CGANT JMA
http://www.jcgantunej.or.id
Volume 2, No 1, 2021
DOI:
10.25037/cgantjma.v2i1.50

Journal of
MATHEMATICS AND APPLICATIONS

e-ISSN 2722-7774

Analisa *Antimagicness* Super dari *Shackle* Graf Parasut dan Aplikasinya pada *Polyalphabetic Cipher*

Riza Nurfadila[1,2], Ika Hesti Agustin[1,2], Kusbudiono[1,2]
[1]CGANT - Universitas Jember
[2]Jurusan Matematika FMIPA Universitas Jember
rizanurfadila94@gmail.com, ikahestiagustin@gmail.com, kusbudiono@unej.ac.id

Abstract

Super (*a,d*)-$\mathcal{H}$-antimagic total covering on a graph *G*=(*V,E*) is the total labeling of λ of *V(G)* $\cup$ *E(G)* with positive integers $\{1, 2, 3,\ldots,|V(G) \cup E(G)|\}$, for any subgraph *H'* of *G* that is isomorphic to *H* where $\sum H' = \sum_{v \in V(H)} \lambda(v) + \sum_{e \in E(H)} \lambda(e)$ is an arithmetic sequence {*a, a+d, a+2d,...,a+(s-1)d*} where *a*, *d* are positive numbers where *a* is the first term, *d* is the difference, and *s* is the number of covers. If $\lambda(v)_{v \in V} = 1, 2, 3, \ldots, |V(G)|$ then the graph *G* have the label of super $\mathcal{H}$-antimagic covering. One of the techniques that can be applied to get the super antimagic total covering on the graph is the partition technique. Graph applications that can be developed for super antimagic total covering are *ciphertext* and *streamcipher*. *Ciphertext* is an encrypted message and is related to cryptography. *Stream cipher* is an extension of *Ciphertext*. This article study the super (a,d)-$\mathcal{H}$-antimagic total covering on the shackle of parachute graph and its application in *ciphertext*. The graphs that used in this article are some parachute graphs denoted by *shack*$(\mathcal{P}_m, e, n)$.

Keywords : *super (a,d)-$\mathcal{H}$-antimagic total covering, shackle of parachute graph, Ciphertext, Streamcipher.*
Mathematics Subject Classification: 05C78

Pendahuluan

Graf *G* dideskripsikan sebagai pasangan himpunan (*V,E*), dengan *V* merupakan himpunan titik (*vertex*) dan *E* merupakan himpunan sisi (*edge*) pada graf. Banyaknya titik dan banyaknya sisi yang dimiliki graf *G* berturut-turut disimbolkan dengan $|V(G)|$ dan $|E(G)|$. *E(G)* pada suatu graf boleh kosong sedangkan *V(G)* tidak boleh kosong. Dua titik yang bertetangga pada graf adalah dua titik yang dihubungkan oleh sisi diantaranya. Sebuah titik dikatakan bersisian dengan sebuah sisi jika titik tersebut merupakan titik ujung dari sisi tersebut. Salah satu kajian dalam teori graf adalah pelabelan super (a,d)-$\mathcal{H}$-*antimagic* total selimut. Pelabelan *antimagic* total selimut pada graf *G* dengan himpunan titik (*V(G)*) dan himpunan sisi (*E(G)*) didefinisikan sebagai fungsi bijektif dari titik-titik dan sisi-sisi pada himpunan bilangan bulat 1 sampai sejumlah titik dan sisi, secara matematis dapat ditulis dalam bentuk $f : V(G) \cup E(G) \rightarrow \{1, 2, 3, ..., |V(G)| + |E(G)|\}$ sehingga total label pada setiap selimutnya berbeda dan membentuk barisan aritmatika.

