Gusmana, R., Haryansyah, H. and Fitria, F. (2022) "Implementasi Algoritma *Affine Cipher* dan *Caesar Cipher* dalam Mengamankan Data Teks", Sebatik, 26(2)
Submitted: 2022-11-17 Accepted: 2022-12-14 Published: 2022-12-21

Sebatik Vol. 26 No. 2 Desember 2022
ISSN: 1410-3737(p) 2621-069X(e)

DOI:10.46984/sebatik.v26i2.2084

IMPLEMENTASI ALGORITMA *AFFINE CIPHER* DAN *CAESAR CIPHER* DALAM MENGAMANKAN DATA TEKS

Roman Gusmana[1)], Haryansyah[2], dan Fitria[3]

[1,2,3]Teknik Informatika, STMIK PPKIA Tarakanita Rahmawati
[1,2,3]Jl. Yos Sudarso No. 6, Tarakan
E-mail: roman@ppkia.ac.id[1)], ary@ppkia.ac.id[2)], fitria@ppkia.ac.id[3)]

ABSTRAK

Maraknya kasus pencurian data akhir-akhir ini tentu saja menimbulkan berbagai macam reaksi di masyarakat. Sebagian merasa takut apabila data tersebut disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab, sementara lainnya tentu akan semakin waspada sehingga berupaya memaksimalkan perlindungan atas data-data tersebut. Salah satu upaya perlindungan yang dapat dilakukan yaitu memperkuat keamanan data dengan menerapkan teknik penyamaran data yang disebut dengan kriptografi. Hal ini sebagai antisipasi apabila terjadi kebocoran data, informasi yang tersimpan didalamnya akan tetap aman karena terlihat dalam bentuk tak beraturan. Kriptografi merupakan suatu cara yang dapat digunakan untuk menyandikan informasi dengan menggunakan algoritma tertentu. Fokus penelitian ini adalah melakukan penyandian data teks dengan menerapkan kombinasi 2 (dua) metode kriptografi yang berfokus pada metode *Affine Cipher* dan *Caesar Cipher*. Proses diawali dengan enkripsi data menggunakan *Affine Cipher* yang selanjutnya dari hasil enkripsi tersebut, dilakukan proses enkripsi kembali dengan menggunakan *Caesar Cipher*. Sementara untuk proses dekripsi data dilakukan dengan menggunakan *Caesar Cipher* terlebih dahulu, kemudian hasil dekripsi tersebut dilanjutkan kembali proses dekripsinya dengan menggunakan *Affine Cipher*. Kombinasi metode kriptografi dilakukan untuk lebih memperkuat pengamanan terhadap sebuah data. Hasil dari penelitian ini menyatakan bahwa kombinasi metode *Affine Cipher* dan *Caesar Cipher* dapat dilakukan untuk melakukan enkripsi dan dekripsi terhadap sebuah data.

Kata Kunci: *Affine, Caesar, Kriptografi, Keamanan Data, Penyandian, Enkripsi, Dekripsi*

1. PENDAHULUAN

Maraknya kasus pencurian data tentu saja menimbulkan berbagai macam reaksi di masyarakat. Sebagian tentu merasa takut apabila data tersebut disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab, sementara lainnya tentu akan semakin waspada sehingga berupaya memaksimalkan perlindungan atas data-data tersebut.

Berdasarkan catatan Tempo, dari Januari hingga September 2022, sedikitnya telah terjadi tujuh kasus besar dugaan kebocoran data telah terjadi. Data tersebut diantaranya adalah data perbankan, data pasien, data pelamar kerja di suatu perusahaan BUMN, data perusahaan, serta data pelanggan di beberapa perusahaan penyedia layanan jasa. Bahkan di beberapa waktu lalu, data *SIM Card* dan data pribadi juga tak luput dari target operasi kegiatan pencurian ini.

Hal ini tentu berdampak pada hilangnya kepercayaan dan rasa aman kepada perusahaan penyedia layanan yang dalam kegiatannya turut menghimpun data dari masyarakat, terkhusus data-data yang bersifat pribadi atau rahasia. Kekhawatiran tersebut sangat beralasan, bisa dibayangkan apa yang akan terjadi apabila data-data bocor dan disalahgunakan oleh pihak-pihak yang tidak bertanggungjawab. Terlebih di era digital seperti saat ini dimana menjaga keutuhan maupun keamanan data merupakan salah satu tantangan terberat, terkhusus pada sebuah organisasi, data maupun informasi merupakan hal yang penting untuk dijaga kerahasiaannya (Fadlan dkk., 2019).

Dalam kegiatan sehari-hari, sering kali data maupun informasi berpindah dari suatu pihak ke pihak lain seperti mengirimkan data melalui aplikasi pesan digital atau melalui perangkat seperti *flashdisk*. Informasi yang masuk dapat bersifat pribadi maupun rahasia, oleh karena itu keamanan adalah aspek penting yang harus menjadi perhatian dalam dunia teknologi informasi (Kurniawan & Pradana, 2018; Syahroji & Pradana, 2018). Oleh karena itu, dibutuhkan suatu cara atau metode yang tepat untuk menjaga kerahasiaan suatu data maupun informasi. Salah satu cara yang dapat dilakukan adalah dengan menggunakan teknik penyamaran data yang disebut dengan kriptografi.

