JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XV No 1, 2026
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

IMPLEMENTASI KRIPTOGRAFI HILL CIPHER PADA PENGAMANAN DOKUMEN MENGGUNAKAN KODE ASCII

Chintya Jennifer Octoviane Kamal[1)], Eliasta Ketaren[2)], Chriestie Ellyane Julliet Clara Montolalu[3)]
Program Studi Sistem Informasi
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Sam Ratulangi
Alamat Kampus UNSRAT 95115
e-mail: chintyakamal106@student.unsrat.ac.id[1)], eliasketaren@unsrat.ac.id[2)], chriestelly@unsrat.ac.id[3)]

Abstrak
Perkembangan teknologi informasi mendorong meningkatnya penggunaan dokumen digital dalam berbagai aktivitas. Dokumen digital sering kali mengandung informasi penting sehingga memerlukan mekanisme pengamanan untuk mencegah akses oleh pihak yang tidak berwenang. Salah satu metode yang dapat digunakan adalah kriptografi. Penelitian ini bertujuan untuk mengimplementasikan algoritma Hill Cipher dengan matriks kunci berordo 4×4 berbasis kode ASCII pada sistem informasi berbasis web untuk mengamankan dokumen digital. Metode yang digunakan adalah Hill Cipher dengan operasi aritmatika modulo 256. Proses enkripsi dilakukan dengan mengalikan matriks kunci dengan vektor plaintext yang telah dikonversi ke nilai ASCII, sedangkan proses dekripsi menggunakan matriks invers untuk mengembalikan ciphertext menjadi plaintext semula. Sistem yang dibangun mampu melakukan enkripsi dan dekripsi pada dokumen berformat .txt, .docx, dan .pdf. Hasil penelitian menunjukkan bahwa ciphertext yang dihasilkan tidak dapat dibaca secara langsung dan dapat dikembalikan ke bentuk semula menggunakan kunci yang sesuai. Dengan demikian, metode Hill Cipher 4×4 berbasis ASCII dengan modulo 256 dapat digunakan sebagai mekanisme pengamanan dokumen digital.
Kata kunci : ASCII; dekripsi; enkripsi; Hill Cipher; kriptografi

1. Pendahuluan

Perkembangan teknologi informasi menyebabkan meningkatnya penggunaan dokumen digital dalam berbagai bidang. Dokumen digital sering digunakan untuk menyimpan informasi penting sehingga membutuhkan sistem pengamanan yang baik. Tanpa adanya pengamanan, dokumen dapat dengan mudah diakses oleh pihak yang tidak berwenang, terutama jika masih dalam bentuk plaintext. Kasus kebocoran data menunjukkan bahwa sistem keamanan berbasis kontrol akses saja belum cukup untuk melindungi informasi. Oleh karena itu, diperlukan mekanisme tambahan berupa kriptografi. Kriptografi merupakan teknik untuk mengamankan informasi dengan cara mengubah data menjadi bentuk yang tidak dapat dibaca. Salah satu metode kriptografi klasik yang masih relevan adalah Hill Cipher. Algoritma ini menggunakan operasi matriks dan aritmatika modulo untuk melakukan enkripsi dan dekripsi data. Penelitian ini mengembangkan Hill Cipher dengan menggunakan matriks berordo 4×4 berbasis ASCII dan modulo 256 untuk meningkatkan kompleksitas kunci dan memperluas ruang enkripsi. Tujuan penelitian ini adalah mengimplementasikan algoritma Hill Cipher dalam sistem berbasis web yang mampu melakukan enkripsi dan dekripsi dokumen digital secara efektif.

2. Landasan Teori

Landasan teori dalam penelitian ini digunakan sebagai dasar konseptual dalam memahami metode yang diterapkan. Beberapa konsep utama yang digunakan meliputi kriptografi, algoritma Hill Cipher, serta kode ASCII sebagai representasi karakter dalam proses enkripsi dan dekripsi. Pemahaman terhadap konsep-konsep ini diperlukan untuk menjelaskan mekanisme kerja sistem yang dibangun.

Kriptografi

Kriptografi merupakan ilmu yang mempelajari teknik matematis untuk menjaga keamanan informasi, terutama dalam proses komunikasi data. Kriptografi bekerja melalui dua proses utama, yaitu enkripsi dan dekripsi. Enkripsi mengubah plaintext menjadi ciphertext, sedangkan dekripsi mengembalikan ciphertext menjadi plaintext [1].

Hill Cipher

Hill Cipher adalah algoritma kriptografi klasik yang diperkenalkan oleh Lester S. Hill pada tahun 1929. Metode ini menggunakan operasi aljabar linier berupa perkalian matriks untuk melakukan transformasi data. Plaintext terlebih dahulu dikonversi ke dalam bentuk numerik, kemudian diproses menggunakan matriks kunci untuk menghasilkan ciphertext [2].

Matriks Invers dan Syarat Kunci

Dalam Hill Cipher, matriks kunci harus memiliki invers agar proses dekripsi dapat dilakukan. Matriks dikatakan memiliki invers jika determinannya tidak bernilai nol dan relatif prima terhadap modulo yang digunakan [3]. Jika syarat ini tidak terpenuhi, maka ciphertext tidak dapat dikembalikan ke bentuk semula.

Kode ASCII dan Modulo 256

ASCII (American Standard Code for Information Interchange) merupakan standar pengkodean karakter dalam komputer yang merepresentasikan karakter dalam bentuk numerik. Dalam penelitian ini digunakan modulo 256 agar mencakup seluruh karakter ASCII (0–255), sehingga sistem dapat mengenkripsi berbagai jenis karakter, termasuk simbol dan spasi [4].

Gambar 1. Contoh Tabel ASCII

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

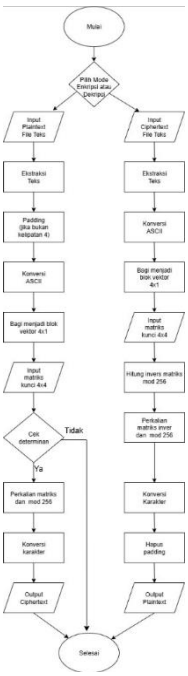
32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	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

3. Metode Penelitian

Metode penelitian meliputi beberapa tahapan yaitu studi literatur, pengumpulan data, perancangan sistem, implementasi, dan pengujian sistem. Penelitian ini bertujuan untuk mengimplementasikan algoritma kriptografi Hill Cipher berbasis kode ASCII dengan matriks kunci berordo 4×4 pada sistem informasi berbasis web.



