Modifikasi *Cipher* Kriptografi *Caesar* yang Dapat Dibaca dengan Menggunakan Kamus Bahasa Indonesia

Nissa Cahyaningtyas Safitri[1], Agus Prihanto[2],

[1,2] Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya
[1]nissasafitri@mhs.unesa.ac.id, [2] agusprihanto@unesa.ac.id

***Abstrak*— Saat ini kita ketahui bahwa kemajuan komunikasi telah didorong oleh perkembangan teknologi yang primitif hingga teknologi yang lebih maju, namun keamanan, keutuhan serta mutu informasi merupakan hal yang masih perlu kita perhatikan. Tentunya agar informasi tidak jatuh ke tangan orang yang tidak berkepentingan, maka menuntut perlunya diterapkan suatu mekanisme yang baik dalam mengamankan data. Oleh karena itu, dibutuhkan pengamanan agar pesan yang dikirimkan dapat terjaga kerahasiaannya sampai pada penerima. Pada kenyataannya, penggunaan kriptografi sering dapat dipecahkan/diterjemahkan oleh kriptanalis[6]. Saat kita mengetahui pola dan mengetahui konstruksi bentuk untuk memecahkan kriptografi sebenarnya cukup mudah, yaitu cukup memperhatikan susunan huruf dan kata dalam teks yang tampaknya tidak bermakna namun sebenarnya mengandung sebuah pesan tertentu. Tentunya pesan yang nampak tak bermakna tersebut membuat sesesorang curiga bahwa pesan teks tersebut sebenarnya mengandung arti tertentu [13]. Hal ini membuat kriptanalis terpicu untuk memecahkan pesan terenkripsi. Dalam penelitian ini, penulis melakukan modifikasi *Caesar Cipher* yaitu dengan menghasilkan *ciphertext* yang dapat dibaca, sehingga harapannya pesan *ciphertext* tersebut tidak menimbulkan kecurigaan terhadap orang asing yang tidak memiliki hak menerima pesan tersebut. Dengan mengadaptasi metode *Caesar Cipher Monoalfabetik*, penulis melakukan substitusi dari 3 jenis alfabet yaitu: vokal, konsonan dan huruf yang jarang digunakan, maka akan dihasilkan 70 tabel hasil substitusi yang nantinya akan dicocokan dengan Kamus Bahasa Indonesia (KBI), sehingga *ciphertext* yang dihasilkan dapat dibaca dan tidak menimbulkan kecurigaan.**

***Kata Kunci*— Kriptografi, *Caesar Cipher Monoalfabetik*, Kamus Bahasa Indonesia.**

I. PENDAHULUAN

Sejarah kemajuan dalam komunikasi telah dilakukan sejak awal abad ke-19. Dimulai dari telegram awalnya digunakan pada masa Revolusi Industri di Prancis menggunakan kode *semaphore* yaitu kode atau isyarat dari pengibar bendera dengan membedakan posisi peletakan bendera untuk mendapatkan isyarat tertentu [8]. Dewasa ini, kita tahu bahwa kemajuan komunikasi telah didorong oleh perkembangan teknologi yang primitif hingga teknologi yang lebih maju. Hal ini tidak dapat kita pungkiri bahwa sistem komunikasi manusia terus ditentukan dan dibentuk oleh teknologi. [9]. Namun, keamanan informasi merupakan hal yang harus diperhatikan. Upaya menjaga informasi agar tidak jatuh ke tangan orang yang tidak berkepentingan, menuntut perlunya diterapkan suatu mekanisme yang baik dalam mengamankan pesan. Oleh karena itu, dibutuhkan pengaman agar pesan yang dikirimkan dapat terjaga kerahasiaannya sampai pada penerima.

Ilmu yang mempelajari tentang pesan rahasia disebut Kriptografi. Keamanan menyembunyikan pesan atau biasa kita kenal dengan Kriptografi telah banyak dilakukan dengan berbagai metode. Sebagaian besar penelitian menghasilkan *ciphertext* berupa huruf acak, tanda baca acak, atau berupa kode yang tidak kita mengerti seperti penelitian yang baru-baru ini ditemukan berjudul '*An Extended Hybridization of Vigenere and Caesar Cipher Techniques for Secure Communication'* oleh Saraswat, dkk pada tahun 2016 [12]. Penelitian ini memperluas tabel *vigenere* dengan memasukkan data numerik, sehingga angka-angka tersebut juga dapat dienkripsi menggunakan teknik ini. Peneliti menggabungkan proses enkripsi *vigenere* dan *Caesar Cipher* untuk mendapatkan teks *cipher* dari *plaintext* dan kunci yang diberikan. Hasil keluaran berupa huruf dan angka acak seperti ini FW7ORRDFV.

Namun, sebuah penelitian Kriptografi dengan inovasi baru telah dilakukan oleh Benni Purnama dan Hetty Rohaya pada tahun 2015 [10] menghasilkan sebuah *ciphertext* yang manusiawi (dapat dibaca). Mereka menjelaskan dalam jurnalnya alasan yang melatar belakangi penelitian ini agar penerima pesan yang tidak berkepentingan atau kriptanalis tidak merasa curiga sebab pada pesan tersebut tidak terdapat huruf-huruf acak atau simbol-simbol yang menimbulkan kecurigaan, melainkan kata-kata yang terdapat pada Kamus Bahasa Indonesia. Akan tetapi dalam penelitian ini, Benni Purnama dan Hetty Rohaya belum mengimplementasikan penelitian mereka pada suatu aplikasi atau perangkat lunak apapun.

Setelah dilakukan pencarian, metode *Caesar Cipher* telah banyak di implementasikan pada aplikasi berbasis *web* menggunakan bahasa pemrograman PHP. Beberapa contoh jurnal yang mengimplementasikan *Caesar Cipher* berbasis *web* antara lain 'Aplikasi Kriptografi dengan Metode *Vigenere Cipher* Berbasis *Web*' oleh Melati Mawardina dan Entik Insanudin, M.T. Selain itu, adapula 'Perancangan Aplikasi Kriptografi Berbasis *Web* dengan Algoritma *Double Caesar Cipher* Menggunakan Tabel ASCII' oleh Handayani pada tahun 2017. [5]

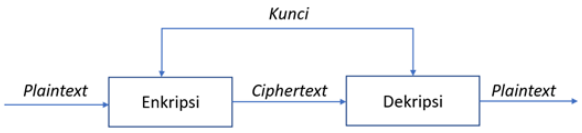
Dalam mendukung keakuratan dalam pengambilan kata Bahasa Indonesia. Penulis memerlukan *database* yang akan dijadikan acuan untuk melakukan pergantian kata yang tidak bermakna menjadi kata yang ada pada kamus Bahasa Indonesia. *Database* ini dikembangkan oleh Fendi Dwi Fauzi secara *open source* dari KBBI-QT. *Database* tersebut adalah salah satu dari sekian banyak *database* yang memiliki Hak Cipta Badan Pengembangan dan Pembinaan Bahasa, Kemdikbud (Pusat Bahasa).

