MODIFIKASI ALGORITMA AFFINE CIPHER UNTUK MENGAMANKAN DATA

Adnan Buyung Nasution

Program Studi Sistem Informasi Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan
JL. IAIN No.1, Medan 20235 Sumatera Utara, Indonesia
adnanbuyungnasution@uinsu.ac.id

Abstract - The Affine Cipher algorithm is part of classical cryptography which is a method of substitution in data security, namely shifting the plaintext by multiplying the plaintext value with a key with a prime number value. The weakness of the Affine Cipher Algorithm is its easy to use key. Therefore, the author wants to modify the Affine Cipher algorithm by dividing the plaintext into blocks with the length of the specified key (k). After that, change the position of each character by reversing its position and then do the Affine Cipher algorithm process. This algorithm modification can hide the message and return it in the form of original data (plaintext) without causing data loss. This research concludes that the encrypted message by modifying the affine cipher becomes more random. The process in this research only uses the alphabetic characters (a to z) so it is hoped that it can add other characters such as numbers, symbols and others.
Keywords **-** Affine Cipher, Data Security, Plaintext, Weaknesses, Affine Cipher Modification

Abstrak - Algoritma Affine Cipher merupakan bagian dari kriptografi klasik yang merupakan metode substitusi dalam pengamanan data, yaitu melakukan pergeseran plaintext dengan mengalikan nilai plaintext dengan kunci bernilai bilangan prima. Kelemahan Algoritma Affine Cipher adalah kunci yang mudah digunakan. Oleh sebabnya, penulis ingin memodifikasi algoritma Affine Cipher dengan membagi *plaintext* menjadi blok-blok dengan panjang sebanyak kunci (k) yang telah ditentukan. Setelah itu, merubah posisi setiap karakter dengan cara membalikkan posisinya dan selanjutnya melakukan proses algoritma Affine Cipher. Modifikasi algoritma ini dapat menyembunyikan pesan dan mengembalikan dalam bentuk data asli (plaintext) tanpa menyebabkan kehilangan data. Penelitian ini memiliki kesimpulan bahwa pesan yang terenkripsi dengan memodifikasi affine cipher menjadi lebih teracak. Proses pada penelitian ini hanya menggunakan karakter alphabet (a hingga z) sehingga diharapkan dapat menambahkan karakter lainnya seperti angka, simbol dan lainnya.
Kata Kunci **-** Affine Cipher, Pengamanan Data, *Plaintext*, Kelemahan, Modifikasi Affine Cipher

I. PENDAHULUAN

Saat ini, ada banyak orang atau organisasi yang tidak berkepentingan mengambil dan menyalahgunakan data orang lain sehingga data tidak lagi terjaga. Banyak cara yang dapat Anda lakukan untuk mengirim informasi berupa pesan atau data yang akan dikirim. [1] Adapun proses yang digunakan yaitu kriptografi.

Kriptografi adalah ilmu yang mengajarkan bagaimana cara agar data atau pesan tetap terjaga secara aman saat data atau pesan terkirim dari pemberi pesan ke penerima pesan tanpa ada gangguan dari pihak ketiga. [2] Adapun yang merupakan salah satu algoritma pada kriptografi adalah Algoritma Affine Cipher. Algoritma Affine Cipher adalah pengembangan dari metode *Caesar Cipher* dimana pesan asli akan dikalikan nilai suatu *integer* kemudian ditambahkan dengan pergeseran sebanyak kunci *integer*. [3]

Algoritma Affine Cipher adalah salah satu yang kriptografi klasik membentuk metode substitusi data, yaitu dengan mengalikan plaintext dengan nilai prima dan menambahkannya ke pergeseran. [4] Algoritma Affine Cipher menggunakan kombinasi dua kunci yang berbeda. [5]

Kelemahan dari Algoritma Affine Cipher adalah kunci yang mudah digunakan yaitu dengan menggunakan 2 buah kombinasi kunci sehingga plainteks mudah dilakukan proses enkripsi dan deskripsi. Oleh karena itu, penulis tertarik untuk melakukan modifikasi pada algoritma Affine Cipher untuk meningkatkan keamanan data.