Kriptografi berasal dari kata "*crypto*" yang berarti rahasia dan "*graphy*" yang berarti tulisan. Jadi, dapat dikatakan bahwa kriptografi adalah tulisan yang tersembunyi. Dengan adanya tulisan yang tersembunyi ini, orang-orang tidak mengetahui bagaimana tulisan tersebut disembunyikan dan tidak mengetahui bagaimana cara membaca maupun menerjemahkan tulisan tersebut. William Stallings mendefinisikan kriptografi sebagai "*The Art and Science of keeping message secure*" (Rohmanu, 2017).

Kriptografi memasuki era barunya yang disebut era kriptografi modern, dimana algoritma-algoritma yang dikembangkan memainkan dan mengolah bit dari pesan

yang hendak dienkripsi. Semakin banyaknya penggunaan komputer digital merupakan salah satu faktor yang mendorong terjadinya perkembangan kriptografi untuk menjaga kerahasiaan informasi digital (Tantoni dkk., 2018).

Fungsi dari kriptografi adalah untuk menyamarkan pesan menjadi pesan yang tersandi (Nasution, 2019). Dengan menerapkan teknik ini pada suatu sistem keamanan perangkat lunak, data akan disamarkan sehingga seandainya data tersebut diperoleh dan dibaca oleh pihak yang tidak bertanggungjawab, informasi yang terkandung didalamnya akan terjaga kerahasiaannya.

Beberapa istilah dalam kriptografi, diantaranya enkripsi, dekripsi, dan kunci. Enkripsi diartikan sebagai proses diubahnya data atau pesan yang hendak dikirim menjadi bentuk yang hampir tidak dikenali oleh pihak ketiga setelah data atau pesan itu sampai kepada penerima, maka penerima melakukan dekripsi yang merupakan kebalikan dari enkripsi. Dekripsi diartikan sebagai proses mengubah data atau pesan kembali kebentuk semula sehingga data atau pesan dapat tersampaikan dan dimengerti oleh penerima, data atau pesan asli dinamakan *plaintext* sedangkan sesudah dikodekan dinamakan *chipertext*. Proses enkripsi dan dekripsi memerlukan kunci dalam mekanismenya dan biasanya berupa *string* atau deretan bilangan (Dwi Putri dkk., 2019).

Kriptografi memiliki beragam metode untuk menyandikan pesan atau informasi yang ingin kita sembunyikan, seperti *Caesar Cipher*, *Affine*, *Monoalphabetic*, *Polyalphabetic*, *Vigenere*, *Transposisi*, dan banyak lagi (Aditya Permana, 2018).

Ragam metode tersebut memiliki rangkaian penyelesaian yang berbeda dengan metode lainnya. Banyak penelitian yang membahas serta mengimplementasikan satu metode dalam mengamankan suatu data. Hal ini tentu dapat dilakukan, namun apabila ingin memberikan pengamanan data yang lebih kuat, maka diperlukan kombinasi dari beberapa metode.

Berdasarkan penelitian tentang kombinasi metode kriptografi yang pernah dilakukan sebelumnya, disimpulkan bahwa dengan mengembangkan atau memodifikasi suatu algoritma kriptografi dapat meningkatkan level keamanan data. Faktor yang menyebabkan kedua algoritma tersebut harus dikombinasikan adalah agar mempersulit kerumitan pemecahan dan kerumitan pencurian teks (Muda Siregar, 2019).

Penelitian lainnya dilakukan oleh (Fadlan dkk., 2019) dalam mengamankan data berupa teks menggunakan Kombinasi dilakukan karena kedua metode tergolong rentan untuk diretas jika diterapkan secara sendiri-sendiri.

Penelitian ini, penulis mengkombinasikan dua algoritma kriptografi, yaitu Affine Cipher dan *Caesar Cipher*, yaitu dengan melakukan dua kali proses enkripsi untuk mengamankan pesan, serta melakukan dua kali proses dekripsi untuk mengembalikan ke pesan semula. Sementara untuk data uji menggunakan data teks.

Affine Cipher merupakan perluasan dari *Caesar Cipher*, yang mengalikan *plaintext* dengan sebuah nilai dan menambahkannya dengan sebuah pergeseran. *Affine Cipher* bukanlah *cipher* yang aman sebab kuncinya dapat ditemukan dengan *exhaustive key search* (Satriana & Karo, 2020).

Sementara *Caesar Cipher* merupakan salah satu jenis cipher substitusi yang membentuk cipher dengan cara melakukan penukaran satu karakter diganti dengan karakter yang berada di sejumlah digit sebelah kanan atau kirinya, tergantung arah pergeserannya (Dwi Putri dkk., 2019).

2. RUANG LINGKUP

Penelitian ini mengusung konsep kombinasi metode (*hybrid*) dari dua metode kriptografi, yaitu algoritma Affine Cipher dan *Caesar Cipher*. Cakupan penelitian ini adalah optimalisasi penggunaan kedua algoritma tersebut dalam mengamankan data berupa teks.

Proses diawali dengan menyamarkan karakter teks murni (*plaintext*) menggunakan metode Affine Cipher terlebih dahulu untuk menghasilkan *ciphertext* yang pertama. Proses selanjutnya, *ciphertext* tersebut bertindak sebagai *plaintext* yang baru untuk disamarkan kembali dengan metode kedua, *Caesar Cipher*, untuk menghasilkan *ciphertext* baru yang merupakan hasil akhir dari proses enkripsi secara menyeluruh.

Sementara proses dekripsinya diawali dengan mengubah *ciphertext* menjadi *plaintext* menggunakan metode *Caesar Cipher*. Selanjutnya *plaintext* tersebut bertindak sebagai *ciphertext* untuk proses dekripsi kedua menggunakan Affine Cipher.