Salah satu penelitian mengangkat algoritma-algoritma deteksi dokumen yang efektif untuk mendeteksi suatu dokumen berbahasa Indonesia atau bukan. Algoritma yang akan dikaji berdasarkan N-gram, dimana algoritma tersebut berhubungan dengan Tri-gram, Bigram dan Unigram. Hasil dari penelitian tersebut berupa diagram kemunculan penggunaan alfabet dalam situs-situs berbahasa Bahasa Indonesia di internet [4]. Dari hasil diagram ini dapat memudahkan peneliti dalam menentukan huruf-huruf yang sering muncul dalam penggunakan kata Bahasa Indonesia.

Dalam bidang komputer yang khususnya melibatkan perhitungan matematis yang berhubungan dengan sistem keamanan, kriptografi menjadi peranan penting dalam menjamin keamanan suatu informasi. Salah satu upaya pengamanan sistem informasi yang dapat dilakukan adalah kriptografi. Kriptografi pada dasarnya merupakan ilmu yang mempelajari teknik matematis yang ditujukan untuk mengamankan suatu sistem informasi. Terdapat 4 aspek dalam mengamankan informasi antara lain keaslian, integritas data, kerahasiaan, serta tidak ada gangguan atau penyangkalan. Keempat aspek tersebut yang menjadi tujuan utama dari suatu sistem kriptografi [7].

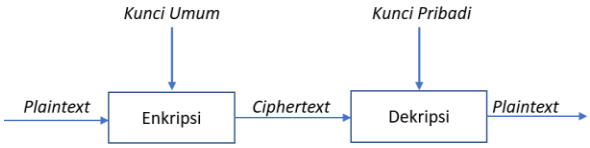
Kunci Kriptografi dibagi 2 macam yakni kriptografi simetris dan kriptografi asimetris. Untuk lebih jelas akan dijelaskan sebagai berikut :

1. Kriptografi Simetris
   Symmetric cryptosystem atau kriptografi simetris adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi. Kriptografi simetris adalah jenis kriptografi yang paling umum digunakan. Kunci untuk pengirim pesan dengan kunci untuk penerima pesan harus sama. Jadi, apabila kunci yang digunakan berbeda maka isi informasi sebenarnya dari pesan tersebut tidak dapat dibuka. Contoh algoritma kunci simetris yang paling sering digunakan adalah RC-4 dan DES *(Data Encryption Standard)* sebagaimana ditunjukkan pada Gambar 1 berikut:



Gbr. 1 Proses Enkripsi menggunakan kunci Simetris

2. Kriptografi Asimetris
   Pada pertengahan tahun 70-an Whitfield Diffie dan Martin Hellman menemukan teknik enkripsi baru yang menggemparkan dunia kriptografi. Enkripsi tersebut menggunakan kunci asimetris, yakni kunci yang berbeda untuk enkripsi dan dekripsi. Kunci publik dapat dimiliki semua orang untuk menjalankan proses enkripsi. Namun, hanya satu orang yang memiliki kunci privat yang dapat membuka informasi pesan sebenarnya. Meskipun tingkat keamanan kriptografi ini lebih baik dibanding kriptografi yang menggunakan kunci simetris, namun waktu yang dibutuhkan untuk proses enkripsi kriptografi asimetris jauh lebih lambat daripada waktu yang dibutuhkan untuk proses enkripsi dengan kunci simetris. Contoh algoritma yang menggunakan kunci asimetris adalah RSA (Rivest, Shamir dan Adleman) [2]. Proses enkripsi dan dekripsi asimetris dapat dilihat pada Gambar 2 dibawah.



Gbr. 2 Proses Enkripsi menggunakan kunci Asimetris

Penelitian ini menggunakan kriptografi simetris yakni kunci yang digunakan untuk proses enkripsi dan dekripsi adalah sama.

Julius Caesar menggunakan *cipher* aditif untuk berkomunikasi dengan perwiranya. Untuk alasan ini, *cipher* aditif kadang-kadang disebut sebagai *Cipher Caesar*. Caesar menggunakan 3 kunci untuk komunikasinya [3].

Dalam Kriptografi, *Caesar cipher* adalah salah satu algoritma enkripsi-dekripsi yang paling dikenal. *Caesar Cipher* adalah jenis subtitusi jenis *cipher* dalam jenis *cipher* ini setiap huruf dalam *plaintext* digantikan oleh huruf beberapa posisi tetap pada alfabet. Enkripsi diwakili menggunakan *modular arithmetic* [11]. Misalnya, dengan pergeseran 5, A akan digantikan oleh F, B akan digantikan oleh G, C akan digantikan oleh K, dan seterusnya. Jadi skema ini seperti yang ditunjukkan pada Gambar 3. Baris pertama menunjukkan huruf asli dan baris kedua menunjukkan apa yang masing-masing alfabet asli akan diganti.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

Gbr. 3 Skema Caesar Cipher Monoalfabetik

Kemudian algoritma dapat diekspresikan sebagai berikut. Untuk setiap huruf *plaintext*, gantilah huruf *ciphertext*:

$$C = E(5, p) = (p + 5) \bmod 26 \qquad (1)$$

Pergeseran mungkin dalam jumlah berapa pun, sehingga algoritma Caesar umum adalah:

$$C = E(k, p) = (p + k) \bmod 26 \qquad (2)$$

Di mana k mengambil nilai dalam rentang 1 hingga 26. Untuk perhitungan Algoritma dekripsi sebagai berikut :

$$p = D(k, C) = (C - k) \bmod 26 \qquad (3)$$

Sebagai contoh pada Gambar 4, pesan yang akan dienkripsi adalah ‘SERANG MUSUH MALAM INI’. Sehingga hasil *ciphertext* dalam contoh ini adalah ‘XJWFSL RZXZM RFQFR NSN’. Dari contoh sederhana ini dapat kita sadari bahwa hasil enkripsi dari metode *Caesar Cipher Monoalfabetik* akan menumbulkan kecurigaan, sebab enkripsi yang dihasilkan berupa huruf-huruf acak dan tidak dapat dibaca.