Gambar 2. Alur Sistem

Metode yang digunakan dalam penelitian ini bersifat eksperimental, yaitu dengan merancang, mengimplementasikan, dan menguji sistem enkripsi serta dekripsi dokumen digital untuk mengetahui kinerja dan keakuratan metode yang digunakan.

Proses Enkripsi

$$C = K \times P \bmod 256 \quad (1)$$

Proses Dekripsi

$$C = K^{-1} \, mod \, 256 \quad (2)$$

Keterangan:
K = matriks kunci
P = plaintext
C = ciphertext

Data yang digunakan dalam penelitian ini berupa data teks dalam bentuk karakter *ASCII* yang diambil melalui input manual melalui file dokumen berformat .txt, .docx, dan .pdf, data tersebut selanjutnya dikonversi ke dalam representasi numerik untuk diproses menggunakan algoritma *Hill Cipher* 4×4.

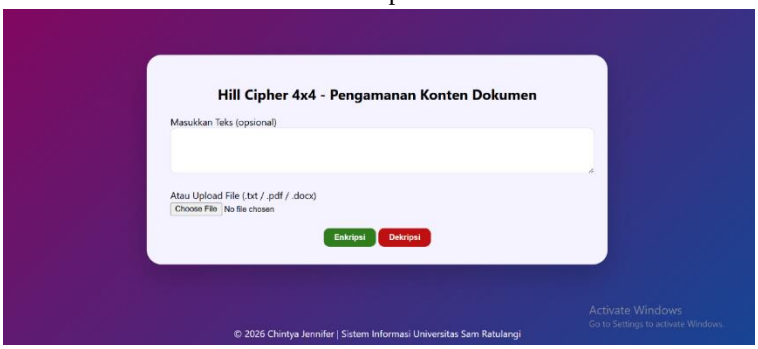
4. Hasil Penelitian

Sistem yang dikembangkan dalam penelitian ini merupakan aplikasi berbasis web yang mengimplementasikan algoritma Hill Cipher berordo 4×4 berbasis ASCII dengan operasi modulo 256. Sistem ini dirancang untuk mengamankan dokumen digital melalui proses enkripsi dan dekripsi secara sederhana dan efisien.

Implementasi Sistem

Sistem yang dibangun merupakan sistem informasi berbasis website yang mengimplementasikan algoritma *Hill Cipher* dengan matriks kunci berordo 4×4 berbasis kode ASCII dan operasi modulo 256. Sistem ini dirancang untuk melakukan proses enkripsi dan dekripsi pada data teks maupun dokumen digital berformat .txt, .pdf, dan .docx. Pada proses enkripsi, sistem membaca isi teks dari dokumen kemudian melakukan ekstraksi teks sebelum karakter dikonversi menjadi nilai ASCII. Nilai ASCII tersebut diproses menggunakan operasi perkalian matriks *Hill Cipher* dan modulo 256 untuk menghasilkan *ciphertext*. Sedangkan pada proses dekripsi, sistem menggunakan matriks invers modulo 256 untuk mengembalikan *ciphertext* menjadi *plaintext* semula. Sistem dikembangkan menggunakan bahasa pemrograman PHP dan dirancang dengan tampilan sederhana agar memudahkan pengguna dalam melakukan pengamanan isi dokumen.

Gambar 3. Tampilan Awal Sistem



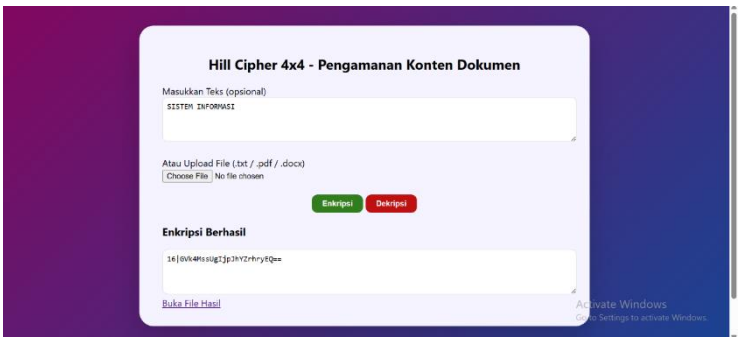
Gambar 3. menunjukkan tampilan awal sistem pengamanan dokumen berbasis website. Pada halaman ini pengguna dapat memasukkan *plaintext* secara langsung melalui textarea atau mengunggah dokumen berformat .txt, .pdf, dan .docx. Sistem juga menyediakan tombol enkripsi dan dekripsi untuk menentukan proses yang akan dijalankan.

```
name="text"></textarea>

<input type="file" name="file">

<button type="submit" name="mode" value="encrypt">Enkripsi</button>

<button type="submit" name="mode" value="decrypt">Dekripsi</button>
```

Kode di atas digunakan untuk membangun antarmuka sistem. Tag *<textarea>* digunakan sebagai input *plaintext*, *<input type="file">* digunakan untuk mengunggah dokumen, sedangkan tombol submit digunakan untuk menentukan proses enkripsi atau dekripsi.

Implementasi Proses Enkripsi

Proses enkripsi dilakukan dengan mengalikan blok *plaintext* dengan matriks kunci menggunakan operasi modulo 256. Hasil enkripsi ditampilkan dalam bentuk *Ciphertext* dan disimpan dalam *file output.*



Gambar 4. Tampilan Hasil Enkripsi

Gambar 4. menunjukkan hasil proses enkripsi *plaintext* menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256. *Plaintext* yang dimasukkan pengguna diproses menjadi *ciphertext* sehingga isi dokumen tidak dapat dibaca secara langsung.

```
$ascii = array_map('ord', str_split($text));

$sum += $key[$r][$c] * $ascii[$i + $c];

$cipher .= chr($sum % 256);
```

Fungsi *ord()* digunakan untuk mengubah karakter *plaintext* menjadi nilai ASCII. Selanjutnya sistem melakukan proses perkalian matriks *Hill Cipher* menggunakan matriks kunci 4×4. Hasil perkalian dihitung menggunakan operasi modulo 256 agar nilai tetap berada pada rentang ASCII 0–255 sebelum dikonversi kembali menjadi karakter *ciphertext*.