Penelitian ini menggunakan 107 karakter teks yang terdiri dari huruf, angka maupun tanda baca. Sementara untuk kunci menggunakan kombinasi angka.

Hasil penelitian ini adalah penarikan kesimpulan dari hasil uji coba dan analisa yang dilakukan, khususnya terkait dengan pengamanan data teks menggunakan konsep yang telah dirancang.

3. BAHAN DAN METODE

Berikut ini adalah teori-teori pendukung yang menjadi acuan penulis dalam menyusun penelitian ini.

3.1 Kriptografi

Kriptografi (*cryptography*) berasal dari "*crypto*" berarti "*secret*" (rahasia) dan "*graphy*" berarti "*writing*" (tulisan). Jadi, kriptologi adalah ilmu dan seni untuk menjaga keamanan pesan agar tidak diketahui oleh pihak lain. Data atau pesan tersebut diubah menjadi kode-kode yang tidak beraturan sehingga sulit untuk dipahami (Dwi Putri dkk., 2019).

Menurut sisi teknis, kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta

autentikasi data (Dwi dkk., 2018; Dwi Haryono dkk., 2018; Hartanto, 2018).

Algoritma adalah jantung komputer atau informatika (Muda Siregar, 2019). Algoritma pada kriptografi terbagi atas kriptografi klasik dan kriptografi modern (Syarif, 2020). Algoritma klasik pada dasarnya hanya terdiri dari *cipher* subtitusi dan *cipher* transposisi. *Cipher* subtitusi yaitu proses mensubtitusi karakter – karakter yang ada pada *plaintext*. Sedangkan cipher transposisi adalah proses pertukaran huruf – huruf yang terdapat dalam suatu *string* (Prasetyo & Ariyani, 2018).

Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext* tatau teks-jelas (*cleartext*). Pesan dapat berupa data atau informasi yang dikirim atau yang disimpan di dalam media perekaman. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan disandikan ke bentuk lain. Bentuk pesan yang tersandi disebut *ciphertext* atau histogram (*cryptogram*). *Ciphertext* harus dapat ditransformasi kembali menjadi *plaintext*.

3.2 *Affine Cipher*

Metode *Affine Cipher* bekerja dengan mengalikan plainteks dengan sebuah nilai lalu menambahkannya dengan sebuah pergeseran (Rumapea & Sawato Zebua, 2017). Sebagai algoritma *stream cipher*, proses enkripsi dan dekripsi dilakukan per karakter (Nainggolan dkk., 2019).

Teknik yang dilakukan adalah pertukaran karakter secara monoalfabet dimana setiap satu karakter pada *plaintext* akan ditukarkan dengan karakter sesuai dengan rumus yang digunakan (Pangestu & Syahputra, 2020).

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 (Khairani dkk., 2021).

Enkripsi dengan menggunakan Affine Cipher dapat dinyatakan dalam persamaan (1) (Dwi dkk., 2018; Dwi Haryono dkk., 2018; Muda Siregar, 2019; Rumapea & Sawato Zebua, 2017; Satriana & Karo, 2020).

$$C = E_{(a, b, P)} = (aP + b) \bmod m \qquad (1)$$

Proses enkripsi *E* diawali dengan *plaintext P* akan dikalikan dengan sebuah nilai *a* dan ditambahkan dengan sebuah pergeseran *b*. Untuk mendapatkan *ciphertext C*, hasil penjumlahan tersebut di-modulo *m* dimana *m* adalah bilangan bulat yang merujuk pada besaran jumlah karakter yang disiapkan.

Sementara untuk proses dekripsi D, terlebih dahulu perlu diketahui invers dari nilai *a* yaitu a^{-1} yang dinyatakan dalam persamaan (2) (Dwi dkk., 2018; Dwi Haryono dkk., 2018; Muda Siregar, 2019; Rumapea & Sawato Zebua, 2017; Satriana & Karo, 2020).

$$a^{-1};\ a.a^{-1} \bmod m = 1 \qquad (2)$$

Sebagai contoh, diketahui nilai dari *a* adalah 3, dan *m* adalah 26, maka nilai a^{-1} dapat dirunut sebagai berikut :

Tabel 1. Runut Perhitungan Invers

a^{-1}	a	$a.a^{-1}$	$a.a^{-1}$ **mod 26**
1	3	3	3
2	3	6	6
3	3	9	9
.	.	.	.
.	.	.	.
.	.	.	.
9	3	27	**1**

Dari Tabel 1, dapat dilihat bahwa nilai a^{-1} adalah 9. Artinya, apabila hasil dari perkalian modulo adalah 1, maka proses pencarian nilai invers akan terhenti pada nilai invers tersebut.

Selanjutnya, proses dekripsi *D* dilakukan dengan nilai invers a^{-1} dikalikan dengan pengurangan *ciphertext C* terhadap nilai pergeseran *b*. Untuk mendapatkan *plaintext P*, hasil perkalian tersebut di-modulo *m*. Persamaan tersebut dapat dilihat pada persamaan (3) berikut (Dwi dkk., 2018; Dwi Haryono dkk., 2018; Muda Siregar, 2019; Rumapea & Sawato Zebua, 2017; Satriana & Karo, 2020).

$$P = C_{(a^{-1}, b, C)} = a^{-1}(C - b) \bmod m \qquad (3)$$

3.3 *Caesar Cipher*

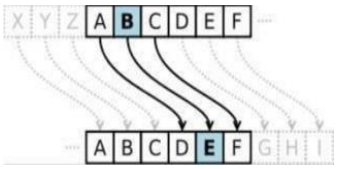
Caesar Cipher adalah konsep awal dari banyak pengembangan metode enkripsi berbasis karakter seperti *Affine Cipher*. Selain *shift cipher*, *Caesar Cipher* juga dikenal dengan istilah *Caesar's code* atau *Caesar shift*.