S	E	R	A	N	G		M	U	S	U	H		M	A	L	A	M		I	N	I
X	J	W	F	S	L		R	Z	X	Z	M		R	F	Q	F	R		N	S	N

Gbr. 4 Simulasi Caesar Cipher Monoalfabetik

Selain itu, jika dalam kasus diketahui bahwa *ciphertext* yang diberikan adalah *Caesar Cipher*, maka pembacaan sandi dengan sistem *brute force* akan lebih mudah yakni dengan cara mencoba seluruh kemungkinan 25 kunci. Ada beberapa titik lemah tentang *Caesar Cipher* yang memungkinkan kita untuk menggunakannya serangan *brute force*. [1] Antara lain sebagai berikut :

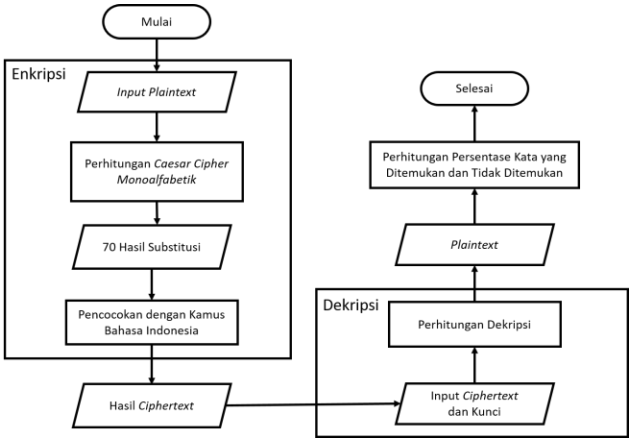
1. Algoritma enkripsi dan dekripsi dikenal.
2. Hanya 25 kunci yang dicoba.
3. Bahasa *plaintext* mudah dikenali

Oleh sebab itu, dibutuhkan modifikasi metode ini agar menghasilkan *output* lebih baik agar tidak mudah dipecahkan.

II. METODOLOGI PENELITIAN

Penelitian yang dilakukan adalah mengenai modifikasi *Caesar Cipher* yang menghasilkan *ciphertext* yang dapat dibaca (manusiawi).

Penelitian ini menggunakan metode algoritma *Caesar Cipher* yang dimodifikasi yakni memperhitungkan frekuensi kemunculan huruf bahasa Indonesia yang nantinya diproses menggunakan aturan baru tabel *vigenere* sehingga dapat menghasilkan sebuah *ciphertext* yang dapat dibaca dan tidak dicurigai bagi orang yang tidak berkepentingan. Diagram alur dari proses penelitian ini dapat dilihat pada Gambar 5 dibawah.



Gbr. 5 Flowchart algoritma Caesar Cipher yang dimodifikasi

A. *Input Plaintext*

Plaintext diinputkan pada ruang obrolan yang digunakan kedua pengguna untuk berkomunikasi. *Input*-an *plaintext* dapat berupa kata maupun kalimat. Kata yang digunakan haruslah menggunakan kata baku, tidak boleh disingkat, atau terdapat salah ketik *(typographical error)*.

B. *Perhitungan Caesar Cipher Monoalfabetik yang Dimodifikasi*

Berdasarkan penelitian 'Deteksi Bahasa untuk Dokumen Teks Bahasa Indonesia' yang dilakukan oleh Amir Hamzah pada tahun 2010 [4], maka ditetapkan beberapa poin aturan substitusi. Hal ini dilakukan agar hasil modifikasi *ciphertext* dapat dibaca (manusiawi). Subtitusi yang dilakukan adalah Monoalfabetik yang artinya setiap huruf memiliki aturanya sendiri. Aturan substitusi dijelaskan sebagai berikut:

1. 5 alfabet vokal (A, I, U, E, O) akan disubstitusi dengan alfabet vokal juga.
2. 14 alfabet konsonan, (B, C, D, F, H, J, K, L, M, P, R, S, T, W) juga diganti dengan alfabet konsonan.
3. 5 alfabet Q, V, X, Y, Z tidak disubstitusi karena penggunaan alfabet dalam bahasa Indonesia jarang digunakan dan untuk menghindari pembentukan hasil enkripsi yang tidak biasa dalam hal penggunaan bahasa Indonesia.
4. 2 Alfabet N dan G tidak mengalami substitusi, sebab kedua huruf tersebut mayoritas penggunaannya berdampingan.

Berdasarkan uraian di atas, dapat dinyatakan bahwa dalam proses metode enkripsi *Cipher Monoalphabetic* melakukan perhitungan substitusi terhadap alfabet yang satu golongan yaitu vokal dengan vokal sebanyak 5 huruf, dan konsonan sebanyak 14 huruf. Sementara 7 huruf lainnya tidak mengalami substitusi.

Pernyataan ini dapat dinotasikan dalam bentuk persamaan sebagai berikut:

$$Cv = (pv + bk) \bmod 5$$
$$Cc = (pc + bk) \bmod 14 \quad (4)$$

Cv = *ciphertext vocal*
Cc = *ciphertext consonant*
pv = *plaintext vocal*
pc = *plaintext consonant*
bk = baris kunci

Dari penjelasan di atas, ini merupakan hasil modifikasi baru dari *Caesar Cipher*, bernama *Cipher Monoalphabetic* yang dapat dilihat pada Gambar 6 dan Gambar 7 dibawah.

		1	2	3	4	5
Plaintext	0	A	I	U	E	O
Ciphertext	1	I	U	E	O	A

Gbr. 6 Substitusi huruf vokal

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Plaintext	0	B	C	D	F	H	J	K	L	M	P	R	S	T	W
Ciphertext	1	C	D	F	H	J	K	L	M	P	R	S	T	W	B

Gbr. 7 Substitusi huruf konsonan

Dari persamaan sebelumnya Cv = (pv + bk) mod5 (4) dan Cc = (pc + bk) mod 14 (5), maka jumlah huruf yang diganti secara keseluruhan dapat diekspresikan dalam persamaan berikut:

$$CVc = \{(pv + bk) + (pc + bk)\}(mod5 \times mod14)$$
$$CVc = \{(pv + bk) + (pc + bk)\} \bmod 70 \quad (5)$$

CVc = *Ciphertext Vocal consonant*
Pv = *Plaintext vocal*
bK = baris kunci
Pc = *Plaintext consonant*

C. *70 Tabel Substitusi*

Berdasarkan persamaan di atas diketahui bahwa proses substitusi pada alfabet terjadi sebanyak 70 kali dengan perincian sebagai berikut:

1. Proses substitusi huruf vokal = 5 x 14 = 70 substitusi
2. Proses substitusi huruf konsonan = 14 x 5 = 70 substitusi
3. Proses subtitusi huruf yang tidak mengalami substitusi, maka ia mengikuti dari jumlah yang ada atau substitusi.

Maka prosesnya dapat dinyatakan 70 x 1 = 70 substitusi.