Aplikasi graf mulai berkembang lagi pada pengembangan *ciphertext* dan *stream cipher*. *Ciphertext* merupakan proses pengembangan dari kriptosistem. *Stream cipher* merupakan pengembangan dari *ciphertext*. *Stream cipher* disebut dengan sandi aliran. Salah satu keuntungan dari *stream cipher* adalah tidak dibatasi oleh panjang *plaintext*, sehingga

Received: 01 26 2021; **Revised:** 03 15 2021; **Accepted:** 04 20 2021; **Published:** 06 22 2021

stream cipher cocok untuk digunakan pada enkripsi suatu komunikasi yang berlangsung secara berlanjutan. Graf yang digunakan dalam penelitian ini adalah *shackle* graf parasut. *Shackle* dinotasikan dengan *shack*$(G_1, G_2, G_3, \ldots, G_n)$ yang artinya sebuah graf yang dibentuk dari n terhubung tak trivial $G_1, G_2, G_3, \ldots, G_n$.

Terdapat beberapa hasil penelitian super (a,d)-$\mathcal{H}$-*antimagic* total selimut, antara lain Pengembangan Total Selimut Super pada Graf Shackle Triangular Book [4]. Pelabelan Total Super (a,d)-$\mathcal{H}$-*Antimagic* Total Covering pada Graf Semi Parasut [3]. Pelabelan Total Super (a,d)-$\mathcal{H}$-Antimagic Total Selimut pada Graf Semi Jahangir [2].Lemma yang digunakan untuk batas atas dari super (a,d)-$\mathcal{H}$-antimagic total selimut adalah sebagai berikut.

Lemma 1. *[1] Diberikan G adalah graf yang memiliki titik sebanyak p dan sisi sebanyak q. Jika G adalah super (a,d)-$\mathcal{H}$-*antimagic *total selimut maka $d \leq \frac{(p_G - p_H)p_H + (q_G - q_H)q_H}{n-1}$, untuk $p_G = |V(G)|$, $q_G = |E(G)|$, $p_H = |V(H)|$, $q_H = |E(H)|$, dan n=banyaknya selimut.*

Diketahui bahwa $p_G = np_H - 2n + 2$ and $q_G = nq_H - n + 1$, sehingga kita mempunyai corollary.

Corollary 1. *Untuk $n \geq 2$, jika graf $G = shack(H, e, n)$ memiliki super (a,d)-$\mathcal{H}$-* antimagic *total* selimut *maka*

$$\begin{aligned} d &\leq \frac{(p_G - p_H)p_H + (q_G - q_H)q_H}{n-1} \\ &= \frac{[(np_H - 2n + 2) - p_H]p_H + [(nq_H - n + 1) - q_H]q_H}{(n-1)} \\ &= \frac{np_H^2 - 2np_H + 2p_H - p_H^2 + nq_H^2 - nq_H + q_H - q_H^2}{(n-1)} \\ &= \frac{(n-1)p_H^2 + (n-1)q_H^2 - 2(n-1)p_H - (n-1)q_H}{(n-1)} \\ d &\leq p_H^2 + q_H^2 - 2p_H - q_H \end{aligned}$$

Hasil Penelitian

Penelitian dalam bab ini pelabelan super dari *shackle* graf parasut digunakan untuk membangun *ciphertext* dengan menggunakan metode eliminasi. Metode eliminasi adalah sebuah metode yang dilakukan dengan cara menghapus sisi yang labelnya tidak memenuhi syarat. Metode ini menghasilkan *ciphertext* yang berkorespondensi satu-satu dengan *plaintext* sehingga tidak akan terjadi perubahan makna *plaintext* yang akan didapat dari proses deskripsi. Dalam pembentukan *ciphertext* pada penelitian ini huruf kapital maupun huruf kecil tidak diperhitungkan dan spasi dihiraukan. Super (*a,d*)-$\mathcal{H}$-*antimagic* total selimut pada *shackle* graf parasut ($shack(\mathcal{P}_m, e, n)$) dapat menghasilkan berbagai macam *ciphertext* bergantung pada nilai *n* dan *m*, pelabelan, dan banyaknya keseluruhan karakter yang akan dibentuk *ciphertext*. Disajikan dua contoh *ciphertext* yang dibangun dari super (*a,d*)-$\mathcal{H}$-*antimagic* total selimut pada *shackle* graf parasut, yaitu contoh *ciphertext* dari alfabet ke alfabet dan satu contoh *ciphertext* dari alfabet ke simbol. Selain itu pelabelan super dari *shackle* graf parasut digunakan untuk membangun *stream cipher*. *Stream cipher* merupakan pengembangan dari *ciphertext*. *Stream cipher* disebut dengan sandi aliran. *Stream cipher*

