Jatekom (Jurnal Aplikasi Teknologi dan Komputasi)
Vol. 1, No. 1, Maret 2025, pp. 41~51

Analisis Keamanan Algoritma Beaufort Cipher Berdasarkan Panjang dan Tingkat Keacakan Kunci

Wahyu Suryaning Tyas*, Gilardinho Javiere Oscoraldo Pedrosa Soares, dan Muhammad Fauzi Ardiansyah
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Kota Semarang

Artikel Info

Kata kunci:

Beaufort cipher
Avalanche effect
Bit error rate
Character error rates
Kriptografi

ABSTRAK

Algoritma Beaufort Cipher adalah algoritma substitusi polialfabet yang dikembangkan pada abad ke-19 oleh Francis Beaufort. Algoritma Beaufort Cipher menggunakan kunci dari serangkaian huruf alfabet yang berarti kuncinya terbatas pada jumlah huruf alfabet yaitu sebanyak 26. Setiap karakter plaintext pada algoritma Beaufort Cipher disubstitusi dengan karakter dalam kunci pada posisi yang sama. Penelitian ini bertujuan untuk menguraikan hal-hal apa saja mempengaruhi keamanan teks pada algoritma Beaufort Cipher dengan melakukan percobaan enkripsi dan dekripsi pesan secara berulang-ulang menggunakan panjang teks dan panjang kunci yang berbeda. Berdasarkan percobaan ini, dapat disimpulkan bahwa keamanan algoritma Beaufort Cipher dipengaruhi oleh panjang dan keacakan kunci yang digunakan. Hasil menunjukkan rata-rata persentase Avalanche Effect lebih dari 50%, Character Error Rate sebesar 0%, Bit Error Rate sebesar 0%, dan nilai rata-rata Entropy sebesar 5.

Penulis Korespondensi :

Wahyu Suryaning Tyas,
Program Studi Teknik Informartika
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro, Semarang, 50131
Email: 111202214731@mhs.dinus.ac.id

1. PENDAHULUAN

Kriptografi digunakan oleh orang Yunani kuno sejak sebelum masehi. Salah Satu algoritma yang digunakan adalah Caesar Cipher yang dibuat oleh Julius Caesar pada abad ke-1 SM. Algoritma ini menggunakan kunci yang berbentuk angka. Untuk mengenkripsi, setiap karakter dalam pesan diubah menjadi karakter alfabet dengan menggesernya berdasarkan jumlah kunci [1], [2], [3]. Caesar Cipher [4], [5] adalah algoritma terlemah karena menggunakan pergeseran tetap, artinya setiap karakter dalam teks asli atau plaintext mengikuti pola yang sama. Hal ini memudahkan penyerang untuk mengidentifikasi pola perubahan dan kemudian mendekripsi pesan. Selain itu, Caesar Cipher hanya memiliki 26 kemungkinan kunci, bergantung pada jumlah huruf dalam alfabet [6]. Hal ini memungkinkan penyerang untuk secara sistematis mencoba 26 kemungkinan kunci.

Pada abad ke-16 Blaise de Vigenère mengembangkan algotitma Vigenère Cipher [7]. Dibandingkan dengan Caesar Cipher, Vigenère Cipher lebih aman karena kunci yang digunakan lebih panjang dan menggunakan serangkaian huruf abjad [8], [9], [10], [11], [12]. Namun algoritma ini mudah dipecahkan menggunakan teknik analisis frekuensi jika panjang kuncinya diketahui oleh penyerang.

Vigenère Cipher mulai dianggap tidak aman pada abad ke-19 setelah digunakan secara luas selama berabad-abad [13], [14]. Untuk alasan ini, seorang perwira angkatan laut, Sir Francis Beaufort, mengembangkan algoritma Beaufort Cipher [15], [16], [17]. Algoritma ini hampir sama dengan Vigenère Cipher yaitu menggunakan kunci berupa serangkaian huruf abjad. Namun kunci digunakan secara terbalik untuk mengenkripsi pesan. Tidak seperti algoritma sebelumnya, Beaufort Cipher adalah algoritma yang sederhana untuk diterapkan tanpa memerlukan perangkat khusus. Beaufort Cipher bahkan digunakan oleh

Napoleon Bonaparte selama Perang Napoleon untuk berkomunikasi dengan jendralnya. Terbukti bahwa algoritma ini cukup efektif untuk digunakan dalam komunikasi militer [18].

Algoritma Beaufort Cipher [7], [16] termasuk salah satu cipher paling awal yang penting karena menjadi dasar untuk mengambangkan cipher yang lebih kuat serta menjadi pengantar untuk memahami sejarah kriptografi dan prinsp-prinsipnya. Dengan algoritma yang sederhana, kekuatan Beaufort Cipher masih mampu digunakan untuk aplikasi praktis. Hal-hal tersebut menjadi alasan kami untuk menjadi alasan kami untuk mempelajari Beaufort Cipher lebih dalam. Algoritma Beaufort Cipher merupakan kelanjutan dari algoritma Vigenere Cipher dengan menggunakan kunci yang bersifat transposisi. Algoritma Beaufort Cipher bekerja dengan mengganti setiap karakter dalam plaintext dengan karakter yang terletak pada posisi tertentu dalam alfabet sesuai kuncinya. Saat ini Beaufort Cipher mampu menggunakan 256 karakter (menggunakan seluruh tabel ASCII) padahal awalnya hanya menggunakan 26 karakter. Algoritma Beaufort Cipher bekerja dengan menggunakan kunci yang diulang-ulang untuk mengenkripsi dan mendekripsi pesan [19]. Hal ini dapat membuatnya dengan mudah dipecahkan oleh kriptoanalisis dengan cara memakai metode kasiski. Metode Kasiski adalah modus yang digunakan untuk menyerang cipher substitusi polialfabet. Dengan menggunakan kunci yang diulang-ulang dalam Beaufort Cipher menjadikan metode kasiski bekerja dengan cara mencari pola frekuensi huruf yang berulang pada ciphertext. Namun, modus tersebut tidak berjalan dengan mulus untuk membobol pesan jika sebuah kunci yang digunakan relatif panjang.

