffie-Hellman* pernah digunakan oleh beberapa peneliti, yaitu Siregar (2019) melakukan penelitian dengan menerapkan *Affine Cipher* dan algoritma *columnar transposition* untuk keamanan *text*. Dalam penelitian tersebut, dilakukan proses enkripsi dan dekripsi menggunakan tiga kunci. Astuti *et al.* (2019) menyandikan pesan dengan *Affine Cipher* berdasarkan barisan Fibonacci. Dalam penelitian tersebut pembentukan kunci penggeser dilakukan menggunakan aturan barisan Fibonacci. Pada penelitian lain, Paruliyan *et al.* (2015) merancang dan mengimplementasikan *secure cloud* dengan menggunakan algoritma *Diffie-Hellman* dan *triple* DES *algorithm* (3DES). Dalam penelitian tersebut, *Diffie-Hellman* digunakan untuk proses pertukaran dan pembangkitan kunci, sedangkan *triple* DES *algorithm* (3DES) digunakan untuk proses enkripsi dan dekripsi.

Berdasarkan penjelasan diatas, penulis melakukan pengkodean *monoalphabetic* menggunakan *Affine Cipher* dengan kunci *Diffie-Hellman*. *Diffie-Hellman* digunakan untuk proses pertukaran dan pembangkitan kunci privat, kemudian kunci privat tersebut digunakan sebagai kunci penggeser pada proses enkripsi dan dekripsi menggunakan *Affine Cipher*. Selanjutnya *Affine Cipher* dengan kunci *Diffie-Hellman* dibandingkan dengan *Affine Cipher* dengan kunci biasa.

2. Metode

Data yang digunakan pada penelitian ini adalah *plaintext* (berupa abjad, angka maupun simbol pada ASCII *printable characters*), kunci (bilangan bulat *G* dan *R* dengan syarat GCD(*G*,*R*) =1 dan GCD(*R*,95) = 1) dan dua buah kunci privat (bilangan bulat positif untuk masing-masing pihak (*a* dan *b*)).

2.1. Pertukaran dan Pembangkitan kunci

Proses pertukaran dan pembangkitan kunci dilakukan menggunakan algoritma *Diffie-Hellman*, dimana kedua pihak saling bertukar kunci publik yang telah dikodekan untuk mendapatkan kunci penggeser. Kunci penggeser tersebut akan digunakan pada proses enkripsi dan dekripsi menggunakan *Affine-Cipher*.

Berikut ini merupakan langkah-langkah pada proses pertukaran dan pembangkitan kunci:

Langkah 1. Misalkan Joy dan Kay adalah pihak-pihak yang berkomunikasi. Mula-mula Joy dan Kay menyepakati dua bilangan *G* dan *R*.

Langkah 2. Joy menyiapkan sebuah bilangan bulat positif *a*, kemudian dikodekan dengan Persamaan (3). Hasilnya akan dikirimkan kepada Kay.

$$A = G^a\ mod\ R \quad (3)$$

Langkah 3. Kay menyiapkan sebuah bilangan bulat positif *b*, kemudian dikodekan dengan Persamaan (4). Hasilnya akan dikirimkan kepada Joy.

$$B = G^b\ mod\ R \quad (4)$$

Langkah 4. Joy menerima bilangan *B* dari Kay dan mendeskripsinya dengan Persamaan (5).

$$X = B^a\ mod\ R \quad (5)$$

Langkah 5. Kay menerima bilangan *A* dari Joy dan mendeskripsinya dengan Persamaan (6).

$$X = A^b \ mod \ R \tag{6}$$

Hasil *X* adalah kunci privat yang digunakan sebagai kunci penggeser pada proses enkripsi dan dekripsi menggunakan *Affine Cipher*.

2.2 Enkripsi

Proses enkripsi pada penelitian ini menggunakan *Affine Cipher* dengan kunci penggeser yang telah dikodekan menggunakan algoritma *Diffie-Hellman*.

