PENERAPAN ALGORITMA VERNAM CIPHER DAN BASE64 UNTUK KEAMANAN DATA PERNIKAHAN PADA KUA KECAMATAN HARJAMUKTI

Igen Meyasha*[1], Susi Widyastuti[2], Rahmat Maulana[3]
[1] *Program Studi Manajemen Informatika, Sekolah Tinggi Ilmu Komputer Poltek Cirebon*
[2] *Program Studi Sistem Informasi, Sekolah Tinggi Ilmu Komputer Poltek Cirebon*
[3,4] *Program Studi Teknik Informatika, Sekolah Tinggi Ilmu Komputer Poltek Cirebon*
*e-mail: **[1]meyashaigen@gmail.com, [2]umiadell@gmail.com,[3]rahmattmaulana10@gmail.com

Abstrak

Keamanan dari suatu data merupakan hal yang perlu diperhatikan dalam menjaga kerahasiaan informasi terutama yang berisi informasi yang hanya boleh diketahui isinya oleh pihak yang berhak saja. Penelitian ini dilakukan untuk memberikan keamanan data pernikahan pada KUA Kecamatan Harjamukti. Data pernikahan dienkripsi terlebih dahulu menggunakan algoritma Vernam Cipher dengan key yang sudah disiapkan kemudian dilakukan proses encoding menggunakan algoritma Base64. Algoritma Base64 merupakan salah satu algoritma untuk encoding dan decoding. Tujuan dari encoding adalah untuk mengubah suatu data ke dalam format ASCII yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metode yang digunakan untuk melakukan encoding (penyandian) terhadap data binary. Kombinasi kedua algoritma tersebut memberikan sistem keamanan berlapis pada data sehingga sulit untuk dibaca oleh pihak selain KUA Kecamatan Harjamukti meskipun data tersebut diambil atau direkam, dan semua karakter dapat diproses tanpa mengalami kerusakan data seperti perubahan, penambahan, atau pengurangan karakter.

***Kata Kunci** : Vernam Chiper, Base64, Keamanan Data*

1. PENDAHULUAN

Dunia teknologi informasi yang terus berkembang membuat tantangan keamanan data juga menjadi sangat kompleks. Organisasi atau bisnis menyimpan data pelanggan seperti identitas pribadi dan data transaksi pelanggan. Melindungi data pelanggan penting dalam menghadapi ancaman seperti pencurian identitas, serangan dunia maya dan pelanggaran data [1].

Keamanan dari suatu data merupakan hal yang perlu diperhatikan dalam menjaga kerahasiaan informasi terutama yang berisi informasi yang hanya boleh diketahui isinya oleh pihak yang berhak saja. Dalam ilmu pendidikan menjelaskan bahwa ilmu yang mempelajari tentang proses pengamanan data adalah kriptografi. Salah satu cara untuk menjaga keamanan dan kerahasiaan suatu data adalah dengan Teknik enkripsi dan dekripsi [2].

Enkripsi (encryption) adalah Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut plaintext) menjadi pesan yang tersembunyi (disebut ciphertext). Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi data anda disandikan (encrypted) dengan menggunakan sebuah kunci (key). Untuk membuka (decrypt) data tersebut digunakan juga sebuah kunci yang dapat sama dengan kunci untuk mengenkripsi [3].

Adapun teknik kriptografi yang digunakan dalam penelitian ini adalah metode Vernam Chiper dan Base64. Vernam Cipher merupakan algoritma kriptografi yang ditemukan oleh Mayor J. Mougborne dan G. Vernam. Algoritma vernam cipher diadopsi dari one-time pad cipher, dimana dalam hal ini karakter diganti dengan bit (0 atau 1). Dengan kata lain, Vernam

History of article:
Received: April, 2025 : Accepted: April, 2025

Cipher merupakan versi lain dari one-time pad cipher. Algoritma kriptografi Vernam Cipher merupakan algoritma kriptografi berjenis symmetric key [2].

Algoritma Base64 merupakan salah satu algoritma untuk encoding dan decoding. Tujuan dari encoding adalah untuk mengubah bentuk atau format data. Algoritma Base64 mengubah suatu data ke dalam format ASCII yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metode yang digunakan untuk melakukan encoding (penyandian) terhadap data binary [4].

Penelitian di KUA Kecamatan Harjamukti yang beralamatkan di Jln. Pramuka No.2, Kalijaga, Kecamatan Harjamukti, Kota Cirebon, Jawa Barat 45144 memiliki visi mewujudkan keluarga muslim kecamatan harjamukti yang Sakinah, Mawaddah, dan Warahmah di daerah sekitarnya. KUA Harjamukti selain bertugas mencatat dan membantu pernikahan juga bertugas melayani manasik haji, kemasjidan, zakat, ibadah sosial, dan wakaf.