Berawal dari kekaisaran Romawi yaitu Julius Caesar yang menyandikan pesan yang dikirim kepada para bawahannya, cara kerja algoritma ini yaitu dengan mengganti atau mensubtitusi setiap karakter dengan karakter lain dalam susunan abjad (Susanto & Solichin, 2018).

Caesar Chiper merupakan teknik enkripsi yang paling sederhana dan banyak digunakan. Chiper ini berjenis chiper substitusi, dimana setiap huruf pada *plaintext* digantikan dengan huruf lain yang tetap pada posisi alfabet.

Inti dari algoritma kriptografi *Caesar Cipher* adalah pergeseran, yaitu melakukan pergeseran terhadap semua karakter pada *plaintext* sesuai dengan nilai pergeserannya (Tantoni dkk., 2018).

Misalnya, diketahui bahwa nilai dari sebuah kunci adalah 3, maka huruf "A" akan digantikan oleh huruf "D", huruf "B" menjadi huruf "E", dan seterusnya, untuk proses pergeseseran dapat dilihat pada gambar (Dwi Putri dkk., 2019; Nasution, 2019; Susanto & Solichin, 2018).



Gambar 1. Gambaran *Caesar Cipher*

Gambar 1 merepresentasikan bahwa penyelarasan *plaintext* kedalam bentuk chipertext dilakukan dengan menggeser karakter ke kiri atau kanan sebanyak jumlah kunci yang diinginkan, dalam kasus ini kunci yang digunakan adalah 3.

Penggunaan operator aritmetika modulo pada *Caesar Cipher* dapat dilakukan. Langkah pertama dengan mengidentifikasi karakter yang digunakan, dalam hal ini adalah 26 karakter alfabet. Selanjutnya, masing-masing karakter ditransformasi kedalam bentuk angka, seperti 0 untuk karakter "A", lanjut hingga 25 untuk "Z". Maka, proses enkripsi "E" *Caesar Cipher* dapat dirumuskan (4) (Dwi Putri dkk., 2019; Nasution, 2019; Susanto & Solichin, 2018)

$$C = E_{(k,\,P)} = (P + k)\ mod\ m \qquad (4)$$

Persamaan 4 merupakan persamaan untuk melakukan proses enkripsi dengan ketentuan *P* adalah *plaintext* dijumlahkan dengan besar kunci *k* yang selanjutnya hasil penjumlahan tersebut di-modulo *m*.

Untuk proses dekripsi *D*, dapat dilihat pada formula (5) (Dwi Putri dkk., 2019; Nasution, 2019; Susanto & Solichin, 2018) :

$$P = D_{(k,\,C)} = (C - k)\ mod\ m \qquad (5)$$

Persamaan 5 merupakan persamaan untuk melakukan proses dekripsi dengan ketentuan *C* adalah *ciphertext* dikurangkan dengan besar kunci *k* yang selanjutnya hasil pengurangan tersebut di-modulo *m*.

Kelemahan dari *Caesar Cipher* adalah dapat dipecahkan dengan cara *brute force attack*, suatu bentuk serangan yang dilakukan dengan mencoba-coba berbagai kemungkinan untuk menemukan.

3.4 Daftar Karakter

American Standard Code for Information Interchange (ASCII) atau Kode Standar Amerika untuk Pertukaran Informasi adalah standar pengkodean karakter untuk alat komunikasi. Kode ASCII mewakili teks dalam komputer, peralatan telekomunikasi, dan perangkat lainnya.

Kode ASCII sendiri dapat dikelompokkan lagi ke dalam beberapa bagian, kode yang tidak terlihat simbolnya, kode yang terlihat simbolnya seperti alfabet, dan kode yang tidak ada dikeyboard namun dapat ditampilkan, umumnya untuk kode-kode grafik (Tantoni dkk., 2018).

Sebagai contoh, karakter "A" pada kode ASCII memiliki nilai *hexadecimal* 0041 dan nilai *decimal* 65, tampilan visual karakter "A" pun terlihat sebagaimana mestinya. Terdapat juga karakter dengan nilai hexadecimal 0008 dan nilai decimal 8, namun tampilan visualnya tidak terlihat seperti karakter apapun, melainkan terlihat seperti fungsi pada *keyboard*, karakter ASCII yang dimaksud adalah *backspace*.

Pembuatan daftar karakter perlu ditentukan di tahap persiapan (Fadlan dkk., 2019). Hal ini dilakukan untuk mengantisipasi ketidaktersediaan karakter apabila dilakukan konversi dengan menggunakan ASCII. Daftar karakter tersebut dapat dilihat pada tabel 2.