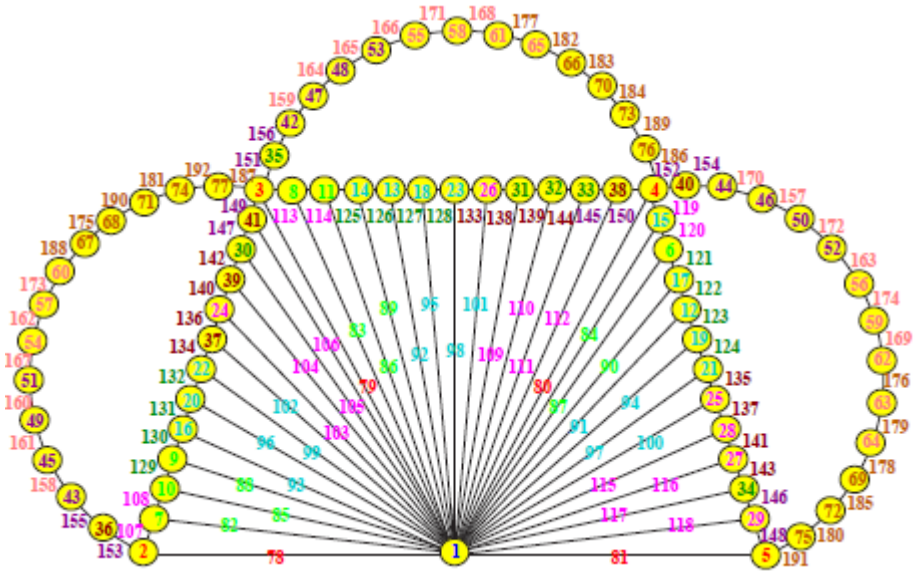

Figure 1: Pelabelan Super (6220,12)-$\mathcal{P}_m$-*Antimagic* Total Selimut

digunakan secara luas di internet dan telepon seluler. Salah satu keuntungan dari *stream cipher* adalah tidak dibatasi oleh panjang *plaintext*, sehingga *stream cipher* cocok untuk digunakan pada enkripsi suatu komunikasi yang berlangsung secara berlanjutan seperti komunikasi melalui telepon. *Stream cipher* juga menggunakan aturan Julius Caesar sehingga *plaintext* $y_1, y_2, y_3, \ldots$ dengan $y_i \in Z_{26}$ dan kunci aliran $k_1, k_2, k_3, \ldots$ dengan $k_i \in Z_{26}$. *Cipher* $x_1, x_2, x_3, \ldots$ diperoleh dengan proses enkripsi $x_i = y_i + k_i(mod\ \ 36)$. Kemudian untuk proses dekripsi dilakukan untuk memperoleh *plaintext* $y_1, y_2, y_3, \ldots$ dengan $y_i = x_i - k_i(mod\ \ 36)$. Berikut pelabelan super (*a,d*)-$\mathcal{H}$-*antimagic* total selimut pada *shackle* graf parasut yang digunakan untuk membangun *ciphertext* dan *stream cipher*.