Berikut merupakan langkah-langkah proses enkripsi:

Langkah 1. Joy menyiapkan pesan (*plaintext*). Kemudian karakter pada *plaintext* dikonversi menjadi bilangan desimal dengan melihat tabel kode ASCII.

Langkah 2. Karakter *plaintext* yang telah dikonversi kemudian dienkripsi menggunakan *Affine Cipher* yaitu dengan Persamaan (7). Rumus yang digunakan mengalami modifikasi berupa adanya pengurangan dan penambahan 32. Hal tersebut dilakukan untuk membuat karakter yang muncul berada pada rentang 32-126 sesuai kode ASCII *printable characters*.

$$C_n = \big((R(D_n - 32) + X) \ mod \ 95\big) + 32 \tag{7}$$

Langkah 3. Bilangan *plaintext* yang telah dienkripsi kemudian dikonversi menjadi karakter pada kode ASCII dan menjadi *ciphertext*. *Ciphertext* dikirim kepada Joy.

2.3 Dekripsi

Proses dekripsi pada penelitian ini menggunakan *Affine Cipher* dengan kunci penggeser yang telah dikodekan menggunakan algoritma *Diffie-Hellman*.

Berikut merupakan langkah-langkah proses dekripsi:

Langkah 1. Kay menerima pesan dari Joy (*ciphertext*). Kemudian karakter *ciphertext* dikonversi menjadi bilangan desimal dengan melihat tabel kode ASCII.

Langkah 2. Karakter *ciphertext* yang telah dikonversi kemudian didekripsi menggunakan *Affine Cipher* yaitu dengan Persamaan (8). Rumus dekripsi juga mengalami modifikasi seperti pada proses enkripsi yaitu berupa pengurangan dan penambahan 32.

$$P_n = (R^{-1}(D_n - 32 - X) \ mod \ 95) + 32 \tag{8}$$

Langkah 3. Bilangan *ciphertext* yang telah didekripsi kemudian dikonversi kembali menjadi karakter pada kode ASCII dan menjadi *plaintext*.

3. Hasil dan Pembahasan

Berikut ini merupakan tabel berisi data yang digunakan pada penelitian ini:

Tabel 1. Data pengkodean

Plaintext	Tika161061@UNEJ
Kunci (*G*,*R*)	68 dan 71
Kunci Privat (*a*,*b*)	8 dan 5

3.1. Pertukaran dan Pembangkitan Kunci

Berikut ini merupakan langkah-langkah dari proses pertukaran dan pembangkitan kunci dari pengkodean menggunakan *Affine cipher* dengan kunci *Diffie-Hellman*:

Langkah 1. Misalkan Joy dan Kay adalah pihak-pihak yang berkomunikasi. Mula-mula Joy dan Kay menyepakati dua bilangan $G = 68$ dan $R = 71$ sebagai kunci. Nilai *G* dan *R* tidak perlu rahasia.

Langkah 2. Joy menyiapkan sebuah bilangan bulat positif a = 8 sebagai kunci privat, kemudian dikodekan dengan Persamaan (3) dan dikirimkan kepada Kay. Hasilnya sebagai berikut:

$$A = 68^8 \ mod \ 71 = 29$$

Langkah 3. Kay juga menyiapkan sebuah bilangan bulat positif b = 5 sebagai kunci privat, kemudian dikodekan dengan Persamaan (4) dan dikirimkan kepada Joy. Hasilnya sebagai berikut:

$$B = 68^5 \ mod \ 71 = 41$$

Langkah 4. Joy menerima bilangan rahasia *B* dari Kay dan mendeskripsi bilangan *B* dengan persamaan (5). Hasilnya sebagai berikut:

$$X = 41^8 \ mod \ 71 = 30$$

Langkah 5. Kay menerima bilangan rahasia *A* dari Joy dan mendeskripsi bilangan *A* dengan persamaan (6) dan hasilnya sebagai berikut:

$$X = 29^5 \ mod \ 71 = 30$$

Hasil *X* adalah kunci privat yang digunakan sebagai kunci penggeser pada proses enkripsi dan dekripsi menggunakan *Affine Cipher*.