Tabel 2. Daftar Karakter

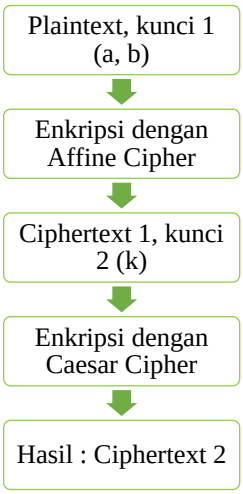
Dec	Char	Dec	Char	Dec	Char
0	0	37	b	74	^
1	1	38	c	75	§
2	2	39	d	76	:
3	3	40	e	77	†
4	4	41	f	78	–
5	5	42	g	79	°
6	6	43	h	80	÷
7	7	44	i	81	$
8	8	45	j	82	.
9	9	46	k	83	spasi
10	A	47	l	84	¢
11	B	48	m	85	…
12	C	49	n	86	=
13	D	50	o	87	!
14	E	51	p	88	«
15	F	52	q	89	‖
16	G	53	r	90	–
17	H	54	s	91	#
18	I	55	t	92	(
19	J	56	u	93	)
20	K	57	v	94	‰
21	L	58	w	95	%
22	M	59	x	96	¶
23	N	60	y	97	+
24	O	61	z	98	?
25	P	62	&	99	»
26	Q	63	<	100	;
27	R	64	>	101	/
28	S	65	*	102	~
29	T	66	@	103	_
30	U	67	\	104	|
31	V	68	{	105	£
32	W	69	}	106	¥
33	X	70	[		
34	Y	71	]		

35	Z	72	¦
36	a	73	•

Tabel 2 berisi daftar karakter yang akan digunakan dalam penelitian ini. Total karakter yang terdapat pada tabel tersebut sebanyak 107 karakter, yang terdiri atas karakter huruf, angka dan simbol-simbol tertentu yang terdapat pada keyboard. Karakter-karakter yang terdapat pada tabel ini lah yang digunakan dalam melakukan proses enkripsi maupun dekripsi pada *Affine Cipher* dan *Caesar Cipher*.

3.5 Proses Enkripsi

Untuk proses enkripsi *plaintext* menggunakan kombinasi metode *Affine Cipher* dan *Caesar Cipher* dapat dilihat pada Gambar 2.

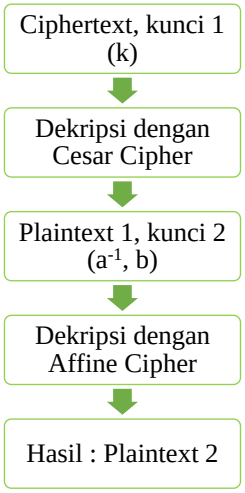


Gambar 2. Proses Enkripsi

Proses enkripsi diawali dengan menyiapkan *plaintext* dan kunci 1 yang terdiri atas nilai *a* dan pergeseran *b*, yang selanjutnya diolah dengan menggunakan *Affine Cipher*. Hasil pengolahan ini akan menghasilkan *ciphertext* 1 dimana akan menjadi *plaintext* untuk proses enkripsi selanjutnya. Selanjutnya diolah kembali dengan menggunakan *Caesar Cipher*. Hasil pengolahan ini akan menghasilkan *ciphertext* 2 dimana ini adalah hasil akhir dari keseluruhan proses enkripsi.

3.6 Proses Dekripsi

Proses dekripsi ciphertext menggunakan kombinasi metode *Affine Cipher* dan *Caesar Cipher* dapat dilihat pada Gambar 3.



Gambar 3. Proses Enkripsi

Proses dekripsi diawali dengan menyiapkan *ciphertext* 1 dan kunci 1 atau kunci *k* yang selanjutnya diolah dengan *Caesar Cipher*. Hasil pengolahan tersebut menghasilkan *plaintext* 1 yang selanjutnya akan menjadi *ciphertext* untuk proses dekripsi selanjutnya. *Plaintext* 1 dan kunci 2 yang terdiri atas invers *a* a^{-1} dan pergeseran *b* diolah kembali dengan menggunakan *Affine Cipher*. Hasil pengolahan ini adalah *plaintext* 2 yang merupakan hasil akhir dari keseluruhan proses dekripsi.

4. PEMBAHASAN

Bagian pembahasan ini dibagi ke dalam beberapa sub-pembahasan, yaitu proses enkripsi, dekripsi dan perancangan aplikasi.

4.1 Proses Enkripsi

Sebagaimana yang tergambarkan pada Gambar 2, proses diawali dengan menyiapkan *plaintext P* dan kunci *a* dan *b*. *Plaintext P* yang dijadikan contoh pada penelitian ini adalah "SEBATIK" dengan kunci *a* dan *b* adalah 3 dan 5. Modulo *m* yang digunakan adalah 107, diambil dari jumlah data karakter pada Tabel 2. Hasil enkripsi tahap pertama ini menggunakan *Affine Cipher* dapat dilihat pada Tabel 3.

Tabel 3. Hasil Enkripsi Affine Cipher

Chr_P	P	a	b	C; (aP+b) mod m	Chr_C
S	28	3	5	89	‖
E	14	3	5	47	l
B	11	3	5	38	c
A	10	3	5	35	Z
T	29	3	5	92	(
I	18	3	5	59	x
K	20	3	5	65	*

Langkah pertama yaitu mengubah karakter *plaintext* ke bentuk angka. Angka ini berdasarkan nomor index

dari daftar karakter pada Tabel 2, seperti karakter "S" berada di index ke-28, "E" di index ke-14, dan seterusnya.

Setelah masing-masing karakter telah diubah ke bentuk indexnya, langkah selanjutnya adalah menerapkan Persamaan 1 untuk mencari nilai enkripsi *C*. Selanjutnya dari hasil perhitungan tersebut, nilai *C* diubah kembali ke bentuk karakter berdasarkan index pada Tabel 2, seperti index 89 untuk karakter "‖", 47 untuk karakter "l", dan seterusnya.