Modifikasi yang dilakukan dengan menggunakan kombinasi 3 kunci dimana plainteks (data) akan dibagi menjadi blok-blok dengan panjang sebanyak kunci (k) yang telah ditentukan. Setelah itu, merubah posisi setiap karakter yang ada di setiap blok dengan cara membalikkan posisinya. Karakter yang telah diubah kemudian dikalikan dengan kunci yang bernilai prima kemudian ditambahkan dengan kunci selanjutnya.

II. METODE PENELITIAN

A. Algoritma Affine Cipher

Affine Cipher ialah sebuah bagian cipher monoalphabetic yang mana huruf-huruf dalam alphabetik dibuat ke dalam bentuk numerik, kemudian dienkripsikan dengan fungsi suatu matematika sangat sederhana, dan diubah kembali dalam bentuk karakter kata. [6] Algoritma Affine Cipher ialah algoritma yang *file* atau teks awal yang dikirim berubah dimana awalnya dimengerti maknanya oleh manusia menjadi file atau teks yang terenkripsi kemudian dimasukan dlam media penyimpanan. [7] Affine cipher dalam algoritma affine merupakan pengembangan algoritma Caesar Cipher dimana plaintext (P) dikalikan nilai (b) kemudian ditambahkan ke pergeseran (k). [8] Plaintext P yang hasilnya berupa chipertext C dapat diekspresikan (enkripsi) dengan fungsi berikut:

$$C = ((b \times P) + k) \text{ modulus } 26 \quad \ldots\ldots\ldots (1)$$

Di mana 26 ialah panjang karakter alfabet, dimanaa persamaan 1 untuk tahap enkripsi. Tahap dekripsi dapat menggunakan dari persamaan 2 berikut:

$$P = b^{-1} (C-k) \text{ modulus } 26 \quad \ldots\ldots\ldots\ldots (2)$$

b merupakan sebuah bilangan relatif prima dari 26. Dapat disimpulkan bahwa faktor persekutuan terbesar atau gcd (b, 26) harus memiliki hasil sama dengan 1.

Ukuran setiap alfabet atau alfabet yang digunakan pada Affine Cipher antara lain [4]:

Tabel 1 : Nilai Alfabet

Huruf	Nilai
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7
I	8
J	9
K	10
L	11
M	12
N	13
O	14
P	15
Q	16
R	17
S	18
T	19
U	20
V	21
W	22
X	23
Y	24
Z	25

B. Invers Modulo

Pada aritmetika yang bilangan riil, invers pembagian adalah perkalian begitu juga sebaliknya. Contohnya, bilangan 5 dapat diinverskan menjadi 1/5 sebab 5 x 1/5 = 1. Di aritmetika untuk modulo, modulo juga memiliki balikan modulo yang disebut dengan invers modulo. Masalah dalam menghitung invers pada modulo sedikit rumit.

Jika b dan panjang karakter alfabet relatif prima dan panjang karakter alfabet> 1, maka invers dari modulus panjang karakter alfabet dapat ditemukan. Invers dari b (modulus panjang karakter alfabet), disebut juga sebagai inversi multiplikasi, di mana bilangan bulat b-1 sedemikian rupa : [3]

$$b^{-1} = b.\ n \text{ modulus } 26 = 1 \quad \ldots\ldots\ldots\ldots\ldots (3)$$

b adalah bilangan bulat, angka 26 adalah jumlah alfabet dan n adalah nilai untuk memeriksa.

C. *Great Common Divisor* (Faktor Persekutuan Terbesar)

Faktor Persekutuan Terbesar (FPB) ialah suatu bagian terbesar dalam set pembagi 2 (dua) bilangan bulat. Keduanya mungkin mempunyai beberapa bagian pembagi sama yaitu satu dari bilangan tersebut [3]. Contoh faktor persekutuan terbesar untuk 20 dan 30.

Divisior oleh 20 = {1,2,4,5,10,20}
Divisior dengan 30 = {1,2,3,5,6,10,15,30}.
Set angka yang sama yaitu {1,2,5,10}
Jumlah terbesar dari himpunan yang sama adalah 10. Kemudian, FPB Dari 20 dan 30 adalah GCD (20,30) = 10.