Dalam Menerapkan sistem keamanan data menggunakan algoritma Vernam Chiper dan Base64 untuk mencegah tindak kegiatan kriminal yang bisa merugikan KUA Kecamatan Harjamukti, sekaligus membuat pengolahan data pernikahan KUA Kecamatan Harjamukti. Penulis berharap dapat menjamin keamanan data pernikahan KUA Kecamatan Harjamukti dan sistem pengolahan data KUA Kecamatan Harjamukti dengan sistem keamanan berlapis.

2. METODE PENELITIAN

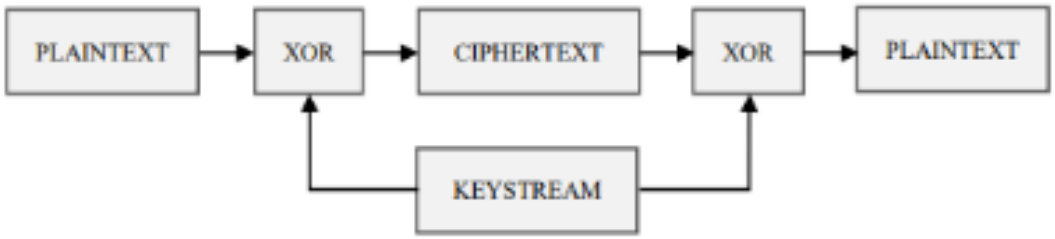
Secara umum pesan/data dikategorikan menjadi dua. Pesan yang bersifat rahasia dan pesan yang tidak bersifat rahasia. Pesan yang tidak bersifat rahasia biasanya tidak akan terlalu diperhatikan. Yang sangat perlu diperhatikan adalah Pesan yang bersifat rahasia, dimana setiap informasi yang ada didalamnya akan sangat berharga bagi pihak yang membutuhkan karena data tersebut dapat dengan mudah digandakan. Untuk mendapatkan informasi didalamnya, biasanya dilakukan berbagai cara yang tidak sah. Salah satu hal penting dalam menjaga kerahasiaan dan keamanan file teks adalah dengan proses enkripsi [5].

Kata kriptografi (cryptography) berasal dari bahasa Yunani yaitu Kryptos (tersembunyi atau rahasia) dan Graphein (menulis atau tulisan). Sehingga dapat dijabarkan bahwa secara harfiah makna kriptografi adalah menulis secara tersembunyi untuk menyampaikan pesan-pesan yang perlu dijaga kerahasiaannya [6]. Kriptografi memiliki arti lain, yaitu suatu ilmu yang mempelajari tentang teknik-teknik matematika yang berkaitan dengan aspek keamanan informasi data atau keamanan pesan dengan menggunakan dua proses dasar kriptografi yaitu enkripsi dan deskripsi [6].

Enkripsi dapat diartikan sebagai cipher atau kode, merupakan proses penyembunyian sebuah data pesan, dengan cara mengubah plaintext (Pesan yang bisa dibaca) menjadi ciphertext (Pesan acak yang tidak bisa dibaca) [6].

Deskripsi merupakan kebalikan dari enkripsi yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal [16], yaitu mengubah ciphertext (Pesan acak yang tidak bisa dibaca) menjadi plaintext (Pesan yang bisa dibaca).

Vernam Cipher merupakan algoritma kriptografi yang ditemukan oleh Mayor J. Mougborne dan G. Vernam. Algoritma Vernam Cipher diadopsi dari one-time pad cipher, dimana dalam hal ini karakter diganti dengan bit (0 atau 1). Dengan kata lain, vernam cipher merupakan versi lain dari one-time pad cipher. Algoritma kriptografi vernam cipher merupakan algoritma kriptografi berjenis symmetric key. Kunci yang digunakan untuk melakukan enkripsi dan deskripsi menggunakan kunci yang sama. Dalam melakukan proses enkripsi, algoritma vernam cipher menggunakan cara stream cipher dimana cipher berasal dari hasil operasi XOR antara bit plaintext dan bit key. Pada cipher aliran, bit hanya mempunyai dua buah nilai, sehingga proses enkripsi hanya menyebabkan dua keadaan pada bit tersebut, yaitu berubah atau tidak berubah. Dua keadaan tersebut ditentukan oleh kunci enkripsi yang disebut dengan aliran bit kunci (keystream). Secara sederhana proses enkripsi dan dekripsi algoritma vernam cipher dapat pada gambar [2].