Pembangunan *Ciphertext* Alfabet dari Super (6220,12)-$\mathcal{P}_m$-*Antimagic* Total Selimut pada *Shackle* Graf Parasut

Dalam pembangunan *ciphertext* alfabet digunakan *shackle* graf parasut sebagai graf kunci dengan super (6220,12)-$\mathcal{H}$-*antimagic* total selimut. Label titik dimulai dari 1 sampai 77 dan pelabelan sisi dimulai dari 78 sampai 192. Pelabelannya dapat dilihat pada Figure 1. Sebagai tahap awal, jumlahkan banyaknya titik dengan banyaknya abjad, yaitu $77 + 26 = 103$. Kemudian eliminasi sisi yang labelnya melebihi 103 agar tidak ada bilangan mod 26 yang berulang. Setelah mengeliminasi sisi, langkah selanjutnya adalah membuat diagram pohon yang berakar dari 1 dan akar selanjutnya mengikuti pola pelabelan graf. Kemudian mencantumkan label sisi pada graf pohon sesuai pelabelan pada graf yang digunakan. Perlu diingat bahwa sisi yang telah dieliminasi tidak perlu dilibatkan dalam diagram pohon. Langkah selanjutnya adalah mencantumkan abjad dari a sampai z pada diagram pohon. Penempatan abjad harus berurutan dari kiri ke kanan dan dimulai dari layer teratas. Sehingga diagram pohon terbentuk seperti Figure 2.

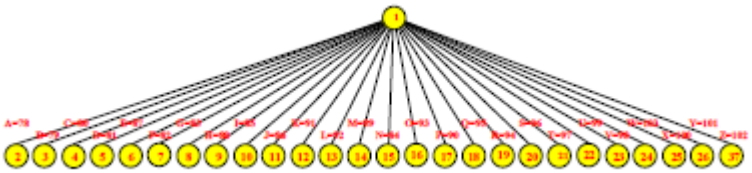

Figure 2: Diagram Pohon

Tahapan terakhir adalah membuat tabel *ciphertext* dari abjad seperti pada Tabel 1.

Table 1: Pembentukan *ciphertext* alfabet dari Figure 1

Plaintext	Label Sisi	(Mod 26)	*Ciphertext*	*Plaintext*	Label Sisi	(Mod 26)	*Ciphertext*
A	78	0	A	N	84	6	G
B	79	1	B	O	93	14	O
C	80	2	C	P	90	11	L
D	81	3	D	Q	95	16	Q
E	87	9	J	R	94	15	P
F	82	4	E	S	96	17	R
G	83	5	F	T	97	18	S
H	88	10	K	U	99	20	U
I	85	7	H	V	98	19	T
J	86	8	I	W	103	24	Y
K	91	12	M	X	100	21	V
L	92	13	N	Y	101	22	W
M	89	25	Z	Z	102	23	X

Tabel ini dibuat berdasarkan aturan Julius Caesar. Sebagai contoh, *plaintext* ”Maaf, nomor yang anda tuju sedang tidak aktif atau berada di luar jangkauan. Silahkan hubungi beberapa saat lagi!” akan dirubah menjadi *ciphertext*. Langkah pertama adalah menghapus tanda baca dan spasi, serta hiraukan huruf kapital sehingga pesan menjadi ”maafnomoryanga ndatujusedangtidakaktifatauberadadiluarjangkauansilahkanhubungi beberapasaatlagi”. Dengan menggunakan Tabel 1 lakukan proses enkripsi sehingga didapatkan *ciphertext* ”zaaeg ozopwagfagdasuiurjdagfshdamamsheasaubjpadadhnuapiagfmauagrhnakmagkubugfhbjbjpa laraasnafh”.