Berdasarkan rangkaian proses di atas, maka hasil enkripsi *plaintext* "SEBATIK" dengan *a* adalah 3, *b* adalah 5 dan *m* adalah 107 menghasilkan *ciphertext* "‖lcZ(x*". Hasil enkripsi ini selanjutnya menjadi karakter baru yang akan dienkripsi kembali menggunakan *Caesar Cipher*.

Merujuk pada Gambar 2, untuk proses enkripsi kedua menggunakan *Caesar Cipher*, diperlukan *ciphertext* 1 dan kunci *k*. Telah diketahui untuk *ciphertext* yang akan diolah adalah "‖lcZ(x*" dan kunci *k* adalah 5. Modulo *m* yang digunakan masih sama, yaitu 107. Langkah selanjutnya adalah menerapkan Persamaan 4 untuk melakukan enkripsi menggunakan *Caesar Cipher* dimana hasil perhitungannya dapat dilihat pada Tabel 4.

Tabel 4. Hasil Enkripsi *Caesar Cipher*

Chr_C	C_1	k	C; (P+k) mod m	Chr_C_2
‖	89	5	94	‰
l	47	5	52	q
c	38	5	43	h
Z	35	5	40	e
(	92	5	97	+
x	59	5	64	>
*	65	5	70	[

Terlihat pada Tabel 4, karakter *ciphertext* diubah kembali menjadi bentuk index sesuai dengan daftar karakter pada Tabel 2. Penerapan persamaan 4 untuk mencari nilai enkripsi *C* dan dilanjutkan dengan mengubah kembali hasil enkripsi tersebut ke bentuk karakter dan menghasilkan *ciphertext* "‰qhe+>[". Hasil enkripsi ini merupakan hasil akhir dari seluruh rangkaian proses enkripsi menggunakan *Affine Cipher* dan *Caesar Cipher* dengan contoh *plaintext* serta kunci yang telah dicontohkan.

4.2 Proses Dekripsi

Proses dekripsi merupakan kebalikan proses enkripsi, dimana data yang diolah adalah *ciphertext* kedua atau hasil akhir dari proses enkripsi menggunakan *Affine Cipher* dan *Caesar Cipher*.

Berdasarkan Gambar 3, proses diawali dengan menyiapkan *ciphertext C* dan kunci *k*. Telah diketahui *ciphertext* C adalah "‰qhe+>[", kunci *k* adalah 5, dan modulo *m* adalah 107. Selanjutnya merujuk pada persamaan 5, dilakukan proses dekripsi pertama menggunakan *Caesar Cipher*. Hasil perhitungan tersebut dapat dilihat pada Tabel 5.

Tabel 5. Hasil Dekripsi *Caesar Cipher*

Chr_C	C	k	P; (C-k) mod m	Chr_C
‰	94	5	89	‖
q	52	5	47	l
h	43	5	38	c
e	40	5	35	Z
+	97	5	92	(
>	64	5	59	x
[	70	5	65	*

Berdasarkan Tabel 5, proses dekripsi pertama menghasilkan karakter "‖lcZ(x*", yang selanjutnya akan dilakukan proses dekripsi kedua menggunakan *Affine Cipher*.

Sebelum masuk pada proses dekripsi kedua, perlu diketahui nilai invers *a* atau a^{-1}. Berdasarkan Persamaan 2, diketahui nilai dari *a* adalah 3, dan modulo *m* adalah 107, maka nilai a^{-1} dapat dirunut sebagai berikut :

Tabel 6. Runut Perhitungan Invers

a^{-1}	a	$a.a^{-1}$	$a.a^{-1}$ mod 107
1	3	3	3
2	3	6	6
3	3	9	9
.	.	.	.
.	.	.	.
.	.	.	.
36	3	108	**1**

Dari Tabel 6, dapat dilihat bahwa nilai a^{-1} adalah 36. Selanjutnya proses dekripsi kedua dapat dilakukan dengan merujuk pada persamaan 3. Hasil perhitungannya dapat dilihat pada Tabel 7.

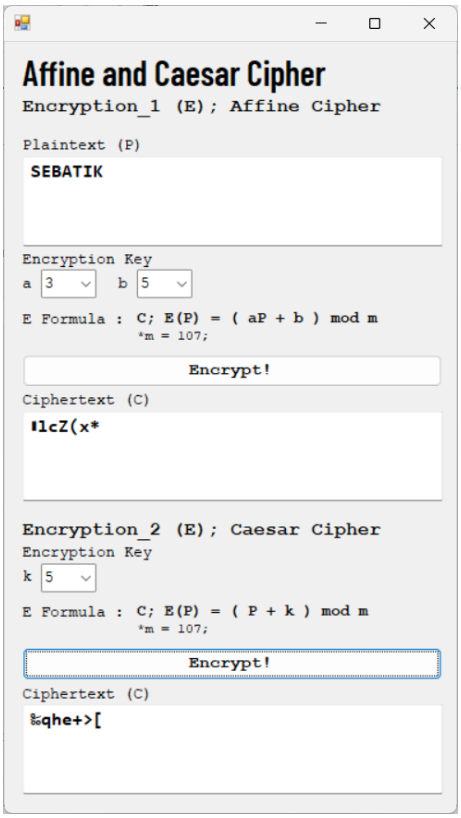
Tabel 7. Hasil Dekripsi *Affine Cipher*

Chr_C	C	a-1	b	P; a-1(C-b) mod m	Chr_P
‖	89	36	5	28	S
l	47	36	5	14	E
c	38	36	5	11	B
Z	35	36	5	10	A
(	92	36	5	29	T
x	59	36	5	18	I
*	65	36	5	20	K

Berdasarkan Tabel 7, proses dekripsi kedua menggunakan *Affine Cipher* menghasilkan karakter *ciphertext* "SEBATIK", dimana karakter ini sama seperti *plaintext* awal pada rangkaian proses enkripsi.