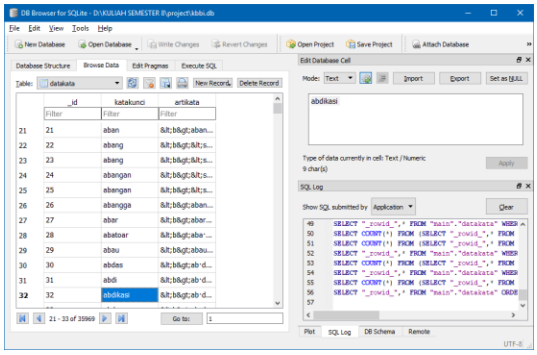
Berdasarkan rincian di atas, proses keseluruhan substitusi *Cipher Monoalfabetik* dapat dilihat pada Tabel I di bawah ini:

TABEL I
HASIL 70 SUBSTITUSI ALFABET

Plaintext		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Ciphertext	0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	1	O	D	F	J	I	H	G	K	U	M	T	S	P	N	E	R	Q	L	C	W	A	V	B	X	Y	Z
	2	E	J	H	M	U	K	G	T	A	P	W	C	R	N	I	L	Q	S	F	B	O	V	D	X	Y	Z
	3	I	M	K	P	A	T	G	W	O	R	B	F	L	N	U	S	Q	C	H	D	E	V	J	X	Y	Z
	4	U	P	T	R	O	W	G	B	E	L	D	H	S	N	A	C	Q	F	K	J	I	V	M	X	Y	Z
	5	A	R	W	L	E	B	G	D	I	S	J	K	C	N	O	F	Q	H	T	M	U	V	P	X	Y	Z
	6	O	L	B	S	I	D	G	J	U	C	M	T	F	N	E	H	Q	K	W	P	A	V	R	X	Y	Z
	7	E	S	D	C	U	J	G	M	A	F	P	W	H	N	I	K	Q	T	B	R	O	V	L	X	Y	Z
	8	I	C	J	F	A	M	G	P	O	H	R	B	K	N	U	T	Q	W	D	L	E	V	S	X	Y	Z
	9	U	F	M	H	O	P	G	R	E	K	L	D	T	N	A	W	Q	B	J	S	I	V	C	X	Y	Z
	10	A	H	P	K	E	R	G	F	I	T	S	J	W	N	O	B	Q	D	M	C	U	V	F	X	Y	Z
	11	O	K	R	T	I	L	G	H	U	W	C	M	B	N	E	D	Q	J	P	F	A	V	H	X	Y	Z
	12	E	T	L	W	U	S	G	K	A	B	F	P	D	N	I	J	Q	M	R	H	O	V	K	X	Y	Z
	13	I	W	S	B	A	C	G	T	O	D	H	R	J	N	U	M	Q	P	L	K	E	V	T	X	Y	Z
	14	U	B	C	D	O	F	G	W	E	J	K	L	M	N	A	P	Q	R	S	T	I	V	W	X	Y	Z
	15	A	D	F	J	E	H	G	B	I	M	T	S	P	N	O	R	Q	L	C	W	U	V	B	X	Y	Z
	16	O	J	H	M	I	K	G	D	U	P	W	C	R	N	E	L	Q	S	F	B	A	V	D	X	Y	Z
	17	E	M	K	P	U	T	G	J	A	R	B	F	L	N	I	S	Q	C	H	D	O	V	J	X	Y	Z
	18	I	P	T	R	A	W	G	M	O	L	D	H	S	N	U	C	Q	F	K	J	E	V	M	X	Y	Z
	19	U	R	W	L	O	B	G	P	E	S	J	K	C	N	A	F	Q	H	T	M	I	V	P	X	Y	Z
	20	A	L	B	S	E	D	G	R	I	C	M	T	F	N	O	H	Q	K	W	P	U	V	R	X	Y	Z
	21	O	S	D	C	I	J	G	L	U	F	P	W	H	N	E	K	Q	T	B	R	A	V	L	X	Y	Z
	22	E	C	J	F	U	M	G	S	A	H	R	B	K	N	I	T	Q	W	D	L	O	V	S	X	Y	Z
	23	I	F	M	H	A	P	G	C	O	K	L	D	T	N	U	W	Q	B	J	S	E	V	C	X	Y	Z
	24	U	H	P	K	O	R	G	F	E	T	S	J	W	N	A	B	Q	D	M	C	I	V	F	X	Y	Z
	25	A	K	R	T	E	L	G	H	I	W	C	M	B	N	O	D	Q	J	P	F	U	V	H	X	Y	Z
	26	O	T	L	W	I	S	G	K	U	B	F	P	D	N	E	J	Q	M	R	H	A	V	K	X	Y	Z
	27	E	W	S	B	U	C	G	T	A	D	H	R	J	N	I	M	Q	P	L	K	O	V	T	X	Y	Z
	28	I	B	C	D	A	F	G	W	O	J	K	L	M	N	U	P	Q	R	S	T	E	V	W	X	Y	Z
	29	U	D	F	J	O	H	G	B	E	M	T	S	P	N	A	R	Q	L	C	W	I	V	B	X	Y	Z
	30	A	J	H	M	E	K	G	D	I	P	W	C	R	N	O	L	Q	S	F	B	U	V	D	X	Y	Z
	31	O	M	K	P	I	T	G	J	U	R	B	F	L	N	E	S	Q	C	H	D	A	V	J	X	Y	Z
	32	E	P	T	R	U	W	G	M	A	L	D	H	S	N	I	C	Q	F	K	J	O	V	M	X	Y	Z
	33	I	R	W	L	A	B	G	P	O	S	J	K	C	N	U	F	Q	H	T	M	E	V	P	X	Y	Z
	34	U	L	B	S	O	D	G	R	E	C	M	T	F	N	A	H	Q	K	W	P	I	V	R	X	Y	Z
	35	A	S	D	C	E	J	G	L	I	F	P	W	H	N	O	K	Q	T	B	R	U	V	L	X	Y	Z
	36	O	C	J	F	I	M	G	S	U	H	R	B	K	N	E	T	Q	W	D	L	A	V	S	X	Y	Z
	37	E	F	M	H	U	P	G	C	A	K	L	D	T	N	I	W	Q	B	J	S	O	V	C	X	Y	Z
	38	I	H	P	K	A	R	G	F	O	T	S	J	W	N	U	B	Q	D	M	C	E	V	F	X	Y	Z
	39	U	K	R	T	O	L	G	H	E	W	C	M	B	N	A	D	Q	J	P	F	I	V	H	X	Y	Z
	40	A	T	L	W	E	S	G	K	I	B	F	P	D	N	O	J	Q	M	R	H	U	V	K	X	Y	Z
	41	O	W	S	B	I	C	G	T	U	D	H	R	J	N	E	M	Q	P	L	K	A	V	T	X	Y	Z
	42	E	B	C	D	U	F	G	W	A	J	K	L	M	N	I	P	Q	R	S	T	O	V	W	X	Y	Z
	43	I	D	F	J	A	H	G	B	O	M	T	S	P	N	U	R	Q	L	C	W	E	V	B	X	Y	Z
	44	U	J	H	M	O	K	G	D	E	P	W	C	R	N	A	L	Q	S	F	B	I	V	D	X	Y	Z
	45	A	M	K	P	E	T	G	J	I	R	B	F	L	N	O	S	Q	C	H	D	U	V	J	X	Y	Z
	46	O	P	T	R	I	W	G	M	U	L	D	H	S	N	E	C	Q	F	K	J	A	V	M	X	Y	Z
	47	E	R	W	L	U	B	G	P	A	S	J	K	C	N	I	F	Q	H	T	M	O	V	P	X	Y	Z
	48	I	L	B	S	A	D	G	R	O	C	M	T	F	N	U	H	Q	K	W	P	E	V	R	X	Y	Z
	49	U	S	D	C	O	J	G	L	E	F	P	W	H	N	A	K	Q	T	B	R	I	V	L	X	Y	Z
	50	A	C	J	F	E	M	G	S	I	H	R	B	K	N	O	T	Q	W	D	L	U	V	S	X	Y	Z
	51	O	F	M	H	I	P	G	C	U	K	L	D	T	N	E	W	Q	B	J	S	A	V	C	X	Y	Z
	52	E	H	P	K	U	R	G	F	A	T	S	J	W	N	I	B	Q	D	M	C	O	V	F	X	Y	Z
	53	I	K	R	T	A	L	G	H	O	W	C	M	B	N	U	D	Q	J	P	F	E	V	H	X	Y	Z
	54	U	T	L	W	O	S	G	K	E	B	F	P	D	N	A	J	Q	M	R	H	I	V	K	X	Y	Z
	55	A	W	S	B	E	C	G	T	I	D	H	R	J	N	O	M	Q	P	L	K	U	V	T	X	Y	Z
	56	O	B	C	D	I	F	G	W	U	J	K	L	M	N	E	P	Q	R	S	T	A	V	W	X	Y	Z
	57	E	D	F	J	U	H	G	B	A	M	T	S	P	N	I	R	Q	L	C	W	O	V	B	X	Y	Z
	58	I	J	H	M	A	K	G	D	O	P	W	C	R	N	U	L	Q	S	F	B	E	V	D	X	Y	Z
	59	U	M	K	P	O	T	G	J	E	R	B	F	L	N	A	S	Q	C	H	D	I	V	J	X	Y	Z
	60	A	P	T	R	E	W	G	M	I	L	D	H	S	N	O	C	Q	F	K	J	U	V	M	X	Y	Z
	61	O	R	W	L	I	B	G	P	U	S	J	K	C	N	E	F	Q	H	T	M	A	V	P	X	Y	Z
	62	E	L	B	S	U	D	G	R	A	C	M	T	F	N	I	H	Q	K	W	P	O	V	R	X	Y	Z
	63	I	S	D	C	A	J	G	L	O	F	P	W	H	N	U	K	Q	T	B	R	E	V	L	X	Y	Z
	64	U	C	J	F	O	M	G	S	E	H	R	B	K	N	A	T	Q	W	D	L	I	V	S	X	Y	Z
	65	A	F	M	H	E	P	G	C	I	K	L	D	T	N	O	W	Q	B	J	S	U	V	C	X	Y	Z
	66	O	H	P	K	I	R	G	F	U	T	S	J	W	N	E	B	Q	D	M	C	A	V	F	X	Y	Z
	67	E	K	R	T	U	L	G	H	A	W	C	M	B	N	I	D	Q	J	P	F	O	V	H	X	Y	Z
	68	I	T	L	W	A	S	G	K	O	B	F	P	D	N	U	J	Q	M	R	H	E	V	K	X	Y	Z
	69	U	W	S	B	O	C	G	T	E	D	H	R	J	N	A	M	Q	P	L	K	I	V	T	X	Y	Z
	70	A	B	C	D	E	F	G	W	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