D. Modifikasi Algoritma Affine Cipher

Modifikasi affine cipher adalah sebuah kriptografi dimana aturannya sama dengan affine cipher yang sudah dipaparkan pada sub bagian sebelumnya kemudian dimodifikasi. Affine Cipher kurang aman untuk sandi substitusi karena rentan terhadap semua serangan yang bekerja terhadap sandi substitusi, selain serangan lainnya. [9] Dalam penelitian sebelumnya, Hartini melakukan modifikasi terhadap pembalikan plaintext abjad sebelum proses enkripsi Affine Cipher. [10] Dari penelitian diatas, penulis mengembangkan modifikasi plaintext yang akan dibagi oleh k pada setiap bagian. Kemudian posisi terbalik dari setiap bagian

abjad. Jika akhir k jumlah abjad bukan bagian akhir jumlah abjad sesuai dengan sisa abjad yang terkandung maka abjad setiap bagian posisi terbalik. Contoh kata "MODIFICATION" dengan membagi abjad menjadi 5 sehingga hasilnya adalah "MODIF ICATI ON".

III. HASIL DAN PEMBAHASAN

A. Perhitungan Modifikasi Affine Cipher

Pada penjelasan mengenai modifikasi Affine Cipher diatas, terdapat dua tahap yaitu tahap enkripsi dan tahap deskripsi.

1. Tahapan Enkripsi

Suatu tahapan dimana mengubah kode yang dapat dimengerti maknanya (plaintext) menjadi kode yang sulit dimengerti maknanya (ciphertext). Contoh kasus, jika plaintext seperti dibawah ini :

Plaintext : CRYPTOGRAPHY
b : 9
k : 12
Abjad pembagi : 4

a. Periksa nilai faktor persekutuan terbesar adalah sama dengan 1 (gcd (b, 26) = 1. Divisior 9 = (1,3,9) dan divisior 26 = (1,2,13,26) jadi gcd (9,26) = 1.
b. Lakukan pembagian plaintext menjadi 4 abjad di setiap bagian. Jadi hasilnya menjadi CRYP TOGR APHY.
c. Lakukan pembalikan abjad di setiap bagian. Hasilnya menjadi PYRC RGOT YHPA.
d. Periksa nilai alfabet dari abjad di mana pada Tabel 1 menunjukkan bahwa P = 15, Y = 24, R = 17, C = 2, R = 17, G = 6, O = 14, T = 19, Y = 24, H = 7, P = 15, A = 0.
e. Kemudian lakukan perhitungan $C_i = ((a \times P_i) + k) \bmod 26$ dan periksa pada tabel 1 alfabet dari i, hasil ciphertext adalah sebagai berikut.

$C_1 = ((bxP_{1)}+k) \bmod 26$
= ((9xP)+12) modulus 26
= ((9x15)+12) modulus 26
= (135+12) modulus 26
= 147 modulus 26
= 17
= R

$C_2 = ((bxP_{2)}+k) \bmod 26$
= ((9xY+12) modulus 26
= ((9x24)+12) modulus 26
= (216+12) modulus 26
= 228 modulus 26
= 20
= U

$C_3 = ((bxP_{3)}+k) \bmod 26$
= ((9xR)+12) modulus 26
= ((9x17)+12) modulus 26
= (153+12) modulus 26
= 165 modulus 26
= 9
= J

$C_4 = ((bxP_{4)}+k) \bmod 26$
= ((9xC)+12) modulus 26
= ((9x2)+12) modulus 26
= (18+12) modulus 26
= 30 modulus 26
= 4
= E

$C_5 = ((bxP_{5)}+k) \bmod 26$
= ((9xR)+12) modulus 26
= ((9x17)+12) modulus 26
= (153+12) modulus 26
= 165 modulus 26
= 9
= J

$C_6 = ((bxP_{6)}+k) \bmod 26$
= ((9xG)+12) modulus 26
= ((9x6)+12) modulus 26
= (54+12) modulus 26
= 66 modulus 26
= 14
= O

$C_7 = ((bxP_{7)}+k) \bmod 26$
= ((9xO)+12) modulus 26
= ((9x14)+12) modulus 26
= (126+12) modulus 26
= 138 modulus 26
= 8
= I

$C_8 = ((bxP_{8)}+k) \bmod 26$
= ((9xT)+12) modulus 26
= ((9x19)+12) modulus 26
= (171+12) modulus 26
= 183 modulus 26
= 1
= B

$C_9 = ((bxP_{9)}+k) \bmod 26$
= ((9xY+12) modulus 26
= ((9x24)+12) modulus 26
= (216+12) modulus 26
= 228 modulus 26
= 20
= U