3.2. Enkripsi

Berikut ini merupakan langkah-langkah proses enkripsi dari pengkodean menggunakan *Affine cipher* dengan kunci *Diffie-Hellman*:

Langkah 1. Joy menyiapkan *plaintext* **Tika161061@UNEJ**. Kemudian karakter pada *plaintext* dikonversi menjadi bilangan desimal dengan melihat tabel kode ASCII *printable characters*. Hasil konversi karakter *ciphertext* dapat dilihat pada Tabel 2.

Tabel 2. Konversi *plaintext* ke desimal

Karakter	Desimal	Karakter	Desimal	Karakter	Desimal
T	84	6	54	@	64
i	105	1	49	U	85
k	107	0	48	N	78
a	97	6	54	E	69
1	49	1	49	J	74

Langkah 2. Setelah dikonversi menjadi bilangan desimal, selanjutnya bilangan *plaintext* dienkripsi menggunakan *Affine Cipher* dengan Persamaan (7). Modulo 95 digunakan untuk menyesuaikan dengan jumlah karakter ASCII *printable characters*. Kunci yang digunakan adalah *R* = 71 sebagai pengali dan *X* = 30 sebagai penggeser. Berikut merupakan proses enkripsinya:

$$\begin{aligned} \text{T} &= \big((71(84-32)+30) \ mod \ 95\big) + 32 \\ &= ((71.52+30) \ mod \ 95) + 32 \\ &= ((3692+30) mod \ 95) + 32 \\ &= (3722 \ mod \ 95) + 32 \\ &= 17 + 32 \\ &= 49 \end{aligned}$$

$$\begin{aligned} \text{i} &= \big((71(105-32)+30) \ mod \ 95\big) + 32 \\ &= ((71.73+30) \ mod \ 95) + 32 \\ &= ((5183+30) mod \ 95) + 32 \\ &= (5213 \ mod \ 95) + 32 \\ &= 83 + 32 \\ &= 115 \\ &\cdot \\ &\cdot \\ &\cdot \\ &\cdot \end{aligned}$$

$$\begin{aligned} \text{J} &= \big((71(74-32)+30) \ mod \ 95\big) + 32 \\ &= ((71.42+30) \ mod \ 95) + 32 \\ &= ((2982+30) mod \ 95) + 32 \\ &= (3012 \ mod \ 95) + 32 \\ &= 67 + 32 \\ &= 99 \end{aligned}$$

Langkah 3. Setelah proses enkripsi dilakukan, diperoleh bilangan *ciphertext* **49 115 67 117 34 104 34 58 104 34 54 120 98 124 99**. Selanjutnya bilangan *ciphertext* dikonversi menjadi karakter pada kode ASCII menjadi *ciphertext* dan dikirim kepada Kay sebagai penerima pesan. Hasil konversi bilangan *ciphertext* dapat dilihat pada Tabel 3.

Tabel 3. Konversi bilangan *ciphertext* ke kode ASCII

Desimal	Karakter	Desimal	Karakter	Desimal	Karakter
49	1	104	h	54	6
115	s	34	“	120	x
67	C	58	:	98	b
117	u	104	h	124	|
34	“	4	“	99	c

3.3. Dekripsi

Berikut ini merupakan langkah-langkah proses dekripsi dari pengkodean menggunakan *Affine Cipher* dengan kunci *Diffie-Hellman*:

Langkah 1. Kay menerima *ciphertext* **1sCu”h”:h”6xb|c**. Karakter *ciphertext* dikonversi menjadi bilangan desimal dengan melihat tabel kode ASCII *printable characters*. Hasil konversi karakter *ciphertext* dapat dilihat pada Tabel 4.