Penelitian yang dilakukan oleh Ndururu tahun 2022 [15] menunjukkan bahwa kunci memiliki kedudukan penting dalam mendukung kekuatan algoritma yang digunakan. Sesukar apapun algoritma yang digunakan dalam mengamankan data, akan menjadi mudah diserang bila kunci yang digunakan tidak memiliki struktur yang baik dan kuat terhadap serangan. Menurut penelitian yang dilakukan Ndururu tahun 2021, sebelumnya, keamanan suatu pesan pada metode Beaufort Cipher tidak tergantung pada algoritma yang digunakan dalam menyandikannya, namun tergantung pada kunci yang digunakan. Namun penelitian tersebut tidak menunjukkan secara spesifik tentang pengaruh panjang kunci terhadap keamanan metode Beaufort Cipher. Penelitian ini diharapkan dapat memberikan informasi yang lebih menjangkau tentang keamanan metode Beaufort Cipher secara luas berdasarkan panjang dan acaknya kunci yang digunakan.

2. METODE

2.1 Kriptografi

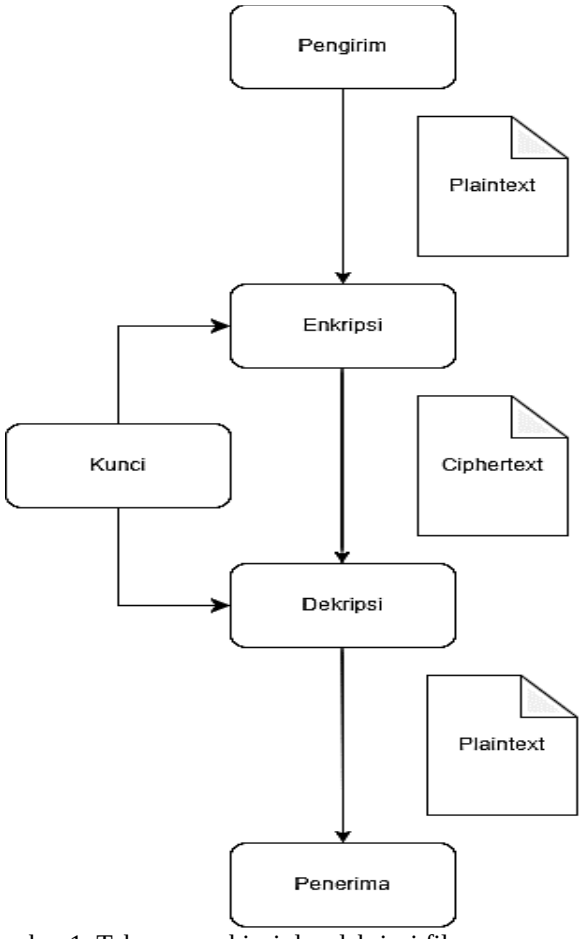
Menurut catatan yang berasal dari masa lampau, kriptografi berasal dari dua kata dalam bahasa Yunani yaitu kryptos yang memiliki makna rahasia dan graphein yang memiliki makna menulis. Secara umum, kriptografi merupakan ilmu serta seni untuk menangani data atau pesan supaya tetap terlindungi [20], [21], [22]. Mengacu pada fakta sejarah saat masa awal muncul kriptografi, kata "seni" memiliki arti setiap orang mempunyai cara yang unik untuk merahasiakan data dan pesan. Data dan pesan yang telah dienkripsi akan menyerupai teks acak yang tidak memiliki arti bagi siapapun yang tidak dituju. Dalam penggunaan teknologi modern, istilah kriptografi berarti implementasi untuk mengamankan pesan atau informasi melalui sejumlah algoritma [23].

Enkripsi merupakan gaya mengubah pesan yang dikirim, diterima, atau disimpan menjadi kode tertentu agar tidak bisa dibaca. Tujuannya untuk mengamankan isi pesan agar tidak dipahami oleh pihak yang tidak memiliki kuasa untuk mengetahuinya [24]. Pada dasarnya, enkripsi mengubah pesan berbentuk teks yang dapat dipahami atau disebut plaintext menjadi bentuk yang tidak dapat dipahami atau disebut ciphertext menggunakan proses dekripsi. Dekripsi merupakan gaya mengubah pesan yang tidak dipahami menjadi bentuk yang dapat dipahami atau dibaca [25], [26]. Tujuannya hampir sama dengan tujuan proses enkripsi karena dua-duanya saling berhubungan, yaitu untuk mengamankan pesan supaya tidak dapat mudah diakses oleh pihak yang tidak bertanggung jawab. Proses dekripsi mengubah ciphertext menjadi bentuk plaintext yang dilakukan ketika pesan telah diterima.