Pada penelitian ini, proses enkripsi dilakukan menggunakan algoritma Hill Cipher dengan matriks kunci berordo 4×4 berbasis kode ASCII dan operasi modulo 256. Sebelum dilakukan proses enkripsi, *plaintext* terlebih dahulu dikonversi menjadi nilai ASCII agar dapat diproses secara matematis menggunakan operasi matriks. *Plaintext* yang digunakan pada contoh perhitungan adalah :

Plaintext : DADU

Kunci :
$$\begin{matrix} 1 & 3 & 5 & 7 \\ 2 & 5 & 6 & 1 \\ 3 & 1 & 2 & 4 \\ 1 & 2 & 3 & 1 \end{matrix}$$

Tabel 1. Plaintext

Karakter	ASCII
D	68
A	65
D	68
U	85

Vektor P :
$$\begin{matrix} 68 \\ 65 \\ 68 \\ 85 \end{matrix}$$

Proses enkripsi dilakukan dengan mengalikan matriks kunci dengan vektor *plaintext* menggunakan operasi modulo 256. Perhitungan dilakukan pada setiap baris matriks untuk menghasilkan nilai ciphertext.

Tabel 2. Proses Enkripsi

Baris	Perhitungan	Hasil
1	(1×68 + 3×65 + 5×68 + 7×85)	1198
2	(2×68 + 5×65 + 6×68 + 1×85)	954
3	(3×68 + 1×65 + 2×68 + 4×85)	745
4	(1×68 + 2×65 + 3×68 + 1×85)	487

Hasil perkalian matriks pada proses enkripsi selanjutnya dihitung menggunakan operasi modulo 256 agar nilai tetap berada pada rentang karakter ASCII 0–255. Hasil operasi modulo tersebut menjadi nilai akhir ciphertext.

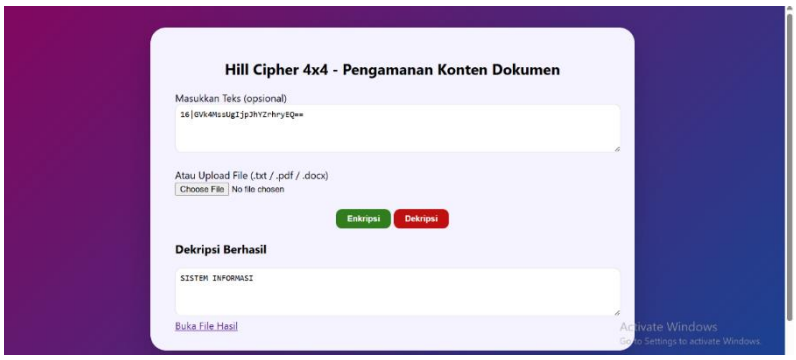
Tabel 3. Hasil Enkripsi

Hasil	Mod 256	*Cipher*
1198	174	174
954	186	186
745	233	233
487	231	231

Vektor tersebut dikalikan dengan matriks kunci dan hasilnya dihitung menggunakan operasi modulo 256. Proses ini menghasilkan ciphertext berupa karakter yang tidak dapat dibaca secara langsung, sehingga mampu menjaga kerahasiaan informasi dalam dokumen.

Implementasi Proses Dekripsi

Proses dekripsi dilakukan dengan menggunakan matriks invers dari kunci yang digunakan pada proses enkripsi. Ciphertext dikonversi kembali ke bentuk numerik, kemudian diproses dalam blok vektor yang dikalikan dengan matriks invers.



Gambar 5. Tampilan Hasil Dekripsi

Gambar 5. menunjukkan proses dekripsi *ciphertext* menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256. Pada proses ini *ciphertext* dikembalikan menjadi *plaintext* semula menggunakan matriks invers sehingga isi dokumen dapat dibaca kembali oleh pengguna yang memiliki kunci yang sesuai.

```
$cipher = base64_decode($encoded);

$ascii = array_map('ord', str_split($cipher));

$sum += $invKey[$r][$c] * ($ascii[$i + $c] ?? 0);
```

Fungsi *base64_decode()* digunakan untuk mengembalikan *ciphertext* dari representasi base64 ke bentuk karakter aslinya sebelum dilakukan proses dekripsi. Sistem melakukan proses perkalian antara matriks invers dengan vektor *ciphertext* untuk mengembalikan data ke bentuk *plaintext* sesuai metode *Hill Cipher*. Operasi modulo 256 digunakan agar hasil perhitungan tetap berada pada rentang karakter ASCII 0–255 sebelum dikonversi kembali menjadi karakter *plaintext* menggunakan fungsi *chr()*. Fungsi *substr()* digunakan untuk mengembalikan *plaintext* sesuai panjang data asli sebelum proses padding dilakukan pada tahap enkripsi. Berdasarkan hasil implementasi, sistem berhasil mengembalikan *ciphertext* menjadi *plaintext* semula menggunakan matriks invers modulo 256. Hal ini menunjukkan bahwa proses dekripsi berjalan sesuai dengan implementasi algoritma *Hill Cipher* 4×4 berbasis ASCII.

Sebelum proses dekripsi dilakukan, matriks kunci harus dipastikan memiliki invers modulo 256. Oleh karena itu dilakukan perhitungan determinan matriks untuk mengetahui apakah matriks dapat diinvers dan digunakan dalam proses dekripsi Hill Cipher. Perhitungan determinan dilakukan menggunakan ekspansi kofaktor pada baris pertama matriks. Determinan matriks 4×4 dihitung menggunakan metode ekspansi kofaktor pada baris pertama matriks dengan rumus:

$$det(k) = \mathrm{a11C11} - \mathrm{a12C12} + \mathrm{a13C13} - \mathrm{a14C14} \quad (3)$$

Subtitusi ke matriks :

$$det(k) = 1(\mathrm{C11}) - 3(\mathrm{C12}) + 5(\mathrm{C13}) - 7(\mathrm{C14}) \quad (4)$$

Nilai kofaktor diperoleh dari determinan submatriks 3×3 yang dihasilkan setelah menghapus baris dan kolom dari elemen yang dihitung. Pada perhitungan determinan matriks 4×4 digunakan metode ekspansi kofaktor. Metode ini dilakukan dengan memilih salah satu baris atau kolom pada matriks, kemudian setiap elemen pada baris atau kolom tersebut dikalikan dengan kofaktornya. Untuk memperoleh nilai kofaktor, baris dan kolom dari elemen yang dihitung dihapus terlebih dahulu sehingga menghasilkan submatriks berordo 3×3.