4.3 Perancangan Aplikasi

Perancangan aplikasi bertujuan untuk memberi gambaran penerapan metode *Affine Cipher* dan *Caesar Cipher* pada proses enkripsi dan dekripsi. Peneliti merancang sebuah aplikasi menggunakan Microsoft Visual Studio. Berikut adalah desain antarmuka *form* proses enkripsi.



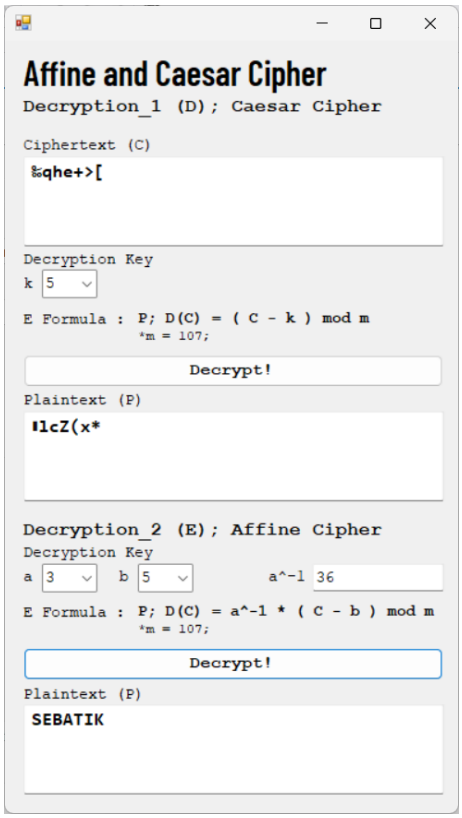
Gambar 4. Desain Antarmuka Form Enkripsi

Desain antarmuka form enkripsi dibangun dengan mengedepankan aspek informatif dan kemudahan pengguna dalam menggunakannya. Pengguna hanya perlu mengetikkan kata atau kalimat pada kolom *plaintext* dilanjutkan dengan mengisi kunci *a* dan *b*. Hasil enkripsi pertama menggunakan *Affine Cipher* akan terlihat pada kolom *ciphertext* yang setelah *button* ”Encrypt!” yang pertama di tekan.

Selanjutnya, *ciphertext* tersebut beserta kunci *k* digunakan untuk proses enkripsi kedua menggunakan *Caesar Cipher*. Sama seperti proses sebelumnya, cipher kedua akan tampil setelah button ”Encrypt!” kedua ditekan.

Proses dekripsi dilakukan dengan langkah-langkah yang hampir sama. Perbedaannya adalah kata atau kalimat *ciphertext* yang digunakan sebagai masukan awal pada kolom *ciphertext*, memasukkan kunci *k*, dilanjutkan dengan menekan button ”Decrypt!”. Proses ini akan menghasilkan *plaintext* yang selanjutnya bertindak selaku *ciphertext* untuk proses dekripsi menggunakan *Affine Cipher*.

Selanjutnya invers *a*, a^{-1} akan diperoleh secara otomatis setelah pengguna memasukkan kunci *a*. A^{-1} inilah yang selanjutnya menjadi kunci untuk proses dekripsi pada *Affine Cipher*. Desain antarmuka form dekripsi dapat dilihat pada



Gambar 5. Desain Antarmuka Form Dekripsi

5. KESIMPULAN

Kombinasi *Affine Cipher* dan *Caesar Cipher* dapat dilakukan. Hasil enkripsi dengan menggunakan *Affine Cipher* dienkripsi kembali menggunakan *Caesar Cipher* untuk meningkatkan keamanan data.

Baik metode *Affine Cipher* dan *Caesar Cipher* sama-sama memiliki kelemahan, yaitu tingkat keamanan yang relatif rendah apabila hanya diimplementasikan secara mandiri tanpa adanya kombinasi dengan metode lain. Kedua metode ini berjenis *cipher* substitusi, dimana setiap huruf pada *plaintext*nya digantikan dengan huruf lain yang tetap pada posisi alphabet, jadi kemungkinan dapat dipecahkan dengan cara *brute force attack*, suatu bentuk serangan yang dilakukan dengan mencoba-coba berbagai kemungkinan untuk menemukan kunci.

Kombinasi kedua metode ini tentu dapat diimplementasikan pada setiap bidang pekerjaan bahkan dapat dikembangkan pada platform lainnya.

6. SARAN

Adapun saran dari penelitian ini adalah menambah jumlah karakter yang digunakan agar hasil enkripsi menjadi lebih sulit untuk dibaca. Selanjutnya penggunaan kunci dalam bentuk karakter juga dapat dilakukan, yaitu dengan mengkonversi karakter tersebut agar bernilai integer terlebih dahulu sebelum diimplementasikan. Selain itu, modifikasi rumus dapat dilakukan agar semakin agar tingkat keamanannya semakin optimal. Terakhir, implementasi kombinasi metode ini dapat diterapkan pada *platform* yang berbeda, seperti *website* maupun aplikasi berbasis selular.