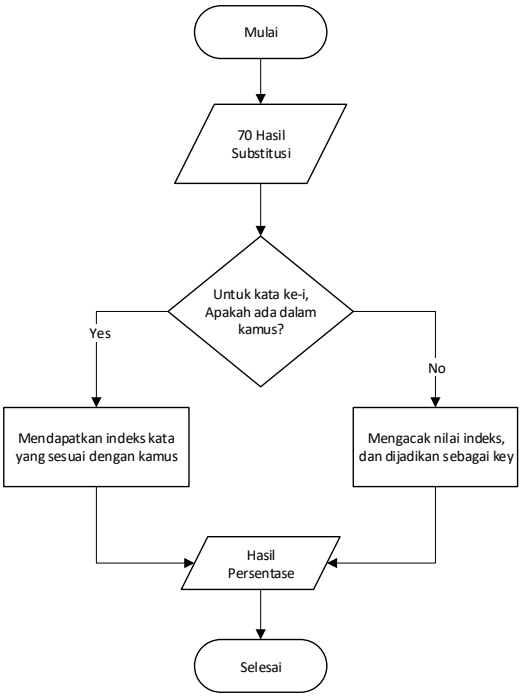
D. Pencocokan dengan Data Kamus

Setelah didapatkan 70 hasil kemungkinan substitusi *ciphertext*, namun belum tentu seluruh kemungkinan tersebut menghasilkan kata-kata yang tersirat atau ada dalam kamus Bahasa Indonesia. Oleh karena itu, diperlukan tahap pencocokan setiap baris yang memiliki kata yang terdapat dalam kamus. Contoh *database* kamus pada SQLite dapat dilihat pada Gambar 8 dibawah.



Gbr. 8 Kamus pada database

Namun, tidak dipungkiri bila pada 70 tabel substitusi terdapat kata yang tidak ditemukan dalam kamus. Oleh sebab itu, diperlukan suatu kondisi untuk menyelesaikan masalah tersebut. Seperti pada Gambar 9, kita akan dihadapkan 2 kemungkinan. Apabila dalam 70 tabel substitusi terdapat kata yang ada pada kamus, maka indeks baris kata tersebut akan dijadikan sebagai kuncinya. Namun, apabila dalam 70 tabel substitusi tersebut tidak terdapat satu katapun yang cocok pada kamus, maka kunci indeks baris akan dipilih secara random. Untuk lebih jelas dapat diperhatikan pada Gambar 9.



Gbr. 9 Flowchart proses pencocokan dengan kamus

E. Hasil Ciphertext

Hasil dari proses pencocokan dengan kamus akan diperoleh kata-kata yang sama seperti pada Kamus, sehingga pesan

ciphertext yang dihasilkan tidak menimbulkan kecurigaan serta dapat dibaca dan bermakna. Gambaran hasil *ciphertext* dapat dilihat pada Tabel II dibawah ini.

TABEL II
KUNCI BARIS YANG TERDAPAT PADA KAMUS

Ciphertext		1	2	3	4	5	6	7	8	9	10	11	12
Plaintext	0	M	U	S	U	H		D	A	T	A	N	G
	1	P	A	C	A	K		J	O	W	O	N	G
	2	R	O	F	O	T		M	E	B	E	N	G
	3	L	E	H	E	W		P	I	D	I	N	G
	4	S	I	K	I	B		R	U	J	U	N	G
	5	C	U	T	U	D		L	A	M	A	N	G
	6	F	A	W	A	J		S	O	P	O	N	G
	7	H	O	B	O	M		C	E	R	E	N	G
	8	K	E	D	E	P		F	I	L	I	N	G
	9	T	I	J	I	R		H	U	S	U	N	G
	10	W	U	M	U	L		K	A	C	A	N	G
	11	B	A	P	A	S		T	O	F	O	N	G
	12	D	O	R	O	C		W	E	H	E	N	G
	13	J	E	L	E	F		B	I	K	I	N	G
	14	M	I	S	I	H		D	U	T	U	N	G
	15	P	U	C	U	K		J	A	W	A	N	G
	16	R	A	F	A	T		M	O	B	O	N	G
	17	L	O	H	O	W		P	E	D	E	N	G

F. *Kunci Baris*

Setelah kita menyimpan kunci indeks baris pada kata yang sesuai dengan Kamus maupun yang diperoleh secara random, maka kunci ini akan digunakan untuk mengembalikan pesan *ciphertext* ke *plaintext*. Jika kunci yang diinputkan nantinya berbeda, maka pesan yang asli *(plaintext)* berupa pesan yang tidak dapat dibaca dan tidak bermakna.