$C_{10} = ((bxP_{10)}+k) \bmod 26$
= ((9xH)+12) modulus 26
= ((9x7)+12) modulus 26
= (63+12) modulus 26
= 75 modulus 26
= 23
= X

$C_{11} = ((bxP_{11)}+k) \bmod 26$
= ((9xP)+12) modulus 26
= ((9x15)+12) modulus 26
= (135+12) modulus 26
= 147 modulus 26
= 17
= R

$C_{12} = ((bxP_{12)}+k) \bmod 26$

= ((9xA)+12) modulus 26
= ((9x0)+12) modulus 26
= (0+12) modulus 26
= 12 modulus 26
= 12
= M

Jadi, hasil Ciphertext dari Plaintext "CRYPTOGRAPHY" is **RUJEJOIBUXRM**.

2. Tahapan Deskripsi

Kebalikan dari tahapan enkripsi, tahapan deskripsi adalah mengubah kode yang sulit dimengerti maknanya (ciphertext) menjadi kode yang dapat dimengerti maknanya (plaintext). Contoh kasus. Jika suatu ciphertext berikut:

Ciphertext : RUJEJOIBUXRM
b : 9
k : 12
Abjad pembagi : 4

a. Periksa nilai faktor persekutuan terbesar adalah sama dengan 1 (gcd (b, 26) = 1. Divisior 9 = (1,3,9) dan divisior 26 = (1,2,13,26) jadi gcd (9,26) = 1.
b. Kemudian tahapan selanjutnya hitung invers modulo dimana a. n mod 26 = 1 yang akan dilakukan adalah :
   Ketika n = 1
   b^{-1} = b. n modulus 26
   9^{-1} = 9.1 modulus 26
   = 9 modulus 26
   = 9
   n = 1 tidak menghasilkan modulo 1.
   Ketika n = 2
   b^{-1} = b. n modulus 26
   9^{-1} = 9.2 modulus 26
   = 18 modulus 26
   = 18
   n = 2 tidak menghasilkan modulo 1.
   ketika n = 3
   b^{-1} = b. n modulus 26
   9^{-1} = 9.3 modulus 26
   = 27 modulus 26
   = 1
   n = 3 menghasilkan modulo 1 maka invers modulo 9^{-1} adalah 3.
c. Periksa nilai alfabet dari kata-kata yang ditunjukkan pada tabel 1 R = 17, U = 20, J = 9, E = 4, J = 9, O = 14, I = 8, B = 1, U = 20, X = 23, R = 17, M = 12.
d. Lakukan penghitungan plaintext di mana P = b^{-1} (C-k) modulus 26. Jika hasil yang diperoleh negatif (-) maka tambahkan 26 hingga diperoleh hasilnya positif (+). Kemudian, hasil perhitungan modulo dapat dicek pada tabel 1 dari alfabet sehingga menghasilkan nilai plaintext.
   P_1= b^{-1}(C_1-k) modulus 26
   = 9^{-1} (R-12) modulus 26
   = 3 (17-12) modulus 26
   = 3 (5) modulus 26
   = 15 modulus 26
   = 15
   = P
   P_2= b^{-1}(C_2-k) modulus 26
   = 9^{-1} (U-12) modulus 26
   = 3 (20-12) modulus 26
   = 3 (8) modulus 26
   = 24 modulus 26
   = 24
   = Y
   P_3= b^{-1}(C_3-k) modulus 26
   = 9^{-1} (J-12) modulus 26
   = 3 (9-12) modulus 26
   = 3 (-3) modulus 26
   = -9 modulus 26
   = -9+26 modulus 26
   = 17 modulus 26
   = 17
   = R
   P_4= b^{-1}(C_4-k) modulus 26
   = 9^{-1} (E-12) modulus 26
   = 3 (4-12) modulus 26
   = 3 (-8) modulus 26
   = -24 modulus 26
   = -24 + 26 modulus 26
   = 2 modulus 26
   = 2
   = C
   P_5= b^{-1}(C_5-k) modulus 26
   = 9^{-1} (J-12) modulus 26
   = 3 (9-12) modulus 26
   = 3 (-3) modulus 26
   = -9 modulus 26
   = -9+26 modulus 26
   = 17 modulus 26
   = 17
   = R
   P_6= b^{-1}(C_6-k) modulus 26
   = 9^{-1} (O-12) modulus 26
   = 3 (14-12) modulus 26
   = 3 (2) modulus 26
   = 6 modulus 26
   = 6
   = G
   P_7= b^{-1}(C_7-k) modulus 26
   = 9^{-1} (I-12) modulus 26
   = 3 (8-12) modulus 26
   = 3 (-4) modulus 26
   = -12+26 modulus 26
   = 14 modulus 26
   = 14
   = O
   P_8= b^{-1}(C_8-k) modulus 26
   = 9^{-1} (B-12) modulus 26
   = 3 (1-12) modulus 26