Pembangunan *Ciphertext* Simbol dari Super (6220,12)-$\mathcal{P}_m$-*Antimagic* Total Selimut pada *Shackle* Graf Parasut

Dalam pembangunan *ciphertext* simbol digunakan *shackle* graf parasut sebagai graf kunci dengan super (6220,12)-$\mathcal{H}$-*antimagic* total selimut. Label titik dimulai dari 1 sampai 77 dan pelabelan sisi dimulai dari 78 sampai 192. Pelabelannya dapat dilihat pada Figure 1. Langkah-langkah dalam pembangunan *ciphertext* simbol sama dengan pembangunan *ciphertext* alfabet, namun terdapat perbedaan pada tahap pengeliminasian. Karena banyaknya

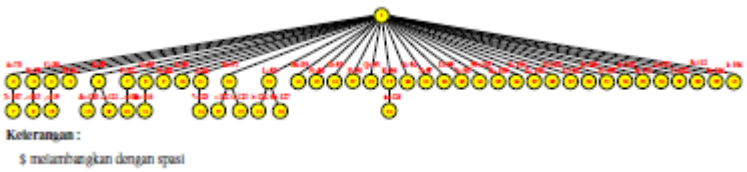

Figure 3: Diagram Pohon

titik dan banyaknya karakter berturut-turut adalah 77 dan 50 maka $|V(G)| + \alpha = 127$ sehingga sisi yang labelnya melebihi 127 dieliminasi agar tidak ada bilangan (mod 50) yang berulang. Sebelum membuat tabel, dibuat terlebih dahulu aturan pengkodean yang ditetapkan karena banyak karakter pada pembangunan *ciphertext* simbol lebih dari 26 sehingga tidak dapat menggunakan aturan Julius Caesar. Aturan tersebut tercantum pada Tabel 2. Tahap terakhir adalah membuat tabel *ciphertext* dari alfabet seperti pada Tabel 3.

Sehingga contoh *plaintext* ”Maaf, nomor yang anda tuju sedang tidak aktif atau berada di luar jangkauan. Silahkan hubungi beberapa saat lagi!” dapat dirubah menjadi *ciphertext* tanpa menghapus tanda baca dan spasi tidak dihilangkan. Namun pada proses pembangunan *ciphertext* simbol, penggunaan huruf kapital tetap diabaikan sehingga pesan menjadi ”maaf, nomor yang anda tuju sedang tidak aktif atau berada di luar jangkauan. silahkan hubungi beberapa saat lagi!”. Dengan menggunakan Tabel 3 lakukan proses enkripsi sehingga didapatkan *ciphertext* ”$\partial\aleph\aleph\wp¥\vartheta\nabla\triangle\partial\triangle\neg\vartheta\sharp\aleph\nabla\Re\vartheta\aleph\nabla-\jmath\aleph\vartheta\top\perp\emptyset\perp\vartheta\surd\ell\jmath\aleph\nabla\Re\vartheta\top\prime\jmath\aleph\angle\vartheta\aleph\angle\top\prime\wp\vartheta\aleph\top\aleph\perp\vartheta\hbar\ell\neg\aleph\jmath\aleph\vartheta$
$\prime\infty\aleph\Im\angle\aleph\nabla\vartheta\Im\perp\hbar\perp\nabla\Re\prime\vartheta\hbar\ell\hbar\ell\neg\aleph\forall\aleph\vartheta\surd\aleph\aleph\top\vartheta\infty\aleph\Re\prime\checkmark$”.

Table 2: Pembentukan *ciphertext* simbol dari Gambar 4.3.

Plaintext	Label Sisi	(Mod 26)	*Ciphertext*	*Plaintext*	Label Sisi	(Mod 26)	*Ciphertext*
A	78	0	$\aleph$	Z	102	23	$\|\|$
B	79	1	$\hbar$	0	115	37	♣
C	80	2	$\imath$	1	118	40	♢
D	81	3	$\jmath$	2	105	27	♡
E	87	9	ℓ	3	109	31	♠
F	82	4	$\wp$	4	110	32	†
G	83	5	$\Re$	5	111	33	‡
H	88	10	$\Im$	6	117	39	§
I	85	7	$\prime$	7	102	24	¶
J	86	8	$\emptyset$	8	112	34	©
K	91	12	$\angle$	9	104	26	£
L	92	13	∞	!	106	28	✓
M	89	25	∂	?	107	29	✠
N	84	6	∇	.	113	35	®
O	93	14	$\triangle$	,	119	41	¥
P	90	11	$\forall$	&	120	42	α
Q	95	16	$\exists$	+	121	43	β
R	94	15	$\neg$	-	108	30	γ
S	96	17	$\surd$	:	114	36	δ
T	97	18	$\top$	*	125	47	ϵ
U	99	20	$\perp$	=	122	44	ε
V	98	19	$\backslash$	(	123	45	ζ
W	103	24	$\flat$	)	126	48	η
X	100	21	$\natural$	@	127	49	θ
Y	101	22	$\sharp$	$	124	46	ϑ