Kofaktor C11 diperoleh dengan menghapus baris pertama dan kolom pertama pada matriks kunci sehingga diperoleh submatriks 3×3 berikut:

$$\begin{matrix} 5 & 6 & 1 \\ 1 & 2 & 4 \\ 2 & 3 & 1 \end{matrix}$$

$$= 5\,((2)\,(1) - (4)\,(3)) - 6\,((1)\,(1) - (4)\,(2)) + 1\,((1)(3) - (2)\,(2))$$
$$= 5\,(2 - 12) - 6\,(1 - 8) + (3 - 4)$$
$$= 5\,(-10) - 6\,(-7) - 1$$
$$= -50 + 42 - 1$$
$$\mathrm{C11} = -9$$

Kofaktor C12 diperoleh dengan menghapus baris pertama dan kolom kedua pada matriks kunci sehingga diperoleh submatriks:

$$\begin{matrix} 2 & 6 & 1 \\ 3 & 2 & 4 \\ 1 & 3 & 1 \end{matrix}$$

$$= ((2)\,(1) - (4)\,(3)) - 6\,((3)\,(1) - (4)\,(1)) + 1\,((3)\,(3) - (2)\,(1))$$
$$= 2\,(2 - 12) - 6\,(3 - 4) + (9 - 2)$$
$$= 2\,(-10) - 6\,(-1) + 7$$
$$= -20 + 6 + 7$$
$$\mathrm{C12} = -7$$

Kofaktor C13 diperoleh dengan menghapus baris pertama dan kolom ketiga pada matriks kunci sehingga diperoleh submatriks:

$$\begin{matrix} 2 & 5 & 1 \\ 3 & 1 & 4 \\ 1 & 2 & 1 \end{matrix}$$

$$= 2\,((1)\,(1) - (4)\,(2)) - 5\,((3)\,(1) - (4)\,(1)) + 1\,((3)\,(2) - (1)\,(1))$$
$$= 2\,(1 - 8) - 5\,(3 - 4) + (6 - 1)$$
$$= 2\,(-7) - 5\,(-1) + 5$$
$$= -14 + 5 + 5$$
$$\mathrm{C13} = -4$$

Kofaktor C14 diperoleh dengan menghapus baris pertama dan kolom keempat pada matriks kunci sehingga diperoleh submatriks:

$$\begin{matrix} 2 & 5 & 6 \\ 3 & 1 & 2 \\ 1 & 2 & 3 \end{matrix}$$

$$= 2\,((1)\,(3) - (2)\,(2)) - 5\,((3)\,(3) - (2)\,(1)) + 6\,((3)\,(2) - (1)\,(1))$$
$$= 2\,(3 - 4) - 5\,(9 - 2) + 6\,(6 - 1)$$
$$= 2\,(-1) - 35 + 30$$
$$C14 = 7$$

Setelah seluruh kofaktor diperoleh, maka determinan matriks dapat dihitung sebagai berikut:

$$\det(k) = 1\,(-9) - 3\,(7) + 5\,(-4) - 7\,(7)$$
$$= -9-21-20-49$$
$$= -99 \bmod 256$$
$$\det(k) = 157$$

Berdasarkan hasil perhitungan diperoleh determinan matriks sebesar:

$$\det(k) = 157$$

Karena nilai 157 relatif prima terhadap 256, maka matriks kunci memiliki invers modulo 256 dan dapat digunakan dalam proses dekripsi Hill Cipher. Langkah selanjutnya adalah mencari invers modulo dari determinan matriks terhadap modulo 256. Invers modulo dicari untuk memenuhi persamaan:

$$157x = 1 \pmod{256}$$

Setelah dilakukan pencarian invers modulo diperoleh:

$$157^{-1} = 181 \pmod{256}$$

Setelah diperoleh invers modulo determinan dan adjoint matriks, maka matriks invers dapat dihitung menggunakan rumus:

$$k^{-1} = \det(k)^{1} \times \operatorname{adj}(k) \qquad (8)$$

Berdasarkan hasil perhitungan diperoleh matriks invers:

Matriks *Invers* : $\begin{matrix} 73 & 109 & 207 & 54 \\ 94 & 149 & 37 & 208 \\ 181 & 58 & 163 & 91 \\ 52 & 219 & 84 & 135 \end{matrix}$

Vektor *Ciphertext* : $\begin{matrix} 174 \\ 186 \\ 233 \\ 231 \end{matrix}$

Tabel 4. Proses dan Hasil Dekripsi

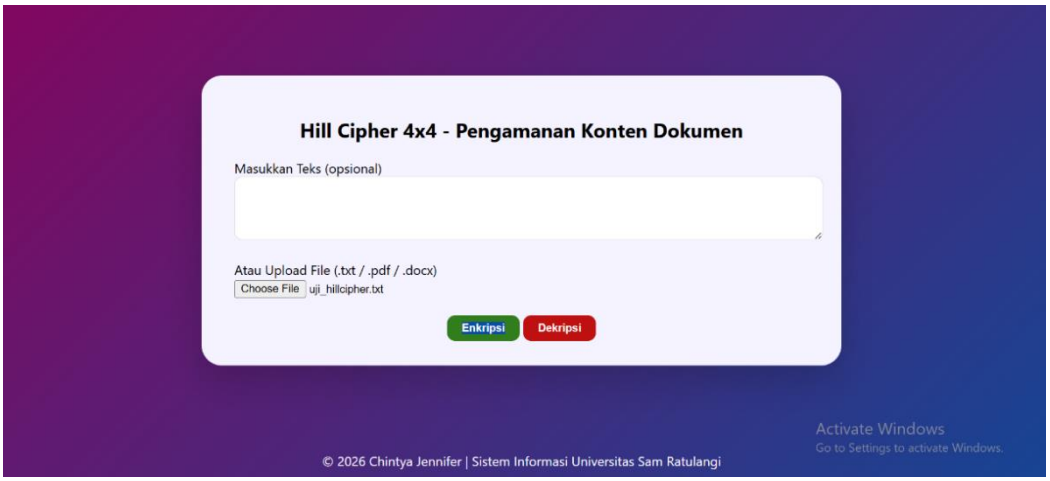
Baris	Perhitungan	Hasil	Mod 256	ASCII
1	(73×174 + 109×186 + 207×233 + 54×231)	93681	68	D
2	(94×174 + 149×186 + 37×233 + 208×231)	70985	65	A
3	(181×174 + 58×186 + 163×233 + 91×231)	100738	68	D
4	(52×174 + 219×186 + 84×233 + 135×231)	100949	85	U

Hasil perhitungan kemudian dikembalikan ke bentuk karakter menggunakan nilai ASCII. Proses ini menghasilkan plaintext yang sama dengan data awal, sehingga membuktikan bahwa sistem mampu melakukan dekripsi dengan benar.