Gambar1 menunjukkan alur ekripsi dan dekripsi plaintext pada Beaufort Cipher. Pertama penetapan kunci, dimana kuncinya berbentuk string alfabet berupa kata atau kumpulan karakter tanpa pola yang tercantum di dalam tabel ascii. Kunci yang tidak memiliki hubungan dengan plaintext dan pola yang bemacam-macam adalah kunci yang dikehendaki karena cukup untuk menghindari pola yang dapat dimanfaatkan untuk memecahkan enkripsi. Kunci dapat dibentuk dengan cara membuatnya secara manual atau mendapatkannya dari orang lain.

Kedua, pengirim mengawali proses dengan membuat pesan yang ingin dikirim. Pesan tersebut dapat berbentuk text, video, gambar, atau data yang lain. Sesudah itu pengirim akan mengenkripsi pesan tersebut memakai algoritma kriptografi. Proses enkripsi dilakukan dengan cara menghitung jumlah bilangan desimal karakter kunci dengan bilangan desimal karakter plaintext mengacu pada tabel ascii. Selanjutnya mengubah bilangan decimal hasil penjumlahan menjadi bentuk karater sesuai dengan tabel ascii. Proses enkripsi akan membentuk ciphertext.

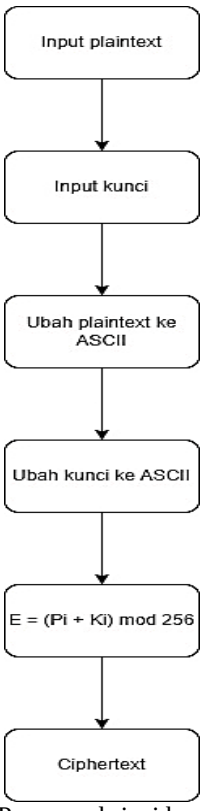
Ketiga, proses dekripsi dilakukan oleh penerima dengan memakai kunci yang sudah didapatkan. Operasi dekripsi mengembalikan ciphertext menjadi plaintext. Caranya mengurangkan bilangan desimal karakter ciphertext dengan bilangan desimal karakter kunci berdasarkan tabel ascii. Ciphertext tidak dapat dibaca jika tidak menggunakan kunci. Proses dekripsi akan mengkonversi ciphertext menjadi plaintext sehingga dapat dibaca oleh penerimanya.



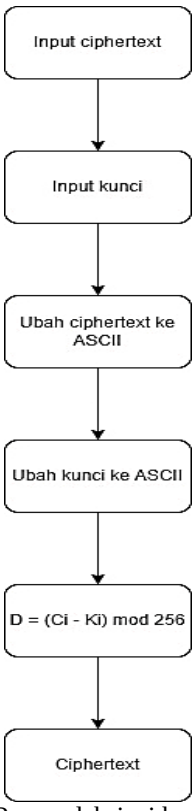
Gambar 1. Tahapan enrkipsi dan dekripsi file secara umum

2.2 Beaufort Cipher

Beaufort Cipher merupakan algoritma kriptografi sederhana yang menggunakan kunci secara berulang-ulang untuk mengenkripsi dan mendekripsi pesan. Kunci yang digunakan dapat berbentuk kata, frasa, ataupun angka dengan panjang sesuai dengan kebutuhan keamanan.



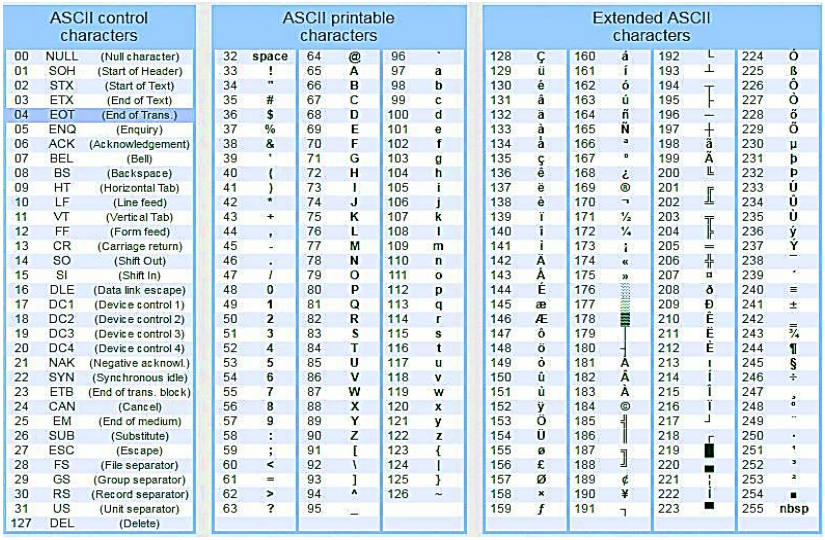
Gambar 2. Proses enkripsi beaufort cipher



Gambar 3. Proses dekripsi beaufort cipher

2.3 ASCII

ASCII merupakan standar pengodean karakter yang digunakan untuk mewakili teks dan karakter-karakter khusus dalam bentuk angka pada komunikasi elektronik. ASCII telah menjadi standar pengodean karakter dominan untuk komunikasi sejak dikembangkan pada tahun 1960-an. Kode ASCII ditetapkan untuk karakter huruf, angka, tanda baca, karakter spasi, dan karakter kontrol dengan menggunakan angka 0 hingga 127 untuk mewakili karakter tersebut.



ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Extended ASCII characters

128	Ç	160	á	192	└	224	Ó
129	ü	161	í	193	┴	225	ß
130	é	162	ó	194	┬	226	Ô
131	â	163	ú	195	├	227	Ò
132	ä	164	ñ	196	─	228	õ
133	à	165	Ñ	197	┼	229	Õ
134	å	166	ª	198	ã	230	µ
135	ç	167	º	199	Ã	231	þ
136	ê	168	¿	200	╚	232	Þ
137	ë	169	®	201	╔	233	Ú
138	è	170	¬	202	╩	234	Û
139	ï	171	½	203	╦	235	Ù
140	î	172	¼	204	╠	236	ý
141	ì	173	¡	205	═	237	Ý
142	Ä	174	«	206	╬	238	¯
143	Å	175	»	207	¤	239	´
144	É	176	░	208	ð	240	≡
145	æ	177	▒	209	Ð	241	±
146	Æ	178	▓	210	Ê	242	‗
147	ô	179	│	211	Ë	243	¾
148	ö	180	┤	212	È	244	¶
149	ò	181	Á	213	ı	245	§
150	û	182	Â	214	Í	246	÷
151	ù	183	À	215	Î	247	¸
152	ÿ	184	©	216	Ï	248	°
153	Ö	185	╣	217	┘	249	¨
154	Ü	186	║	218	┌	250	·
155	ø	187	╗	219	▉	251	¹
156	£	188	╝	220	▄	252	³
157	Ø	189	¢	221	¦	253	²
158	×	190	¥	222	Ì	254	■
159	ƒ	191	┐	223	▀	255	nbsp

Gambar 4. Visualisasi ASCII

2.4 Avalanche Effect (AE)

Dalam kriptografi, avalance effect menjadi tumpuan dalam memastikan seberapa baik sebuah algoritma kriptografi dengan meninjau perubahan bit pada plaintext atau kunci yang menghasilkan perubahan beberapa bit dari ciphertext . Perubahan kecil pada input harus mengakibatkan perubahan yang terlihat secara jelas pada output. Standar avalanche effect terpenuhi ketika perubahan bit menunjukkan hasil 50%, sesuai persamaan (1).

$$\text{Avalance effect} = \frac{Jumlah\ bit\ berbeda}{Total\ bit} x\ 100\% \quad (1)$$

2.5 Bit Error Rate (BER)

Bit error rate adalah perbandingan antara jumlah bit yang salah dengan jumlah bit yang dikirimkan. Semakin rendah nilai presentase BER, berarti sistem kriptografi tersebut semakin baik dalam mengirimkan keaslian data, seperti pada persamaan (2).

$$\text{Bit Error Rate} = \frac{Jumlah\ bit\ yang\ salah}{Jumlah\ total\ bit\ yang\ dikirim} x\ 100\% \quad (2)$$

2.6 Character Error Rate (CER)

Character Error Rate adalah perbandingan antara karakter yang dimuat pada plaintext dengan plaintext yang telah ditambah adau diubah. Semakin rendah nilai presentase CER, berarti sistem kriptografi tersebut semakin baik dalam merahasiakan data, sesuai persamaan (3).

$$\text{Character Error Rate} = \frac{Jumlah\ karakter\ berbeda}{Jumlah\ karakter\ yang\ dikirim} x\ 100\% \quad (3)$$

2.7 Entropy

Entropy adalah tingkat keacakan suatu pesan. Pesan dengan nilai entropy tinggi memiliki arti pesan tersebut lebih aman karena semakin tinggi nilai entropy, maka semakin sulit isi pesan tersebut untuk dibobol, seperti pada persamaan (4).

$$\text{Entropy } = \sum_{i=1}^{n} - Pi\ log2\ Pi \quad (4)$$

3 HASIL DAN PEMBAHASAN

Dari pembahasan diatas akan dilakukan pemeriksaan dengan menggunakan media teks yaitu “Kriptografi adalah kunci keamanan data.” dan “@r!pt0Gr@f! = k3@m@n@n d@t@ t3rperC@y@.” dengan kunci “Kriptografi”. Tahap pertama menyalin karakter pada teks dan karakter pada kunci ke bentuk decimal dengan mencocokannya dengan tabel ASCII. Ulangi sampai karakter paling akhir lalu menghitung jumlah nilai decimal pada pesan dan kunci. Setelah itu salin kembali ke bentuk karakter dengan menyingkronkan dengan tabel ASCII.

Hasil proses enkripsi membuktikan bahwa teks tersandi dengan baik. Selanjutnya teks terenkripsi diubah kembali ke bentuk asal menggunakan kunci yang sama dengan ciphertext “ûΣ╥αΦ▌ ╬Σ┬ ╠╥k ╚═╥α╬╩╧Æ ╠▌╫«▌ë┘ ╩ └╚╧ ╠╫k╓╩Σ ╒” dan “ïΣèαΦƒ«Σí ╠èk»ë▌º» └▒═ª╫k╓⌐Σ┤ Å▌N╚╓£╩╡ ⌐Θ┤”. Tahap pertama menyalin karakter pada teks dan karakter pada kunci ke bentuk decimal dengan menyingkronkannya dengan tabel ASCII. Ulangi hingga karakter terakhir lalu mengurangkan nilai decimal pada pesan dan kunci. Setelah itu menyingkronkan kembali dengan tabel ASCII.