Table 3: Aturan Pengkodean *Ciphertext* Simbol dari Gambar 4.3.

Mod(50)	0	1	2	3	4	5	6	7	8	9
Ciphertext	$\aleph$	$\hbar$	$\imath$	$\jmath$	ℓ	$\wp$	$\Re$	$\Im$	$\prime$	$\emptyset$
Mod(50)	10	11	12	13	14	15	16	17	18	19
Ciphertext	$\angle$	∞	∂	∇	$\triangle$	$\forall$	$\exists$	$\neg$	$\surd$	$\top$
Mod(50)	20	21	22	23	24	25	26	27	28	29
Ciphertext	$\perp$	$\setminus$	$\flat$	$\natural$	$\sharp$	$\parallel$	♣	♢	♡	♠
Mod(50)	30	31	32	33	34	35	36	37	38	39
Ciphertext	†	‡	§	¶	©	£	✓	✠	®	¥
Mod(50)	40	41	42	43	44	45	46	47	48	49
Ciphertext	α	β	γ	δ	ϵ	ε	ζ	η	θ	ϑ

Pembangunan *Stream Cipher* Simbol dari Super (6220,12)-$\mathcal{P}_m$-*Antimagic* Total Selimut pada *Shackle* Graf Parasut

Dalam pembangunan *stream cipher* digunakan *shackle* graf parasut sebagai graf kunci dengan super (6220,12)-$\mathcal{H}$-*antimagic* total selimut. Label titik dimulai dari 1 sampai 77 dan pelabelan sisi dimulai dari 78 sampai 192. Pelabelannya dapat dilihat pada Figure 1. Langkah-langkah dalam pembangunan *stream cipher* juga menggunakan aturan Julius Caesar sehingga *plaintext* $y_1, y_2, y_3, \ldots$ dengan $y_i \in Z_{26}$ dan kunci aliran $k_1, k_2, k_3, \ldots$ dengan $k_i \in Z_{26}$. *Cipher* $x_1, x_2, x_3, \ldots$ diperoleh dengan proses enkripsi $x_i = y_i + k_i(mod\ 36)$. Kemudian untuk proses dekripsi dilakukan untuk memperoleh *plaintext* $y_1, y_2, y_3, \ldots$ dengan $y_i = x_i - k_i(mod\ 36)$. Sebuah *plaintext* "Maaf, nomor yang anda tuju sedang tidak aktif atau berada di luar jangkauan. Silahkan hubungi beberapa saat lagi!" akan diubah menjadi *ciphertext* dengan mengabaikan huruf kapital, spasi, dan tanda baca lainya menjadi "maafnomor yangandatujusedangtidakaktifatauberadadiluar-jangkauansilahkanhubungibeberapasaatlagi". Dengan menggunakan pelabelan super (6220,12) pada *shackle* graf parasut seperti pada figure 1 maka dapat dilakukan proses enkripsi seperti pada tabel 4 sehingga diperoleh *ciphertext* "ncdmxxcinjyakpxutqtlzatttj-ftjhhupcogeaucxfdrbfejsxltoicskvglggahv fnldrpfphfdigkiqaqkwtdgcyoyd". Setelah dila-kukan proses enkripsi oleh pengirim pesan, penerima pesan harus melakukan proses dekripsi agar pesan yang telah dikirimkan dapat dibaca. *Plaintext* $y_1, y_2, y_3, \ldots$ dapat diperoleh dari proses dekripsi $y_i = x_i - k_i(mod\ 26)$. Dari tabel 4.5 dapat dilihat proses dekripsi. Sehingga *ciphertext* yang terlah dibangun yaitu "ncdmxxcinjyakpxutqtlzatttj-ftjhhupcogeaucxfdrbfejsxltoicskvglggahvfnldrpfphfdigkiqaqk wtdgcyoyd" dapat diubah - kembali menjadi *plaintext* "maafnomor yangandatujusedangtidakaktifatauberadadiluarjangkauansilahkanhubungibeberapasaatlagi". Apabila *plaintext* tersebut dilengkapi de-ngan huruf kapital, spasi, dan tanda baca maka didapatkan *plaintext* "Maaf, nomor yang anda tuju sedang tidak aktif atau berada di luar jangkauan. Silahkan hubungi beberapa saat lagi!".