7. DAFTAR PUSTAKA

Aditya Permana, A. (2018). *Penerapan Kriptografi Pada Teks Pesan dengan Menggunakan Metode Vigenere Cipher Berbasis Android* (Vol. 4, Issue 3).

Dwi, B. J., Joko Priono, M., Pengiriman Pesan, P., Suhendri, A., Dwi Juniansyah, B., & Darwis, D. (2018). Implementasi Kombinasi Affine Cipher dan One-Time Pad Dalam. In *Jurnal Informatika* (Vol. 18, Issue 2).

Dwi Haryono, A., Farida Ariyani, P., studi, P., Teknologi Informasi, F., Budi Luhur, U., Raya Ciledug, J., Utara, P., Lama, K., & Selatan, J. (2018). *Aplikasi Pengaman Basis Data pada Nuklindo Lab dengan Algoritma Elgamal dan Affine Cipher* (Vol. 1, Issue MARET).

Dwi Putri, Y., Lutfi, S., Jati Metro, J., & Ternate Selatan, K. (2019). Penerapan Kriptografi Caesar Cipher pada Fitur Chatting Sistem Informasi Freelance. *Jurnal Informatika Dan Komputer) p-ISSN*, *2*(2), 2355–7699. https://doi.org/10.33387/jiko

Fadlan, M., Sinawati, S., Indriani, A., & Bintari, E. D. (2019). Pengamanan Data Teks Melalui Perpaduan Algoritma Beaufort dan Caesar Cipher. *Jurnal Teknik Informatika*, *12*(2), 149–158. https://doi.org/10.15408/jti.v12i2.12262

Hartanto, T. (2018). *Implementasi Web Service Berbasis Rest Menggunakan Algoritma Aes 128 dan Affine Cipher Fitur Bluacademic Aplikasi Blucampus* (Vol. 1, Issue 3).

Khairani, T., Santoso, K. A., & Kamsyakawuni, A. (2021). *PRISMA, Prosiding Seminar Nasional Matematika Pengkodean Monoalphabetic Menggunakan Affine Cipher dengan Kunci Diffie-Hellman*. *4*, 553–559. https://journal.unnes.ac.id/sju/index.php/prisma/

Kurniawan, F., & Pradana, R. (2018). *Algoritma Affine Cipher dan Aes 256 dalam Pengamanan Transfer Data pada Chatting Berbasis Android* (Vol. 1, Issue MARET).

Muda Siregar, I. (2019). Penerapan Algoritma Affine Cipher dan Algoritma Coloumnar Transposition Dalam Keamanan Teks. In *Jurnal Informatika Kaputama (JIK)* (Vol. 3, Issue 1). http://www.stmik-budidama.ac.id,

Nainggolan, I. O., Kementeri, W. M., Balai, P. R. I., Industri, D., & Edan, M. (2019). Implementasi Sandi Affine untuk Pengamanan File Microsoft Office Indra Oloan Nainggolan. In *Jurnal Sains dan Teknologi ISTP* (Vol. 12, Issue 01).

Nasution, A. B. (2019). Implementasi Pengamanan Data dengan Menggunakan Algoritma Caesar Cipher dan Transposisi Cipher. *Jurnal Teknologi Informasi*, *3*(1).

Pangestu, D., & Syahputra, A. (2020). Perancangan Aplikasi Keamanan Cloud Database Menggunakan Operasi XOR dengan Algoritma Affine Berbasis Android Design of Cloud Database Safety Applications Using XOR Operation with Affine Algorithm Based On Android. In *54. IT Journal* (Vol. 8, Issue 1).

Prasetyo, A. I., & Ariyani, F. (2018). *Keamanan Database Nota Penjualan dengan Algoritma Affine Cipher dan Wake Berbasis Web* (Vol. 1, Issue 3).

Rohmanu, A. (2017). Implementasi Kriptografi dan Steganografi dengan Metode Algoritma DES dan Metode End of File. *Jurnal Informatika SIMANTIK*, *2*(1). www.jurnal.stmikcikarang.ac.id

Rumapea, H., & Sawato Zebua, E. (2017). Database pada DBMS MySql Menggunakan Algoritma Affine Cipher Berbasis Java. In *Jurnal METHODIKA* (Vol. 3, Issue 1).

Satriana, E., & Karo, B. (2020). *Penerapan Algoritma Affine Cipher dan Algoritma Electronic Code Book (ECB) dalam Pengamanan Pesan Teks*.

Susanto, I. A., & Solichin, A. (2018). *Enkripsi Data Penggajian dengan Algoritma Caesar Cipher dan Vigenere Cipher pada PT. Kemasindo Cepat Nusantara* (Vol. 1, Issue MARET).

Syahroji, A., & Pradana, R. (2018). *Vigenere Cipher dan Affine Cipher untuk Pengamanan Chatting Berbasis Android* (Vol. 1, Issue 3).

Syarif, A. (2020). Modifikasi Caesar Cipher dengan Permutasi, Transposisi, Binary, Gerbang Logika, ASCII dan Hexa. *Jurnal Teknologi Informasi*, *4*(2).

Tantoni, A., Taufan, M., & Zaen, A. (2018). Implementasi Double Caesar Cipher Menggunakan ASCII. *Jurnal Informatika & Rekayasa Elektronika)*, *1*(2). http://e-journal.stmiklombok.ac.id/index.php/jire