G. *Perhitungan Dekripsi*

Proses dekripsi menggunakan cara yang sama seperti proses enkripsi. Kunci baris yang telah tersimpan pada proses sebelumnya dapat digunakan untuk mencari *plaintext*-nya kembali. Seperti contoh, pada *plaintext* 'MUSUH DATANG' akan menghasilkan *ciphertext* 'PUCUK KACANG', di mana kata 'Pucuk' ada di baris ke-15, dan kata 'Kacang' ada di baris 10. Sehingga kunci baris dalam proses dekripsi dapat dinyatakan bahwa kata 'Pucuk' pada baris -15 menempati posisi 55. Sedangkan kata 'Kacang' pada baris -10 menempati posisi60.

Rumus dekripsi dapat ditulis dalam bentuk persamaan berikut:

$$Pvc = \{(Cv\text{-}bkx1) + (Cc\text{-}bkx1)\} \ mod70 + \{(Cv\text{-}bkx2) + (Cc\text{-}bkx2)\} \ mod70 + \{(Cv\text{-}bkxn) + (Cc\text{-}bkxn)\} \ mod\ 70)\} \quad (6)$$

Sehingga pada kasus dekripsi teks 'MUSUH DATANG' maka diperoleh persamaannya sebagai berikut:

$$Pvc = \{(Cv\text{-}15) + (Cc\text{-}15)\} \ mod70 + \{(Cv\text{-}10) + (Cc\text{-}10)\} \ mod70 \quad (7)$$

Berdasarkan persamaan di atas, dapat ditunjukan gambaran ditemukannya kembali *plaintext* seperti Tabel III dibawah ini agar lebih mudah dimengerti.

TABEL III
KUNCI BARIS PESAN PLAINTEXT

Ciphertext		1	2	3	4	5	6	7	8	9	10	11	12
Plaintext	0	M	U	S	U	H		D	A	T	A	N	G
	1	P	A	C	A	K		J	O	W	O	N	G
	.	R	O	F	O	T		M	E	B	E	N	G
	.	L	E	H	E	W		P	I	D	I	N	G
	.	S	I	K	I	B		R	U	J	U	N	G
	52	B	O	P	O	S		S	E	P	E	N	G
	53	D	E	R	E	C		C	I	R	I	N	G
	54	J	I	L	I	F		F	U	L	U	N	G
	55	M	U	S	U	H		H	A	S	A	N	G
	56	P	A	C	A	K		K	O	C	O	N	G
	57	R	O	F	O	T		T	E	F	E	N	G
	58	L	E	H	E	W		W	I	H	I	N	G
	59	S	I	K	I	B		B	U	K	U	N	G
	60	C	U	T	U	D		D	A	T	A	N	G
	61	F	A	W	A	J		J	O	W	O	N	G
	62	H	O	B	O	M		M	E	B	E	N	G
	63	K	E	D	E	P		P	I	D	I	N	G
	64	T	I	J	I	R		R	U	J	U	N	G
	65	W	U	M	U	L		L	A	M	A	N	G
	66	B	A	P	A	S		S	O	P	O	N	G
	67	D	O	R	O	C		C	E	R	E	N	G
	68	J	E	L	E	F		F	I	L	I	N	G
	69	M	I	S	I	H		H	U	S	U	N	G
	70	P	U	C	U	K		K	A	C	A	N	G

H. *Hasil Plaintext*

Hasil *plaintext* berupa pesan sesungguhnya atau pesan asli.

I. *Hasil Persentase*

Setelah melewati proses pencocokan dengan Kamus Bahasa Indonesia maka akan diketahui jumlah kata yang ditemukan dan tidak ditemukan. Dari *ciphertext* tersebut maka dapat dicari persentase kata-kata yang telah berhasil di enkripsi dengan sempurna. Rumus persentase dapat ditulis sebagai berikut :

$$Hasil\ persentase = \frac{jumlah\ kata\ yang\ ditemukan}{jumlah\ kata\ keseluruhan} \times 100\% \quad (8)$$

III. HASIL DAN PEMBAHASAN

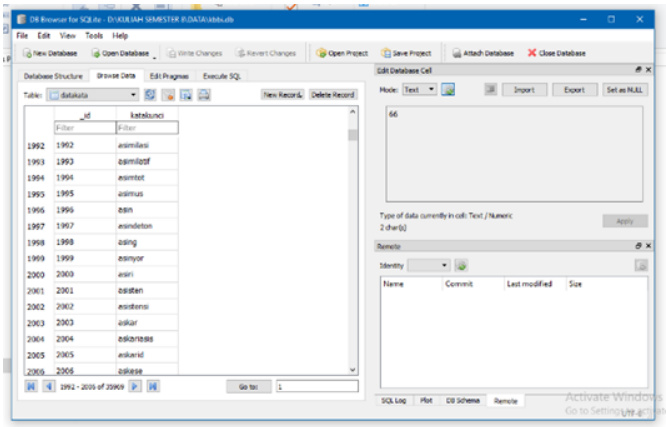
Pengujian modifikasi metode Kriptografi *Caesar Cipher* yakni menghasilkan enkripsi yang dapat dibaca sekaligus pengujian dekripsi untuk mengembalikan ke pesan semula. Implementasi dari penelitian ini dikembangkan berdasarkan hasil analisa dan perancangan yang telah dilakukan peneliti-peneliti sebelumnya. Bab ini akan membahas beberapa tahapan proses sampai pada akhirnya dapat diambil kesimpulan apakah hasil percobaan dari sistem sesuai dengan tujuan yang harapkan.

A. *Data Kamus*

Demi menghasilkan *output* yang maksimal, penelitian perlu didukung dengan data yang berkualitas dan dari sumber yang terpercaya. Dalam proses pencocokan kamus, akan diperlukan

data kata Bahasa Indonesia. Penulis mengambil data kamus Besar Bahasa Indonesia tersebut dari link GitHub berikut https://github.com/efenfauzi/django_kbbi/tree/master/db. Dari data kamus tersebut, penulis menyediakan 2 jenis *database* kamus yakni *database* yang sesuai dengan Kamus Besar Bahasa Indonesia (KBBI) yang berjumlah 35.969 kata, dan *database* kamus yang telah penulis filter sehingga hanya berisi kosa kata yang umum berjumlah 7.212 kata. Dari perbedaan jumlah data kata tersebut dapat kita ketahui yang telah dihilangkan sebanyak 79,95%.