Table 4: Proses Enkripsi

Plaintext	y_i	k_i	$x_i = y_i + k_i(mod\ 26)$	*Ciphertext*
m	12	1	13	n
a	0	2	2	c
a	0	3	3	d
f	5	7	12	m
n	13	10	23	x
o	14	9	23	x
m	12	16	2	c
o	14	20	8	i
r	17	22	13	n
y	24	37	9	j
a	0	24	24	y
n	13	39	0	a
g	6	30	10	k
a	0	41	15	p
n	13	36	23	x
d	3	43	20	u
a	0	45	19	t
t	19	49	16	q
u	20	51	19	t
j	9	54	11	l
u	20	57	25	z
s	18	60	0	a
e	4	67	19	t
d	3	68	19	t
a	0	71	19	t
n	13	74	9	j
g	6	77	5	f
t	19	78	19	t
i	8	79	9	j
d	3	82	7	h
a	0	85	7	h
k	10	88	20	u

Plaintext	y_i	k_i	$x_i = y_i + k_i(mod\ 26)$	*Ciphertext*
a	0	93	15	p
k	10	96	2	c
t	19	99	14	o
i	8	102	6	g
f	5	103	4	e
a	0	104	0	a
t	19	105	20	u
a	0	106	2	c
u	20	107	23	x
b	1	108	5	f
e	4	129	3	d
r	17	130	17	r
a	0	131	1	b
d	3	132	5	f
a	0	134	4	e
d	3	136	9	j
i	8	140	18	s
l	11	142	23	x
u	20	147	11	l
a	0	149	19	t
r	17	153	14	o
j	9	155	8	i
a	0	158	2	c
n	13	161	18	s
g	6	160	10	k
k	10	167	21	v
a	0	162	6	g
u	20	173	11	l
a	0	188	6	g
n	13	175	6	g

Plaintext	y_i	k_i	$x_i = y_i + k_i(mod\ 26)$	*Ciphertext*
s	18	190	0	a
i	8	181	7	h
l	11	192	21	v
a	0	187	5	f
h	7	6220	13	n
k	10	1	11	l
a	0	3	3	d
n	13	4	17	r
h	7	8	15	p
u	20	11	5	f
b	1	14	15	p
u	20	13	7	h
n	13	18	5	f
g	6	23	3	d
i	8	26	8	i
b	1	31	6	g
e	4	32	10	k
b	1	33	8	i
e	4	38	16	q
r	17	35	0	a
a	0	42	16	q
p	15	47	10	k
a	0	48	22	w
s	18	53	19	t
a	0	55	3	d
a	0	58	6	g
t	19	61	2	c
l	11	65	24	y
a	0	66	14	o
g	6	70	24	y
i	8	73	3	d