Pengujian Sistem

Pengujian pada file *.txt* menunjukkan bahwa sistem mampu mengenkripsi dan mendekripsi data teks tanpa mengalami perubahan. Pada file *.docx*, sistem berhasil mengekstrak isi dokumen dan memprosesnya dengan baik. Sementara itu, pada file *.pdf*, sistem juga mampu melakukan proses enkripsi dan dekripsi setelah dilakukan ekstraksi teks dari dokumen.

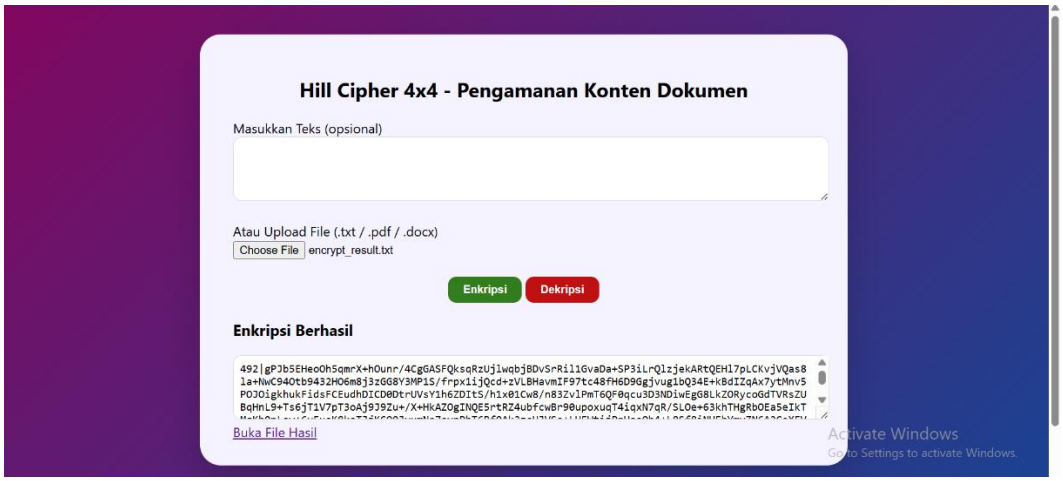
Gambar 6. Tampilan Pengujian Sistem



Gambar 6. dilakukan untuk mengetahui apakah sistem mampu melakukan proses enkripsi dan dekripsi terhadap isi dokumen teks dengan benar menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256.

```
$text = file_get_contents($_FILES['file']['tmp_name']);

$encrypted = encryptHillCipher($text, $key);

$decrypted = decryptHillCipher($encrypted, $inverseKey);
```

Berdasarkan hasil pengujian, sistem berhasil melakukan proses enkripsi dan dekripsi pada file .txt. Hasil dekripsi dapat dikembalikan menjadi *plaintext* semula sehingga menunjukkan bahwa fungsi sistem berjalan dengan baik sesuai implementasi algoritma *Hill Cipher* 4×4 berbasis ASCII dan modulo 256.

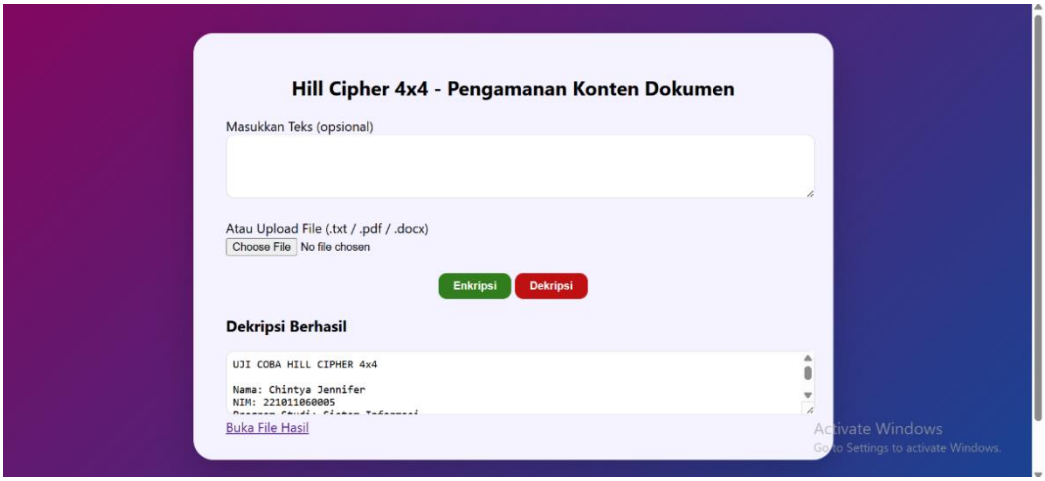


Gambar 7. Tampilan Ciphertext File .txt

Gambar 7. setelah proses enkripsi dilakukan, isi file .txt berhasil diubah menjadi *ciphertext* menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256. *Ciphertext* yang dihasilkan tidak dapat dibaca secara langsung karena karakter asli telah diubah menjadi bentuk terenkripsi.

```
echo htmlspecialchars($encrypted);

echo htmlspecialchars($encrypted);
```

Ciphertext direpresentasikan dalam bentuk Base64 menggunakan fungsi *base64_encode()* agar hasil enkripsi dapat ditampilkan dalam bentuk karakter yang lebih aman dan mudah dibaca. Representasi Base64 juga membantu menghindari munculnya karakter ASCII non-printable pada hasil *ciphertext*. Fungsi *htmlspecialchars()* digunakan untuk menampilkan *ciphertext* pada halaman website tanpa mengubah karakter khusus yang dihasilkan dari proses enkripsi. Berdasarkan hasil pengujian, *ciphertext* berhasil dihasilkan dari proses enkripsi file .txt. Isi file yang sebelumnya dapat dibaca berubah menjadi data terenkripsi sehingga kerahasiaan isi dokumen dapat lebih terjaga.