= 3 (-11) modulus 26
= -33 modulus 26
= -33+26 modulus 26
= -7+26 modulus 26
= 19 modulus 26
= 19
= T

P_9= b^{-1}(C_9-k) modulus 26
= 9^{-1} (U-12) modulus 26
= 3 (20-12) modulus 26
= 3 (8) modulus 26
= 24 modulus 26
= 24
= Y

P_{10}= b^{-1}(C_{10}-k) modulus 26
= 9^{-1} (X-12) modulus 26
= 3 (23-12) modulus 26
= 3 (11) modulus 26
= 33 modulus 26
= 7
= H

P_{11}= b^{-1}(C_{11}-k) modulus 26
= 9^{-1} (R-12) modulus 26
= 3 (17-12) modulus 26
= 3 (5) modulus 26
= 15 modulus 26
= 15
= P

P_{12}= b^{-1}(C_{12}-k) modulus 26
= 9^{-1} (M-12) modulus 26
= 3 (12-12) modulus 26
= 3 (0) modulus 26
= 0 modulus 26
= 0
= A

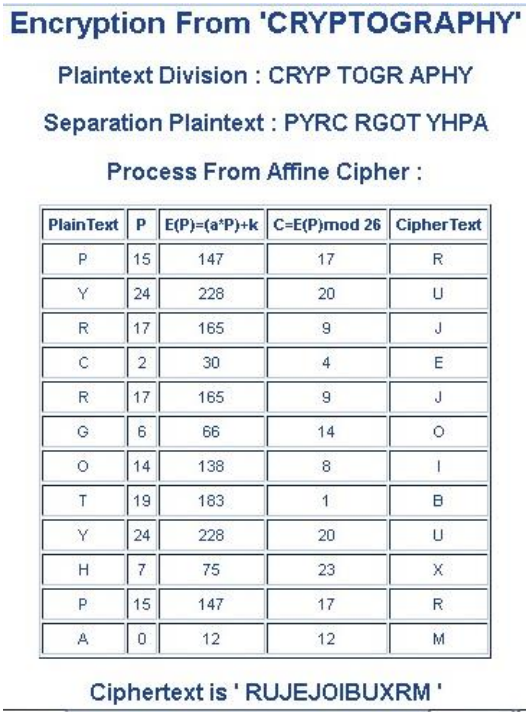
e. Lakukan pembagian dari dekripsi PYRCRGOTYHPA menjadi 4 abjad di setiap bagian.
Jadi hasilnya menjadi PYRC RGOT YHPA.
f. Lakukan pembalikan abjad di setiap bagian. Hasilnya menjadi CRYP TOGR APHY.

Jadi, hasil plaintext dari ciphertext "RUJEJOIBUXRM" is **CRYPTOGRAPHY**.

B. Tampilan Program

1. Tampilan Tahapan Enkripsi

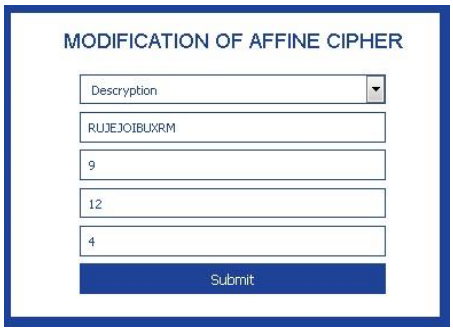
Gambar 1 Tampilan Dari Awal Enkripsi



PlainText	P	E(P)=(a*P)+k	C=E(P)mod 26	CipherText
P	15	147	17	R
Y	24	228	20	U
R	17	165	9	J
C	2	30	4	E
R	17	165	9	J
G	6	66	14	O
O	14	138	8	I
T	19	183	1	B
Y	24	228	20	U
H	7	75	23	X
P	15	147	17	R
A	0	12	12	M