Tabel 1. Proses Enkripsi Teks 1

PLAINTEXT	PLAINTEXT ASCII	KUNCI	KUNCI ASCII	CIPHERTEXT ASCII	CIPHERTEXT
K	75	K	75	150	û
r	114	r	114	228	Σ
i	105	i	105	210	╥
p	112	p	112	224	α
t	116	t	116	232	Φ
o	111	o	111	222	▌
g	103	g	103	206	╬
r	114	r	114	228	Σ
a	97	a	97	194	┬
f	102	f	102	204	╠
i	105	i	105	210	╥
	32	K	75	107	k
a	97	r	114	211	╙
d	100	i	105	205	═
a	97	p	112	209	╤
l	108	t	116	224	α
a	97	o	111	208	╨
h	104	g	103	207	╧
	32	r	114	146	╧
k	107	a	97	204	Æ
u	117	f	102	219	╠
n	110	i	105	215	╫
c	99	K	75	174	«
i	105	r	114	219	«
	32	i	105	137	ë
k	107	p	112	219	ë
e	101	t	116	217	┘
a	97	o	111	208	╨
m	109	g	103	212	╘
a	97	r	114	211	╙
n	110	a	97	207	╧
a	97	f	102	199	╟
n	110	i	105	215	╫
	32	K	75	107	k
d	100	r	114	214	╓
a	97	i	105	202	╩
t	116	p	112	228	Σ
a	97	t	116	213	╒

Tabel 2 Proses Enkripsi Teks 2

PLAINTEXT	PLAINTEXT ASCII	CIPHERTEXT ASCII	CIPHERTEXT
@	64	139	ï
r	114	228	Σ
!	33	138	è
p	112	224	α
t	116	232	Φ
0	48	159	ƒ
G	71	174	«
r	114	228	Σ
@	64	161	í
f	102	204	╠
!	33	138	è
	32	107	k
=	61	175	»
	32	137	ë
k	107	219	▌
3	51	167	º
@	64	175	»
m	109	212	╘
@	64	178	▒
n	110	207	╧
@	64	166	ª
n	110	215	╫
	32	107	k
d	100	214	╓
@	64	169	⌐
t	116	228	Σ
@	64	180	┤
	32	143	Å
t	116	219	▌
3	51	165	Ñ
r	114	211	╙
p	112	214	╓
3	51	156	£
r	114	189	╜
C	67	181	╡
@	64	169	⌐
y	121	233	Θ
@	64	180	┤