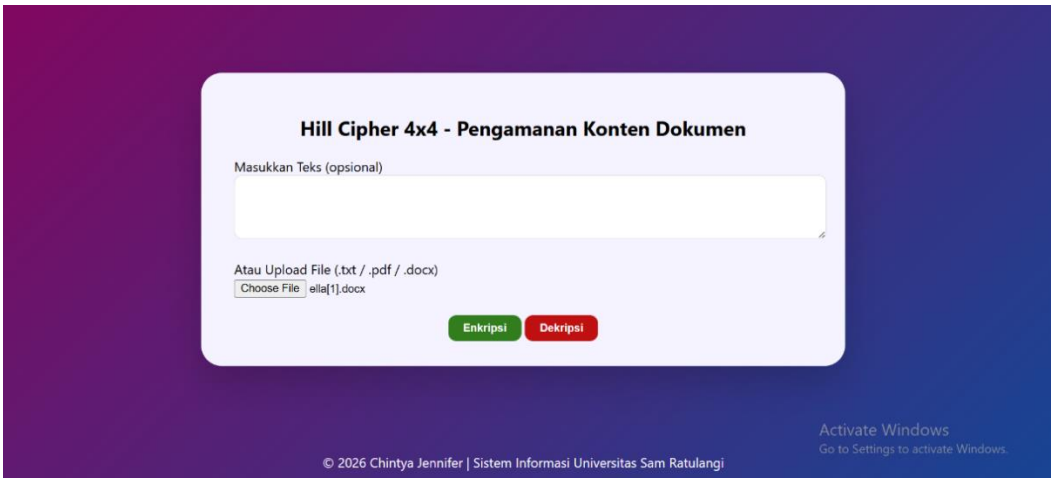
Gambar 8. Tampilan Plaintext File .txt

Gambar 8. setelah proses dekripsi dilakukan, *ciphertext* berhasil dikembalikan menjadi *plaintext* semula menggunakan matriks invers modulo 256. Isi file *.txt* yang sebelumnya terenkripsi dapat dibaca kembali sesuai dengan data asli sebelum proses enkripsi dilakukan.

```
$decrypted = decryptHillCipher($encrypted, $inverseKey);

echo htmlspecialchars($decrypted);
```

Fungsi *decryptHillCipher()* digunakan untuk melakukan proses dekripsi *ciphertext* menggunakan matriks invers *Hill Cipher* sehingga data dapat dikembalikan menjadi *plaintext* semula. Fungsi *htmlspecialchars()* digunakan untuk menampilkan hasil *plaintext* pada halaman website tanpa mengubah karakter asli hasil dekripsi. Hasil pengujian menunjukkan bahwa isi file .txt berhasil dikembalikan ke bentuk semula setelah proses dekripsi dilakukan. Hal ini menunjukkan bahwa implementasi algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256 berjalan dengan

baik pada sistem. Berdasarkan hasil implementasi dan pengujian, sistem mampu melakukan proses enkripsi dan dekripsi file .txt dengan baik sehingga *plaintext* dapat dikembalikan kembali sesuai isi dokumen asli.

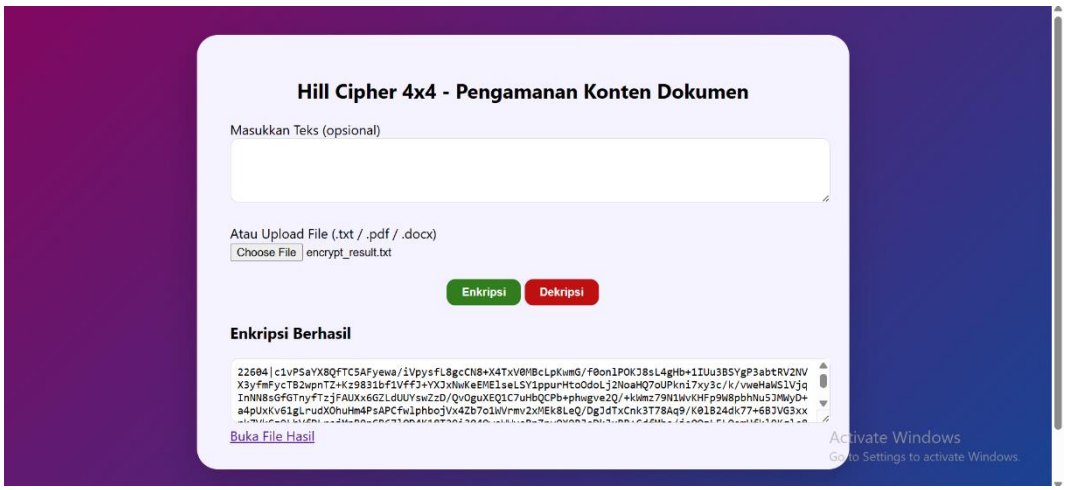


Gambar 9. Tampilan Pengujian File .docx

Gambar 9. menunjukkan proses pengujian dokumen .docx pada sistem. File yang diunggah akan dibaca dan diekstrak isi teksnya sebelum dilakukan proses enkripsi maupun dekripsi.

```
$zip = new ZipArchive;

$data = $zip->getFromName('word/document.xml');

$text = strip_tags($data);
```

Sistem menggunakan *ZipArchive* untuk membaca file .docx karena format .docx tersusun dalam bentuk file XML terkompresi. Isi teks dokumen diambil dari document.xml kemudian fungsi *strip_tags()* digunakan untuk menghapus tag XML sehingga hanya teks yang diproses sebagai *plaintext*



Gambar 10. Tampilan Ciphertext File .docx

Gambar 10. menampilkan hasil enkripsi *file* .docx dalam bentuk *Ciphertext* yang tidak dapat dibaca secara langsung.

```
$result = encryptHillCipher($text, $keyA);

base64_encode($cipher);
```