Penulis mengambil data kata Bahasa Indonesia yang telah berupa *Data Base File* yang akan diproses menggunakan SQLite. SQLite merupakan aplikasi open-source alternatif yang cocok digunakan untuk pengolahan data yang tidak begitu kompleks (file tunggal). Selain SQLite juga terdapat MySQL yang juga tergolong *open-source* namun untuk mengatur pengelolahan data aplikasi *web* memerlukan *server* agar tetap bekerja dan menjaga keamanannya. Selain itu diperlukan konfigurasi pengaturan yang tentunya menyita waktu lebih lama. Oleh karena itu, penulis lebih memilih menggunakan SQLite. Tampilan *database* lebih jelas dapat dilihat pada Gambar 10.



Gbr. 10 Tampilan database

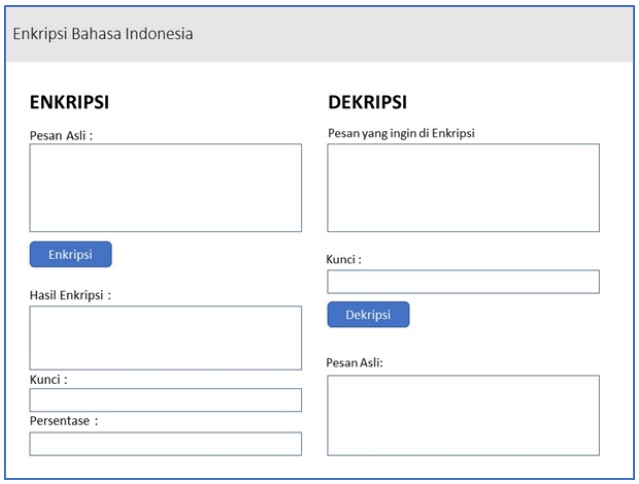
B. *Desain dan Hasil Tampilan*

Berdasarkan kebutuhan sistem, penelitian ini akan diterapkan pada *website*. Dalam 1 *web page* akan dibagi menjadi 2 kolom yakni sisi kiri untuk proses enkripsi dan sisi kanan untuk proses dekripsi. Kolom enkripsi berisi *input* pesan asli yang akan digunakan user untuk memasukan pesan asli yang ingin diubah ke *ciphertext*. Dibawah *input* pesan asli terdapat tombol enkripsi untuk memulai proses enkripsi. Proses tersebut menghasilkan 3 *output* antara lain:

1. Hasil enkripsi yang berupa *ciphertext* yang dapat dibaca dan bermakna.
2. Kunci kata yang telah cocok dengan Kamus maupun yang telah dipilih secara random.
3. Hasil persentase kata yang berhasil ditemukan yang cocok dengan Kamus Bahasa Indonesia.

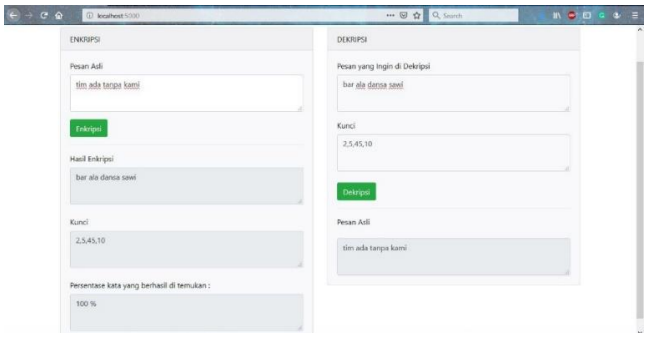
Pada kolom dekripsi berisi *input*-an yang digunakan untuk pesan *ciphertext* yang ingin diubah ke pesan asli *(plaintext)*. Dibawah area inputan terdapat *input*-an kunci yang digunakan untuk mengubah pesan *ciphertext* ke pesan *plaintext*. Jika kunci yang diinputkan benar maka pesan yang asli akan muncul. Begitu juga sebaliknya, jika kunci yang diinputkan tidak benar maka pesan yang dihasilkan tetap tidak dapat dibaca dan bermakna. Setelah kita selesai menginputkan pesan asli dan kunci, *klik button* dekripsi untuk memulai proses dekripsi. Hasil pesan asli *(plaintext)* akan muncul dibawah *button* tersebut.

Rancangan dari tampilan tersebut dapat dilihat pada Lembar Kerja Tampilan (LKT) dari website yang akan dibuat seperti Gambar 11 dibawah ini.



Gbr. 11 Lembar Kerja Tampilan

Sesuai dengan desain Lembar Kerja Tampilan (LKT) diatas, maka hasil dari tampilan yang telah diterapkan pada *website* dapat dilihat pada Gambar 12 dibawah



Gbr. 12 Hasil Tampilan Website

C. *Koneksi Python dan SQLite menggunakan Flask*

SQLite3 adalah bagian dari *Python Standard Library* (Kamus Standar Python). Sebelum melakukan operasi pada *database* SQLite, diperlukan membuat koneksi ke file SQLite. Koneksi yang penulis gunakan yaitu Flask. Flask merupakan kerangka kerja *web* mikro yang ditulis dengan Python. Flask adalah salah satu *web framework* yang berisi *toolkit* untuk aplikasi *Web Server Gateway Interface* (WSGI) dan template untuk bahasa pemrograman Python. Perlu adanya koneksi terbuka ke *database* dengan data yang dimuat sebelumnya dan permintaan yang didefinisikan untuk koneksi tersebut. Proses

ini menggunakan dataset sederhana yang dibuat oleh *Departemen Scientific Computing di Florida State University*.

D. *Hasil Pengujian dan Persentase*

Pengujian menggunakan sampel kata sesuai jurnal acuan dari Betty Purnama dan Hetty Rohayani. Berdasarkan hasil percobaan *website* kriptografi *Caesar Cipher* yang dapat dibaca menggunakan 2 jenis *database* Kamus Bahasa Indonesia, maka didapatkan beberapa kesimpulan sebagai berikut :

1. Hasil *output ciphertext* terdapat 2 jenis sebagai berikut :
   - Jika menggunakan *database* sesuai Kamus Besar Bahasa Indonesia (KBBI), *output* atau hasil enkripsi yang dihasilkan tidak begitu bagus. Meskipun kata-kata yang dihasilkan tersebut telah sesuai perhitungan peraturan Monoalfabetik, namun kata-kata tersebut termasuk tidak umum atau jarang digunakan, maka akan tetap muncul sebagai *output*. Hal ini menyebabkan hasil *output* yang berbeda dengan yang ada di jurnal acuan milik Benni Purnama dan Hetty Rohayani
   - Jika menggunakan *database* kamus Bahasa Indonesia yang telah difilter, maka *output* yang dihasilkan akan lebih baik, sebab kata-kata yang telah masuk perhitungan peraturan Monoalfabetik hanyalah kata-kata yang umum dan sering digunakan. Oleh karena itu, hasil *output* akan lebih mudah dikenal dan dipahami maknanya.