Table 5: Proses Dekripsi

Ciphertext	x_i	k_i	$y_i = x_i - k_i(mod\ 26)$	*Plaintext*
n	13	1	12	m
c	2	2	0	a
d	3	3	0	a
m	12	7	5	f
x	23	10	13	n
x	23	9	14	o
c	2	16	12	m
i	8	20	14	o
n	13	22	17	r
j	9	37	24	y
y	24	24	0	a
a	0	39	13	n
k	10	30	6	g
p	15	41	0	a
x	23	36	13	n
u	20	43	3	d
t	19	45	0	a
q	16	49	19	t
t	19	51	20	u
l	11	54	9	j
z	25	57	20	u
a	0	60	18	s
t	19	67	4	e
t	19	68	3	d
t	19	71	0	a
j	9	74	13	n
f	5	77	6	g
t	19	78	19	t
j	9	79	8	i
h	7	82	3	d
h	7	85	0	a

Ciphertext	x_i	k_i	$y_i = x_i - k_i(mod\ 26)$	*Plaintext*
u	20	88	10	k
p	15	93	0	a
c	2	96	10	k
o	14	99	19	t
g	6	102	8	i
e	4	103	5	f
a	0	104	0	a
u	20	105	19	t
c	2	106	0	a
x	23	107	20	u
f	5	108	1	b
d	3	129	4	e
r	17	130	17	r
b	1	131	0	a
f	5	132	3	d
e	4	134	0	a
j	9	136	3	d
s	18	140	8	i
x	23	142	11	l
l	11	147	20	u
t	19	149	0	a
0	14	153	17	r
i	8	155	9	j
c	2	158	0	a
s	18	161	13	n
k	10	160	6	g
v	21	167	10	k
g	6	162	0	a
l	11	173	20	u
g	6	188	0	a
g	6	175	13	n
a	0	190	18	s

Ciphertext	y_i	k_i	$x_i = y_i + k_i(mod\ 26)$	*Plaintext*
h	7	181	8	i
v	21	192	11	l
f	5	187	0	a
n	13	6220	7	h
l	11	1	10	k
d	3	3	0	a
r	17	4	13	n
p	15	8	7	h
f	5	11	20	u
p	15	14	1	b
h	7	13	20	u
f	5	18	13	n
d	3	23	6	g
i	8	26	8	i
g	6	31	1	b
k	10	32	4	e
i	8	33	1	b
q	16	38	4	e
a	0	35	17	r
q	16	42	0	a
k	10	47	15	p
w	22	48	0	a
t	19	53	18	s
d	3	55	0	a
g	6	58	0	a
c	2	61	19	t
y	24	65	11	l
o	14	66	0	a
y	24	70	6	g
d	3	73	8	i

Kesimpulan

Berdasarkan hasil penelitian diatas, maka kita dapat menyimpulkan bahwa pelabelan super (a, d)-$\mathcal{H}$-antimagic total selimut pada shackle graf parasut dapat diaplikasikan pada pembangunan *ciphertext* alfabet, *ciphertext* simbol, dan *stream cipher*.

Referensi

[1] Dafik, Structural Properties and Labeling of Graphs, Tidak Dipublikasikan, Tesis, Australia: School of Information Technology and Mathematical Sciences University of Ballarat, 2007

[2] Diana Hardiyantik, Super (a,d)-H-Antimagic Total Selimut Pada Graf Semi Jahangir, Jurusan Matematika FMIPA Universitas Jember, 2015.

[3] Putri Rizky H.P., Super (a,d)-H-Antimagic Total Covering Pada Shackle Graf Triangular Book, Prosiding Seminar Matematika dan Pendidikan Matematika, **1** (2014), 68-77.

[4] Karinda Rizqy Aprilia, Pelabelan Total Super (a, d)-H-Antimagic Total Covering pada Graf Semi Parasut SP_{2n-1}, *Mathematics in Computer Science Journal*, (2014). Submitted.

[5] Yuli Nur Azizah, Super (a, d)-H-Antimagic Total Dekomposisi Shackle Generalisasi Antiprisma Untuk Pengembangan Ciphertext dan Keterampilan Berpikir Tingkat Tinggi, Jurusan Pendidikan FKIP Universtas Jember, 2016.