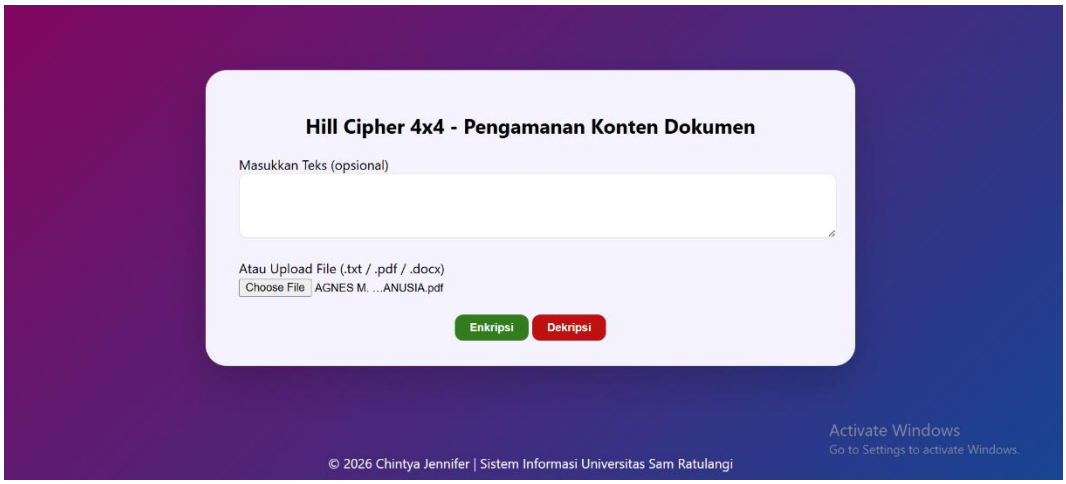
Setelah proses enkripsi dilakukan, isi dokumen .docx berhasil diubah menjadi *ciphertext* menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256. *Ciphertext* yang dihasilkan tidak dapat dipahami secara langsung karena isi teks telah terenkripsi. Fungsi *encryptHillCipher()* digunakan untuk melakukan proses enkripsi terhadap isi teks dokumen .docx menggunakan matriks *Hill Cipher* dan operasi modulo 256. *Ciphertext* direpresentasikan dalam bentuk *Base64* agar hasil enkripsi dapat ditampilkan dalam bentuk karakter yang lebih aman dan mudah dibaca.

Gambar 11. Tampilan Plaintext File .docx

Gambar 11. menunjukkan hasil dekripsi *file* .docx yang berhasil dikembalikan ke bentuk awal tanpa perubahan.

```
$result = decryptHillCipher($text, $invKey);
```

Setelah proses dekripsi dilakukan, *ciphertext* berhasil dikembalikan menjadi *plaintext* semula menggunakan matriks invers modulo 256 sehingga isi dokumen dapat dibaca kembali. Fungsi *decryptHillCipher()* digunakan untuk mengembalikan *ciphertext* menjadi *plaintext* menggunakan matriks invers *Hill Cipher* berbasis modulo 256. Berdasarkan hasil pengujian, sistem berhasil melakukan proses enkripsi dan dekripsi pada file .docx sehingga isi dokumen dapat diamankan dan dikembalikan kembali ke bentuk semula.

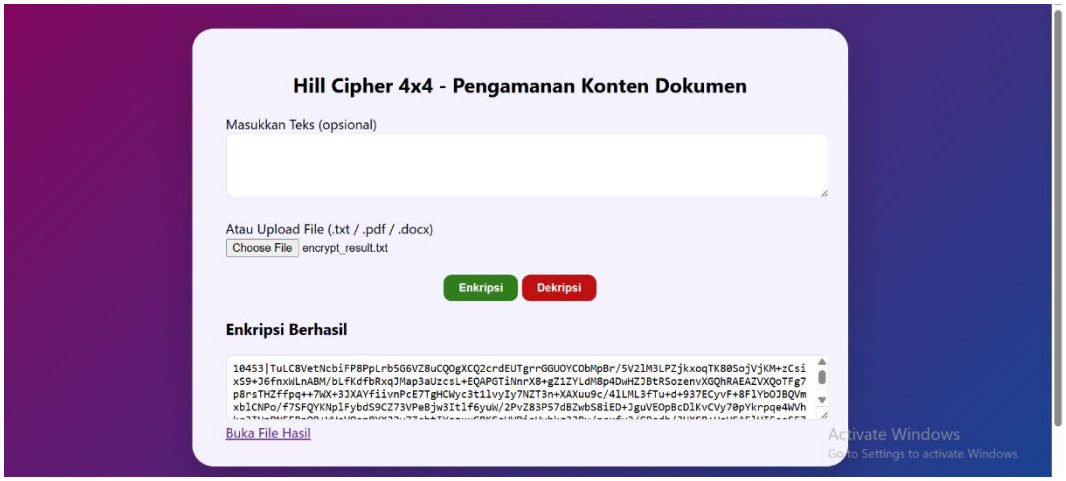


Gambar 12. Tampilan Pengujian File .pdf

Gambar 12. menunjukkan proses pengujian dokumen PDF pada sistem. Setelah file diunggah, sistem melakukan ekstraksi isi teks dari dokumen sebelum diproses menggunakan algoritma *Hill Cipher*.

```
$text = shell_exec("pdftotext \"$tmp\" -");
```

Pengujian file .pdf dilakukan untuk mengetahui kemampuan sistem dalam melakukan proses enkripsi dan dekripsi dokumen .pdf menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256. Sistem menggunakan fungsi *shell_exec()* dengan bantuan utilitas pdftotext untuk melakukan ekstraksi teks dari file .pdf. Proses ini dilakukan karena isi file .pdf tidak dapat langsung diproses sebagai *plaintext* tanpa diekstrak terlebih dahulu.

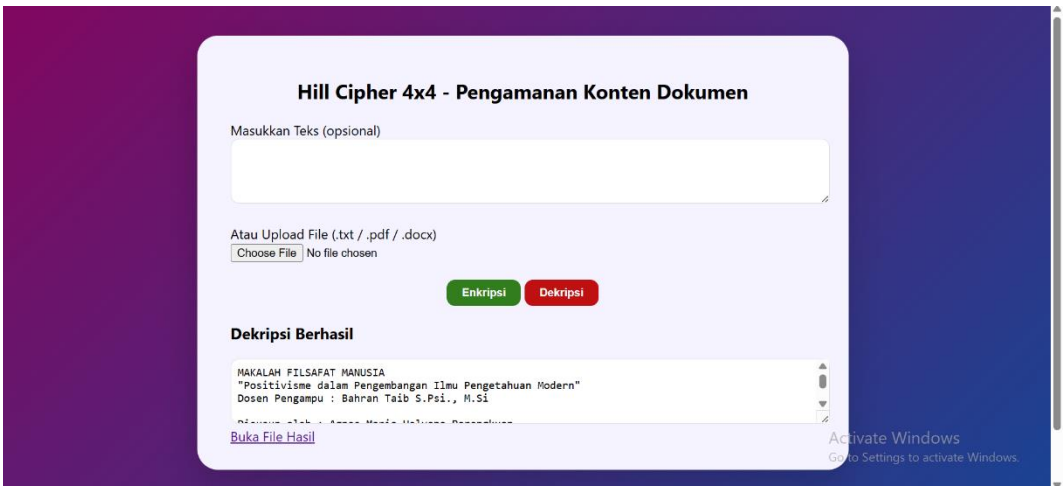


Gambar 13. Tampilan Ciphertext File .pdf

Gambar 13. menampilkan hasil enkripsi *file* .pdf dalam bentuk *Ciphertext* sebagai hasil pengamanan data.

```
$result = encryptHillCipher($text, $keyA);

base64_encode($cipher);
```