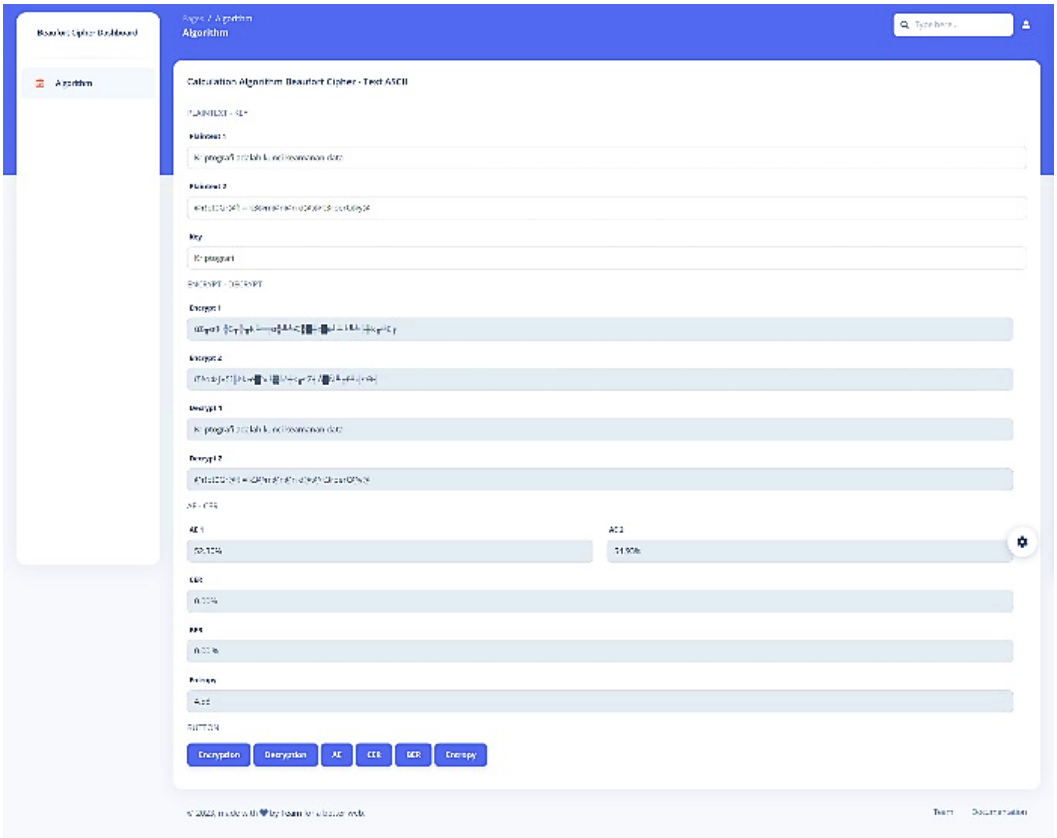
Tabel 3. Proses Dekripsi Teks 1

CIPHERTEXT	CIPHERTEXT ASCII	PLAINTEXT ASCII	PLAINTEXT
û	150	75	K
Σ	228	114	r
╥	210	105	i
α	224	112	p
Φ	232	116	t
▌	222	111	o
╬	206	103	g
Σ	228	114	r
┬	194	97	a
╠	204	102	f
╥	210	105	i
k	107	32	
╙	211	97	a
═	205	100	d
╤	209	97	a
α	224	108	l
╬	208	97	a
╙	207	104	h
╧	146	32	
Æ	204	107	k
╠	219	117	u
▌	215	110	n
╫	174	99	c
«	219	105	i
▌	137	32	
ë	219	107	k
╜	217	101	e
╨	208	97	a
╘	212	109	m
╙	211	97	a
╧	207	110	n
╟	199	97	a
╫	215	110	n
k	107	32	
╥	214	100	d
╨	202	97	a
Σ	228	116	t
╒	213	97	a

Tabel 4. Proses Dekripsi Teks 2

CIPHERTEXT	CIPHERTEXT ASCII	PLAINTEXT ASCII	PLAINTEXT
ï	139	64	@
Σ	228	114	r
è	138	33	!
α	224	112	p
Φ	232	116	t
ƒ	159	48	0
«	174	71	G
Σ	228	114	r
í	161	64	@
╠	204	102	f
è	138	33	!
k	107	32	
»	175	61	=
ë	137	32	
▌	219	107	k
º	167	51	3
»	175	64	@
╘	212	109	m
▒	178	64	@
╧	207	110	n
ª	166	64	@
╫	215	110	n
k	107	32	
╥	214	100	d
⌐	169	64	@
Σ	228	116	t
┤	180	64	@
Â	143	32	
▌	219	116	t
Ñ	165	51	3
╙	211	114	r
╥	214	112	p
£	156	51	3
╜	189	114	r
╡	181	67	C
⌐	169	64	@
Θ	233	121	y
┤	180	64	@

Kesesuaian hasil perhitungan enkripsi dan dekripsi dari dua pesan tersebut dapat dibuktikan dengan membandingkannya dengan hasil enkripsi dan dekripsi dari aplikasi yang ditunjukkan oleh Gambar 5. Pengujian selanjutnya, dilakukan percobaan sebanyak 16 kali diawali dari 8 kata hingga 15 kata pada pesan dan memakai kunci mulai dari 1 kata hingga 4 kata dengan menggunakan campuran antara huruf, karakter, dan angka sesuai Tabel 5.



Gambar 5. Uji coba kriptografi beaufort pada interface berbasis website

Tabel 5. Pengujian AE, BER, CER, dan Entropy

Plainteks	Kunci	Avalance Effect	Bit Error Rate	Character Error Rate	Enthropy
8 kata	1 kata	55.31%	0%	0%	4.72
8 kata	1 kata	55.41%	0%	0%	4.81
8 kata	3 kata	52.40%	0%	0%	5.14
8 kata	4 kata	52.40%	0%	0%	5.17
9 kata	1 kata	51.99%	0%	0%	4.93
9 kata	2 kata	55.28%	0%	0%	5.13
9 kata	3 kata	51.99%	0%	0%	5.13
9 kata	3 kata	51.99%	0%	0%	5.20
12 kata	2 kata	55.57%	0%	0%	5.36
12 kata	4 kata	55.51%	0%	0%	5.35
12 kata	3 kata	52.93%	0%	0%	4.87
15 kata	1 kata	56.78%	0%	0%	5.16
15 kata	1 kata	56.89%	0%	0%	5.23
15 kata	2 kata	56.28%	0%	0%	5.03
15 kata	2 kata	56.34%	0%	0%	5.05
15 kata	3 kata	52.10%	0%	0%	5.36

Hasil pemeriksaan pada Tabel 5 menunjukkan bahwa pengamanan teks menggunakan metode Beufort Cipher membuahkan hasil nilai rata-rata Avalanche Effect lebih dari 50%, nilai rata-rata Bit Error Rate 0%, nilai rata-rata Character Error Rate 0%, dan nilai rata-rata Entropy sebesar 5 yang menunjukkan bahwa metode Beaufort Cipher tersebut dapat mengamankan pesan dengan baik karena algoritma yang baik akan menghasilkan nilai Avalanche Effect yang tinggi dan nilai entropy mendekati 8. Pada hasil percobaan tersebut, semakin panjang plaintext dan kunci semakin besar pula nilai avalanche effectnya. Plaintext yang pendek lebih rawan terhadap serangan avalanche effect. Di sisi lain, plaintext yang acak juga lebih sulit untuk dibobol karena semakin kecil kemungkinan terjadinya avalanche effect yang akan penyerang manfaatkan. Maka disarankan untuk menggunkan plaintext yang panjang dan kunci yang acak untuk meningkatkan keamanan.