Tabel 4. Konversi *ciphertext* ke desimal

Karakter	Desimal	Karakter	Desimal	Karakter	Desimal
1	49	h	104	6	54
s	115	“	34	x	120
C	67	:	58	b	98
u	117	h	104	|	124
“	34	“	4	c	99

Langkah 2. Setelah dikonversi menjadi bilangan desimal, selanjutnya bilangan *ciphertext* didekripsi menggunakan *Affine Cipher* dengan Persamaan (8). Kunci yang digunakan adalah invers dari $R = 71$ yaitu $R^{-1} = 91$ sebagai pengali dan $X = 30$ sebagai penggeser. Berikut merupakan proses dekripsinya:

$$\begin{aligned} 1 &= (91(49-32-30)\ mod\ 95)+32 \\ &= (91.(-13)\ mod\ 95)+32 \\ &= ((-1183) mod\ 95)+32 \\ &= 52+32 \\ &= 84 \end{aligned}$$

$$\begin{aligned} \text{s} &= (91(115-32-30)\ mod\ 95)+32 \\ &= (91.53\ mod\ 95)+32 \\ &= (4823\ mod\ 95)+32 \\ &= 73+32 \\ &= 105 \end{aligned}$$

$$\begin{aligned} \text{C} &= (91(67-32-30)\ mod\ 95)+32 \\ &= (91.5\ mod\ 95)+32 \\ &= (455\ mod\ 95)+32 \\ &= 75+32 \\ &= 107 \end{aligned}$$

$$\begin{aligned} \text{"} &= (91(34-32-30)\ mod\ 95)+32 \\ &= (91\ .\ 2\ mod\ 95)+32 \\ &= (182\ mod\ 95)+32 \\ &= 87+32 \\ &= 119 \end{aligned}$$

$$\vdots$$

$$\begin{aligned} c &= (91(\ 99\ -32-30)\ mod\ 95)+32 \\ &= (91.37\ mod\ 95)+32 \\ &= (3367\ mod\ 95)+32 \\ &= 42+32 \\ &= 74 \end{aligned}$$

Langkah 3. Setelah proses dekripsi dilakukan, diperoleh bilangan *plaintext* **84 105 107 97 49 54 49 48 54 49 64 85 78 69 74**. Selanjutnya bilangan *plaintext* dikonversi menjadi karakter pada kode ASCII menjadi *plaintext*. Hasil konversi bilangan *plaintext* dapat dilihat pada Tabel 5.

Tabel 5. Konversi bilangan *plaintext* ke kode ASCII

Desimal	Karakter	Desimal	Karakter	Desimal	Karakter
84	T	54	6	64	@
105	i	49	1	85	U
107	k	48	0	78	N
97	a	54	6	69	E
49	1	49	1	74	J

3.4. Pembahasan

Terdapat beberapa poin pembahasan pada penelitian ini, yaitu sebagai berikut:

- Pengkodean menggunakan *Affine Cipher* dengan kunci biasa terdiri dari dua proses, yaitu proses enkripsi dan proses dekripsi. Sedangkan pengkodean menggunakan *Affine Cipher* dengan kunci *Diffie-Hellman* terdiri dari tiga proses, yaitu proses pertukaran dan pembangkitan kunci, proses enkripsi dan proses dekripsi. Proses pertukaran dan pembangkitan kunci dilakukan dengan *Diffie-Hellman* untuk mendapatkan kunci penggeser yang akan digunakan pada proses enkripsi dan dekripsi menggunakan *Affine Cipher*.
- Pada pengkodean menggunakan *Affine Cipher* dengan kunci biasa, kunci yang digunakan terdiri dari dua buah bilangan. Satu bilangan sebagai kunci pengali, dan bilangan lain sebagai kunci penggeser. Sedangkan pada pengkodean menggunakan *Affine Cipher* dengan kunci *Diffie-Hellman*, kunci yang digunakan merupakan empat buah bilangan. Satu bilangan sebagai kunci yang dipangkatkan pada saat mengkodekan kunci publik (G), satu bilangan sebagai modulo pada saat mengkodekan kunci publik juga sebagai kunci pengali pada proses enkripsi dan dekripsi (R) dan dua buah bilangan sebagai kunci privat (a,b), dimana kunci privat tersebut hanya dimiliki dan diketahui oleh masing-masing pihak yang bertukar pesan.
- Pada pengkodean menggunakan *Affine Cipher* dengan kunci biasa, kunci penggeser langsung ditentukan tanpa dilakukan pengkodean terlebih dahulu. Sehingga apabila *ciphertext* dan dua buah kunci diketahui oleh pihak ketiga, pesan dapat langsung dipecahkan. Sedangkan pada pengkodean menggunaan *Affine Cipher* dengan kunci *Diffie-Hellman*, kunci penggeser didapatkan melalui proses pengkodean dan pertukaran kunci publik dengan *Diffie-Hellman*. Pengkodean tersebut menggunakan kunci privat yang sifatnya rahasia. Sehingga apabila *ciphertext*, dua buah kunci dan dua buah kunci publik diketahui oleh pihak ketiga, pesan tetap tidak dapat di pecahkan.

4. Simpulan

Pada pengkodean menggunakan *Affine Cipher* dengan kunci biasa, kunci yang digunakan terdiri dari dua buah bilangan. Sedangkan pada pengkodean menggunakan *Affine Cipher* dengan kunci *Diffie-Hellman*, kunci yang digunakan merupakan empat buah bilangan, dengan dua buah bilangan sebagai kunci privat.

Pada pengkodean menggunakan *Affine Cipher* dengan kunci biasa, kunci penggeser langsung ditentukan tanpa dilakukan pengkodean terlebih dahulu. Sedangkan pada pengkodean menggunaan *Affine Cipher* dengan kunci *Diffie-Hellman*, kunci penggeser didapatkan dengan *Diffie-Hellman*.

Pengkodean menggunaan *Affine Cipher* dengan kunci *Diffie-Hellman* lebih aman daripada dengan kunci biasa. Karena jika pada pengkodean *Affine Cipher* dengan kunci biasa salah satu kunci diketahui, maka pesan dapat dengan mudah dipecahkan. Tetapi jika dengan kunci *Diffie-Hellman* meskipun kedua kunci diketahui dan selama kunci privat tetap rahasia maka pesan tetap tidak dapat dipecahkan.

Daftar Pustaka

Ahirwal, R. R., & Ahke, M. (2013). Elliptic Curve Diffie-Hellman Key Exchange Algorithm for Securing Hypertext Information on Wide Area Network. *International Journal of Computer Science and Information Technologies, 4*(2), 363-368.

Astuti, L. F., Santoso, K. A., & Kamsyakawuni, A. (2019). Pengamanan Polyalphabetic dengan Affine Cipher Berdasarkan Baris Fibonacci. *Majalah Ilmiah Matematika dan Statistika*, *19*(2), 95-103.

Paruliyan, B. P., S. M. Nasution, & T. W. Purboyo. (2015). Perancangan dan Implementasi Secure Cloud dengan Menggunakan Diffie-Hellman Key Exchange dan Triple DES Algorithm (3DES*). E-Proceeding of Engineering, 2*(2), 3808-3815.

Permanasari, Y., & E. Harahap. (2018). Kriptografi Polyalphabetic. *Jurnal Matematika*, 17(1), 31-34.

Sadikin, R. (2012). *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: C.V ANDI Offset.

Siregar, I. M. (2019). Penerapan Algoritma Affine Cipher dan Algoritma Columnar Transposition dalam Keamanan Teks. *Jurnal Informatika Kaputama (JIK)*, *3*(1), 6-12.