Setelah proses enkripsi dilakukan, isi teks dari dokumen .pdf berhasil diubah menjadi *ciphertext* menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256 sehingga isi dokumen tidak dapat dibaca secara langsung. Fungsi *encryptHillCipher()* digunakan untuk melakukan proses enkripsi terhadap hasil ekstraksi teks dari dokumen .pdf menggunakan matriks *Hill Cipher* berbasis modulo 256. *Ciphertext* direpresentasikan dalam bentuk *Base64* agar hasil enkripsi dapat ditampilkan dalam bentuk karakter yang lebih aman dan mudah dibaca.



Gambar 14. Tampilan Plaintext .pdf

Gambar 14. menunjukkan hasil akhir dekripsi *file* .pdf yang telah kembali ke bentuk semula, sehingga membuktikan bahwa sistem berhasil mengembalikan ciphertext menjadi *plaintext* semula.

```
$result = decryptHillCipher($text, $invKey);
```

Setelah proses dekripsi dilakukan, *ciphertext* berhasil dikembalikan menjadi *plaintext* semula menggunakan matriks invers modulo 256 sehingga isi dokumen .pdf dapat dibaca kembali. Fungsi *decryptHillCipher()* digunakan untuk mengembalikan *ciphertext* menjadi *plaintext* menggunakan matriks invers *Hill Cipher* berbasis modulo 256. Hasil pengujian menunjukkan bahwa sistem berhasil melakukan proses enkripsi dan dekripsi pada file .pdf. Isi dokumen dapat dikembalikan kembali ke bentuk semula setelah proses dekripsi dilakukan. Berdasarkan hasil implementasi dan pengujian, sistem mampu melakukan proses enkripsi dan dekripsi dokumen .pdf dengan baik menggunakan algoritma *Hill Cipher* 4×4 berbasis ASCII dan operasi modulo 256.

Pengujian Fungsi

Pengujian fungsi bertujuan untuk memastikan bahwa proses enkripsi dan dekripsi pada sistem *Hill Cipher* 4×4 berbasis ASCII dan modulo 256 berjalan dengan baik sesuai implementasi yang diterapkan. Pengujian dilakukan menggunakan beberapa kombinasi teks dan dokumen dengan format berbeda untuk melihat keberhasilan proses enkripsi dan dekripsi.

Tabel 5. Pengujian Fungsi Input Teks

Plaintext	**Hasil Enkripsi**	**Hasil Dekripsi**	**Kerangan**
SISTEM INFORMASI	16|GVk4MssUgI jpJhYZrhryEQ==	SISTEM INFROMASI	Berhasil
Universitas Sam Ratulangi	25|5rwX4tkfKwu2 nyOvdzm0fOy2E+ WGuB3fkL4a3w==	Universitas Sam Ratulangi	Berhasil
Hill Cipher	11|k5HJygZ1NVHv50n6	*Hill Cipher*	Berhasil

Tabel 6. Pengujian Fungsi Dokumen

Nama File	**Format**	**Enkripsi**	**Dekripsi**	**Keterangan**
uji_hillcipher	.txt	Berhasil	Berhasil	Berhasil
AGNES M. H. PARENGKUAN - FILSAFAT MANUSIA	.pdf	Berhasil	Berhasil	Berhasil
ella[1]	.docx	Berhasil	Berhasil	Berhasil

5. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa implementasi algoritma Hill Cipher dengan matriks kunci berordo 4×4 berbasis kode ASCII dan operasi modulo 256 dapat digunakan untuk mengamankan dokumen digital. Sistem yang dibangun berbasis web mampu melakukan proses enkripsi dan dekripsi dengan baik pada berbagai format dokumen seperti *.txt, .docx,* dan *.pdf*. Hasil pengujian menunjukkan bahwa ciphertext yang dihasilkan tidak dapat dibaca secara langsung sehingga mampu menjaga kerahasiaan informasi. Selain itu, proses dekripsi mampu mengembalikan ciphertext menjadi plaintext semula tanpa perubahan, sehingga integritas data tetap terjaga. Penggunaan matriks berordo 4×4 memberikan tingkat kompleksitas yang lebih tinggi dibandingkan matriks berordo kecil, sehingga meningkatkan keamanan terhadap analisis kriptografi sederhana. Dengan demikian, metode Hill Cipher berbasis ASCII dapat menjadi salah satu alternatif dalam pengamanan dokumen digital, khususnya untuk implementasi sederhana berbasis web.

Daftar Pustaka

[1] M. C. d. L. Soffiana, "Implementasi Algoritma Kriptografi Klasik Caesar Untuk Rancang Bangun Aplikasi E-Voting Berbasis Web (Studi Kasus : sman 10 Tangerang)," *Jurnal Informatika*, vol. 6, pp. 10-18, 2022.

[2] R. G. d. A. D. Wibisono, "Implementasi Algoritma Hill Cipher Menggunakan Kunci Matriks 2x2 Dalam Mengamankan Data Teks," *Jurnal Ilmu Komputer,* vol. 7, pp. 23-30, 2023.

[3] P. Rizki, "Analisis Perbandingan Keamanan Kriptografi Klasik Pada Algoritma Secure Hill Cipher Berbasis Kode ascii Dan Monoalphabetic," *Jurnal Teknologi Informasi,*, vol. 9, pp. 12-20, 2023.

[4] J. E. Sujjada A, "Implementasi Algoritma Hill Cipher Untuk Proses Enkripsi Data Menggunakan Media Citra Digital," *Jurnal Matematika Terapan,* vol. 3, pp. 1-17, 2021.