Avalanche effect menunjukkan bahwa perubahan plaintext yang kecil dapat menyebabkan perubahan ciphertext yang besar. Sifat tersebut juga menyebabkan plaintext yang acak akan menghasilkan CER yang lebih rendah daripada plaintext yang tidak acak. Pada plaintext yang acak, setiap karakter memiliki kemungkinan yang sama untuk muncul. Hal ini menyebabkan perubahan kecil pada plaintext akan lebih sulit terdeteksi oleh penyerang. Sebaliknya, pada plaintext yang tidak acak, beberapa karakter berkemungkinan lebih banyak muncul daripada karakter lainnya. Perubahan yang kecil pada plaintext ini akan menyebabkan penyerang lebih mudah untuk mendeteksinya.

Selain itu, plaintext yang digunakan juga mempengaruhi nilai BER. Plaintext yang acak menghasilkan BER yang lebih rendah daripada plaintext yang tidak acak. Pola pada plaintext yang tidak acak dapat dimanfaatkan penyerang utnuk menyebbakan kesalahan bit. Plaintext yang lebih rentan terhadap kesalahan bit, antara lain: plaintext yang mengandung banyak huruf yang sama, plaintext yang mengandung banyak pola, dan plaintext yang mengandung banyak data berulang. Sebaliknya, keunikan plaintext yang menghasilkan nilai BER rendah antara lain: plaintext yang acak, plaintext yang mengandung berbagai karakter huruf, angka, dan symbol, serta plaintext yang tidak mengandung data berulang.

Namun perlu untuk dicatat bahwa kemananan plaintext tersebut juga dipengaruhi oleh panjang dan keacakan kunci. Seperti percobaan yang telah dilakukan sebelumnya, semakin panjang kunci yang digunakan, semakin tinggi pula nilai Entropynya. Entropy yang tinggi memiliki arti bahwa data di dalamnya tidak dapat diprediksi dengan mudah. Sebagai contoh, kunci 128bit akan menghasilkan kemungkinan kombinasi ciphertext sebanyak 2128. Hal tersebut membuat ciphertext akan lebih sulit dipecahkan.

3. KESIMPULAN

Berdasarkan hasil percobaan yang telah dilakukan, dapat disimpulkan bahwa keamanan algoritma Beaufort Cipher bergantung pada kunci yang digunakan. Secara khusus, keamanan algoritma Beaufort Cipher dapat dicapai menggunakan kunci yang panjang dan acak. Kunci yang panjang dan acak akan menghasilkan ciphertext yang memiliki nilai Avalanche Effect yang besar, Bit Error Rate yang rendah, Character Error Rate yang rendah, dan Entropy yang tinggi yang membuat pesan di dalamnya semakin sulit untuk dipecahkan. Hal ini disebabkan algoritma Beaufort Cipher menggunakan kunci untuk mengacak plaintext sehingga ciphertext memiliki tingkat keacakan yang tinggi. Perlu untuk diingat bahwa walaupun algoritma Beaufort Cipher dapat menjadi cukup aman dengan kunci yang panjang dan acak, namun algoritma ini tetap dapat rentan terhadap serangan jika kunci yang digunakan terlalu pendek atau jika terdapat pola yang mudah diidentifikasi dalam penggunaan kunci. Oleh karena itu, selain memperhatikan panjang dan keacakan kunci, perlu juga untuk mengubah kunci secara rutin untuk meningkatkan keamanan. Ini menunjukkan prinsip dasar dalam kriptografi di mana keamanan data sangat erat kaitannya dengaan kualitas dan manajemen kunci yang efektif.

REFERENCES

[1] C. A. Sari, D. W. Utomo, and M. A. S. Doheir, "Visual Analysis Based on CMY and RGB Image Cryptography Using Vigenere and Beaufort Cipher," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, May 2023, doi: 10.22219/kinetik.v8i2.1664.

[2] M. Fadlan, Suprianto, Muhammad, and Y. Amaliah, "Double layered text encryption using beaufort and hill cipher techniques," in *2020 5th International Conference on Informatics and Computing, ICIC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020. doi: 10.1109/ICIC50835.2020.9288538.