Gambar 2 Tampilan Tahapan Enkripsi

2. Tampilan Tahapan Deskripsi



Gambar 3 Tampilan Awal Deskripsi

Invers Modulo From a^-1

Invers Modulo From (9)^ -1 is 3

Descryption 'RUJEJOIBUXRM'

Process From Affine Cipher :

CipherText	C	D(C)a^-1(C-k)	P=D(C)mod 26	PlainText
R	17	15	15	P
U	20	24	24	Y
J	9	-9	17	R
E	4	-24	2	C
J	9	-9	17	R
O	14	6	6	G
I	8	-12	14	O
B	1	-33	19	T
U	20	24	24	Y
X	23	33	7	H
R	17	15	15	P
M	12	0	0	A

Separation Plaintext : PYRC RGOT YHPA

Plaintext Divisions : CRYP TOGR APHY

Plaintext is ' CRYPTOGRAPHY '

Gambar 4 Tampilan Tahapan Deskripsi

IV. KESIMPULAN

Tahapan algoritma modifikasi Affine Cipher dengan menggunakan 3 kombinasi kunci merubah pesan sehingga sulit dimengerti maknanya dan pesan terebut dapat dikembalikan menjadi pesan yang dapat dimengerti. Pesan dapat dienkripsi sehingga keamanan data yang lebih acak dapat ditingkatkan dengan memodifikasi Affine Cipher. Proses pada penelitian ini hanya menggunakan karakter alphabet (a hingga z) sehingga diharapkan dapat menambahkan karakter lainnya seperti angka, simbol dan lainnya. Modifikasi algoritma Affine Cipher tidak dapat digunakan jika nilai a relatif tidak prima ke 26.

REFERENSI

[1] A. B. Nasution, "Implementasi Pengamanan Data Dengan Menggunakan Algoritma Caesar Cipher Dan Transposisi Cipher," *J. Teknol. Inf.*, vol. 3, no. 1, pp. 1–6, 2019.

[2] S. Wibowo, F.E. Nilawati and Suharnawi, "Implementasi Enkripsi Deskripsi Algoritma Affine Cipher Berbasis Android", *Techno.COM*, vol. 13, no. 4, pp. 215-221, 2014.

[3] D. Rachmawati and A Candra, "Implementasi Kombinasi Caesar dan Affine Cipher untuk Keamanan Data Teks", *Jurnal Edukasi dan Penelitian Informatika* (JEPIN), vol. 1, no. 2, pp. 60-63, 2015.

[4] Rahima, "Implementasi Penyembunyian dan Penyandian Pesan Pada Citra Menggunakan Algoritma Affine Cipher dan Metode Least Significant Bit", *Pelita Informatika Budi Darma*, vol. VI, no. 1, pp. 144-148, 2014.

[5] M. Kaushik, "Comparative Analysis of Exhaustive Search Algorithm with ARPS Algorithm for Motion Estimation", *International Journal of Applied Information System* (IJAIS), vol. 1, no. 6, pp. 16-19, 2012.

[6] S. Shukla and P.K. Verma, "Implementation of Affine Substitution Cipher with Keyed Transposition Cipher for Enhancing Data Security", *International Journal of Advanced Research in Computer Science and Software Engineering*. vol. 4, iss. 1, pp. 236-241, 2014.

[7] M.R. Darmawan and Windarto, "Implementasi Algoritma Kriptografi Vigenere Cipher Dan Affine Cipher untuk Mengamankan Pesan Pada Aplikasi Chatting Berbasis Android", *SKANIKA*, vol. 1, no. 2, pp. 584-590, 2018.

[8] Nurjamiyah, "Implementasi Algoritma Affine Cipher untuk Keamanan Data", *Query : Jurnal Sistem Informasi*, vol. 01, no. 01, pp. 51-59, 2020.

[9] Y. Rajput, D. Naik and C. Mane, "An Improved Cryptographic Technique to Encrypt Text using Double Encryption", *International Journal of Computer Application*, vol. 56, no. 6, pp. 25-28, 2014.

[10] Hartini and S. Primaini, "Kriptografi Password Menggunakan Modifikasi Affine Cipher", *Jurnal Sigmata*, vol. 2, no. 1, pp. 40-50, 2014.