TABEL IV
PERBANDINGAN HASIL ENKRIPSI

No	*Plaintext*	*Ciphertext* berdasarkan Kamus yang difilter	*Ciphertext* berdasarkan KBBI
1	Serang siang tiba	Saring tiang poli (28, 5, 48)	Medang tiang rase (10, 5, 7)
2	Jangan lari pagi	Pengen jadi bagi (2, 10, 10)	Ringin jadi lega (3, 10, 2)
3	Kamu teman jahat	Coba kojun kurus (11, 33, 9)	Luti satin kurus (9, 23, 9)
4	Misi gagal	Diri gagak (40, 5)	Masa gegep (2, 7)
5	Jalan rusak	Lahan paloh (60, 41)	Sukun resik (19, 28)
6	Pulang sekarang	Tebing bupeteng (8, 7)	Tebing wumekeng (8, 8)
7	Misi berhasil	Diri pufbekah (40, 32)	Dara ruhdetak (12, 47)
8	Kami aman sudah	Sawi akan sedih (10, 50, 28)	Sawi acan patos (10, 5, 11)
9	Lari jam satu	Jadi rel tamu (10, 17, 5)	Jadi mop tamu (10, 1, 5)
10	Lima dua	Jiwa hiu (10, 9)	Foli pei (3, 3)

2. Dapat dilihat pada Tabel V bahwa hasil persentase keberhasilan dari proses enkripsi yang menggunakan Kamus Besar Bahasa Indonesia (KBBI) lebih baik daripada hasil persentase keberhasilan proses enkripsi yang menggunakan *database* kamus yang telah difilter. Hal ini disebabkan beragamnya kata-kata yang terdapat pada KBBI membuat proses pencocokan jadi lebih mudah ditemukan.

TABEL V
PERBANDINGAN HASIL PERSENTASE

No	*Plaintext*	*Persentase menggunakan Kamus yang Difilter*	*Persentase menggunakan KBBI*
1	Serang siang tiba	100%	100%
2	Jangan lari pagi	100%	100%
3	Kamu teman jahat	66,67%	100%
4	Misi gagal	100%	100%
5	Jalan rusak	50%	100%
6	Pulang sekarang	50%	50%
7	Misi berhasil	50%	50%
8	Kami aman sudah	100%	100%
9	Lari jam satu	100%	100%
10	Lima dua	100%	100%

IV. KESIMPULAN

Berdasarkan hasil percobaan aplikasi website kriptografi *Caesar Cipher* yang dapat dibaca menggunakan kamus Bahasa Indonesia, maka didapatkan beberapa kesimpulan sebagai berikut:

1. Hasil *output ciphertext* terdapat 2 jenis sebagai berikut :
   - Jika menggunakan *database* sesuai Kamus Besar Bahasa Indonesia (KBBI), *output* atau hasil enkripsi yang dihasilkan tidak begitu bagus. Meskipun kata-kata yang dihasilkan tersebut telah sesuai perhitungan peraturan Monoalfabetik, namun kata-kata tersebut termasuk tidak umum atau jarang digunakan, maka akan tetap muncul sebagai *output*. Hal ini menyebabkan hasil *output* yang berbeda dengan yang ada di jurnal acuan milik Benni Purnama dan Hetty Rohayani
   - Jika menggunakan *database* kamus Bahasa Indonesia yang telah difilter, maka *output* yang dihasilkan akan lebih baik, sebab kata-kata yang telah masuk perhitungan peraturan Monoalfabetik hanyalah kata-kata yang umum dan sering digunakan. Oleh karena itu, hasil *output* akan lebih mudah dikenal dan dipahami maknanya.
2. Hasil persentase keberhasilan dari proses enkripsi yang menggunakan Kamus Besar Bahasa Indonesia (KBBI) lebih baik daripada hasil persentase keberhasilan proses enkripsi yang menggunakan *database* kamus yang telah difilter. Hal ini disebabkan beragamnya kata-kata yang terdapat pada KBBI membuat proses pencocokan jadi lebih mudah ditemukan.

UCAPAN TERIMA KASIH

Puji syukur serta rasa terima kasih saya haturkan kepada Allah SWT yang selalu memberi kemudahan dan kelancaran dalam mengerjakan jurnal ini dan semua pihak terkait yang senantiasa memberi saran dan semangat sehingga jurnal ini dapat terselesaikan dengan baik.

REFERENSI

[1] Anupama, M. (September 2013). Enhancing Security Of Caesar Cipher Using Different Methods. International Journal of Research in Engineering and Technology, eISSN: 2319-1163 | pISSN: 2321-7308 Volume: 02 Issue: 09.

[2] Basri. (2016). Kriptografi Simetris dan Asimetris Dalam Perspektif Keamanan Data Dan Kompleksitas Komputasi. Jurnal Ilmiah Ilmu Komputer, Vol. 2, No. 2.

[3] Forouzan, B. (2010). Cryptography and Network Security. New York: MC Graw-Hill.

[4] Hamzah, A. (2010). Bahasa Untuk Dokumen Teks Berbahasa Indonesia. Seminar Nasional Informatika 2010 (semnasIF 2010), ISSN: 1979-2328.

[5] Handayani, E., Pratitis, W. L., Nur, A., Mashuri, S. A., & Nugroho, B. (2017). Perancangan Aplikasi Kriptografi Berbasis Web Dengan Algoritma Double Caesar Cipher. Seminar Nasional Teknologi Informasi dan Multimedia, ISSN : 2302.

[6] Limbong, T., & Silitonga, P. D. (2017). Testing the Classic Caesar Cipher Cryptography using of Matlab. International Journal of Engineering Research & Technology (IJERT), Vol. 6 Issue 02.

[7] Munir, R. (2011). Kriptografi Keamanan. Bandung: Informatika Bandung.

[8] Nuryanto, H. (2012). Sejarah Perkembangan Teknologi dan Informasi. Jakarta Timur: PT Balai Pustaka (Persero).

[9] Onyejelem , T. E. (2018). From Evolution to Revolution: Exploring the Technology Advances in Mass Communication. International Journal of African and Asian Studies, Vol.45.

[10] Purnama, B., & Rohayani, H. (2015). A New Modified Caesar Cipher Cryptography Method With Legible Ciphertext From A Message To Be Encrypted. International Conference on Computer Science and Computational Intelligence (ICCSCI 2015).

[11] S G Srikantaswamy, D. H. (2012). Improved Caesar Cipher with Random Number Generation Technique and Multistage. International Journal on Cryptography and Information Security (IJCIS)., 39-49.

[12] Saraswat, A., Khatri, C., Sudhakar, Thakral, P., & Biswas, P. (2016). An Extended Hybridization of Vigenere and Caesar Cipher Techniques for Secure Communication. 2nd International Conference on Intelligent Computing, Communication & Convergence (ICCC-2016).

[13] Wicaksono, R. (2010, April 27). Memecahkan Kriptografi dengan Chosen-Plaintext Attack. Retrieved from in Cryptography: http://www.ilmuhacking.com/cryptography/memecahkan-kriptografi-dengan-chosen-plaintext-attack