[3] A. Rachmadsyah, A. Perdana, and A. Budiman, "Kombinasi Algoritma Beaufort Cipher Dan Vigenere Cipher Untuk Pengamanan Pesan Teks BerbasisMobile ApplicationAdryan," *Jurnal Minfo Polgan*, vol. 9, no. 2, pp. 12–17, 2020.
[4] F. Mushtaq Sher Ali and F. Hassan Sarhan, "Enhancing Security of Vigenere Cipher by Stream Cipher," *Int J Comput Appl*, vol. 100, no. 1, pp. 975–8887, 2014.
[5] L. Budi Handoko, "Sekuriti Teks Menggunakan Vigenere Cipher Dan Hill Cipher," *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 19, no. 1, pp. 37–47, 2022.
[6] E. H. Rachmawanto and C. A. Sari, "KEAMANAN FILE MENGGUNAKAN TEKNIK KRIPTOGRAFI SHIFT CIPHER," *Techno.COM*, vol. 14, no. 4, pp. 329–335, 2015.
[7] E. Irfan Riaz Shohab Sandhu *et al.*, "An Enhanced Vigenere Cipher For Data Security," *International Journal Of Scientific & Technology Research*, vol. 5, no. 03, 2016, [Online]. Available: www.ijstr.org
[8] B. B. Ahamed and M. Krishnamoorthy, "SMS Encryption and Decryption Using Modified Vigenere Cipher Algorithm," *Journal of the Operations Research Society of China*, Aug. 2020, doi: 10.1007/s40305-020-00320-x.
[9] N. Laila and A. S. Rms, "Implementasi Steganografi Lsb Dengan Enkripsi Vigenere Cipher Pada Citra," *Computer Science Informatics Journal*, vol. 1, no. 2, 2018.
[10] D. Suprihant *et al.*, "Combination Vigenere Cipher and One Time Pad for Data Security," *International Journal of Engineering & Technology*, vol. 7, no. 2, pp. 92–94, 2018.
[11] E. Rahmawan Pramudya and L. Budi Handoko, "Kriptografi Vigenere Untuk Mengamankan Pesan Teks Berbasis Ocr (Optical Character Recognition)," in *Proceeding SENDIU*, 2021, pp. 460–467.
[12] I. Stepheng, C. A. Sari, E. H. Rachmawanto, and F. O. Isinkaye, "A Combination of Vigenere Cipher and Advanced Encryption Standard for Image Security," *Advance Sustainable Science Engineering and Technology*, vol. 5, no. 3, p. 0230305, Oct. 2023, doi: 10.26877/asset.v5i3.17150.
[13] L. B. Handoko and A. D. Krismawan, "Super Encryption Application Of Cryptography Using Combination Of Columnar Transposition And Vigenere Cipher," in *Seminar Nasional LPPM UMP*, 2020, pp. 534–539.
[14] M. Boussif, N. Aloui, and A. Cherif, "Securing DICOM images by a new encryption algorithm using Arnold transform and Vigenère cipher," *IET Image Process*, vol. 14, no. 6, pp. 1209–1216, May 2020, doi: 10.1049/iet-ipr.2019.0042.
[15] E. Ndruru and T. Zebua, "Pembangkitan Kunci Beaufort Cipher Dengan Teknik Blum-blum Shub pada Pengamanan Citra Digital," *Bulletin of Information Technology (BIT)*, vol. 3, no. 2, pp. 149–154, 2022, doi: 10.47065/bit.v3i1.302.
[16] C. Irawan, E. H. Rachmawanto, C. A. Sari, and C. A. Sugianto, "Super Enkripsi File Dokumen Menggunakan Beaufort Cipher Dan Transposisi Kolom," in *Semnas LPPM UMP*, 2020, pp. 556–563.
[17] C. Irawan, E. H. Rachmawanto, C. A. Sari, and C. A. Sugianto, "SUPER ENKRIPSI FILE DOKUMEN MENGGUNAKAN BEAUFORT CIPHER DAN TRANSPOSISI KOLOM," in *Seminar Nasional LPPM Universitad Muhammadiyah Purwokerto*, 2020, pp. 556–563.
[18] E. Ndruru and T. Zebua, "Generate Beaufort Cipher Key Based on Blum-Blum Shub For Secure Digital Image," *Instal : Jurnal Komputer*, vol. 13, no. 1, 2021.
[19] A. K. Sadasivuni, A. Chandrasekhar, D. Chaya, K. 2#, and S. A. Kumar, "Symmetric Key Cryptosystem For Multiple Encryptions," *International Journal of Mathematics Trends and Technology*, [Online]. Available: http://www.ijmttjournal.org
[20] E. H. Rachmawanto and C. A. Sari, "Keamanan File Menggunakan Teknik Kriptografi Shift Cipher," *Techno.COM*, vol. 14, no. 4, pp. 329–335, 2015.
[21] C. C. Ciptohartono and M. K. Dermawan, "Pencegahan Viktimisasi Pencurian Data Pribadi," *DEVIANCE: JURNAL KRIMINOLOGI*, vol. 3, no. 2, pp. 157–169, 2019.
[22] D. Z. Abidin, "Kejahatan Dalam Teknologi Informasi Dan Komunikasi," *Jurnal Ilmiah Media Processor*, vol. 10, no. 2, 2015, [Online]. Available: www.usdoj.gov/criminal/cybercrimes
[23] C. A. Sari, E. H. Rachmawanto, and E. J. Kusuma, "Good Performance Images Encryption Using Selective Bit T-Des," *Jurnal Ilmu Komputer dan Informasi (Journal of a Science and Information)*, vol. 12, no. 1, pp. 41–49, 2019.
[24] K. Raygaputra Ilaga and C. A. Sari, "Analysis of Secure Image Crypto-Stegano Based on Electronic Code Book and Least Significant Bit," 2018.

[25] C. A. Sari and W. S. Sari, "Kombinasi Least Significant Bit (LSB-1) Dan Rivest Shamir Adleman (RSA) Dalam Kriptografi Citra Warna," *Jurnal Masyarakat Informatika*, vol. 13, no. 1, pp. 45–58, 2022.
[26] C. A. Sari, E. H. Rachmawanto, D. W. Utomo, and R. R. Sani, "Penyembunyian Data Untuk Seluruh Ekstensi File Menggunakan Kriptografi Vernam Cipher dan Bit Shiffting Data Hiding for All Extention File using Cryptography Vernam Cipher and Bit Shiffting," *Journal of Applied Intelligent System*, vol. 1, no. 3, pp. 179–190, 2016.