Gambar 1. Proses Enkripsi dan Dekripsi Algoritma Vernam Cipher

$C_i = P_i \oplus K_i$.....(1)

$P_i = C_i \oplus K_i$.....(2)

Keterangan Persamaan 1 dan Persamaan 2 :

P_i adalah angka biner dari plaintext

C_i adalah angka biner dari ciphertext

K_i adalah angka biner dari key

$\oplus$ adalah proses komputasi XOR

Persamaan 1 merupakan rumus cara kerja enkripsi pada algoritma enkripsi Vernam Cipher. Sedangkan Persamaan 2 merupakan rumus cara kerja dekripsi pada algoritma Vernam Cipher [7]. Langkah-langkah proses enkripsi Vernam Chiper adalah sebagai berikut [2]:

a) Tentukan plaintext yang akan dienkripsi beserta kunci.
b) Ubah plaintext dan key menjadi data biner.
c) Kemudian lakukan proses enkripsi dengan menggunakan logika XOR
d) Setelah didapatkan hasil dari proses enkripsi dengan logika Ex-OR, maka nilai-nilai biner tersebut dikonversikan kembali ke dalam bentuk karakter.

Transformasi BASE64 merupakan salah satu algoritma untuk encoding dan decoding suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metode yang digunakan untuk melakukan encoding (penyandian) terhadap data binary. Karakter yang dihasilkan pada transformasi BASE64 ini terdiri dari A..Z, a..z dan 0..9, serta ditambah simbol "+" dan "/" serta satu buah karakter sama dengan (=) di dua karakter terakhir yang dipakai untuk pengisian pad atau dengan kata lain penyesuaian dan menggenapkan data binary. Karakter simbol yang akan dihasilkan akan tergantung dari proses algoritma yang berjalan. Algoritma BASE64 menggunakan kode ASCII dan kode index BASE64 dalam melakukan proses enkripsi ataupun dekripsinya. Berikut tabel index BASE64 [5].

Tabel 1. Index Algoritma Base64

Index	Char	Index	Char	Index	Char	Index	Char
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v	Pad	=
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

Teknik encoding Base64 sebenarnya sederhana, jika ada satu (string) bytes yang akan disandikan ke Base64 maka caranya adalah [5]:

a) Pecah string bytes tersebut ke per-3 bytes.
b) Gabungkan 3 bytes menjadi 24 bit. Dengan catatan 1 bytes = 8 bit, sehingga 3 x 8 = 24 bit.
c) Lalu 24 bit yang disimpan di-buffer (disatukan) dipecah-pecah menjadi 6 bit, maka akan menghasilkan 4 pecahan.
d) Masing-masing pecahan diubah ke dalam nilai decimal, dimana maksimal nilai 6 bit adalah 63.
e) Tambahkan nilai '0' di akhir binary pada pecahan yang kurang dari 6 bit untuk melengkapinya menjadi 6 bit. Jika ada pecahan yang kosong maka tambahkan padding.
f) Terakhir, jadikan nilai-nilai desimal tersebut menjadi indeks untuk memilih karakter penyusun dari Base64 dan maksimal adalah 63 atau indeks ke 64.

3. HASIL DAN PEMBAHASAN

Adapun dalan pengujian ini penulis sudah menyiapkan plaintext = REPOT dan key = 2adca. Untuk key diusahakan gabungan huruf dan angka agar sulit ditebak. Untuk langkah-langkah pengujian enkripsi dan dekripsi. Enkripsi data menggunakan algoritma Vernam Cipher dengan mengubah plaintext dan key ke dalam desimal sesuai tabel ASCII. Bisa dilihat pada tabel di bawah ini.

Tabel 2. Plaintext dan key

Plaintext	R	E	P	O	T
Desimal	82	69	80	79	84

Key	2	a	d	c	a
Desimal	50	97	100	99	97

Langkah selanjutnya adalah mengubah desimal plaintext dan key ke dalam bentuk biner 8 bit. Untuk awal penulis akan menerapkannya pada desimal plaintext terlebih dahulu. Nilai Desimal diambil dari hasil bagi dan diurutkan dari bawah seperti di bawah ini:

R	
82/2 =	41 sisa 0
41/2 =	20 sisa 1
20/2 =	10 sisa 0
10/2 =	5 sisa 0
5/2 =	2 sisa 1
2/2 =	1 sisa 0
1/2 =	0 sisa 1
82 =	01010010

E	
69/2 =	34 sisa 1
34/2 =	17 sisa 0
17/2 =	8 sisa 1
8/2 =	4 sisa 0
4/2 =	2 sisa 0
2/2 =	1 sisa 0
1/2 =	0 sisa 1
69 =	01000101

P	
80/2 =	40 sisa 0
40/2 =	20 sisa 0
20/2 =	10 sisa 0
10/2 =	5 sisa 0
5/2 =	2 sisa 1
2/2 =	1 sisa 0
1/2 =	0 sisa 1
86 =	01010000

O	
79/2 =	39 sisa 1
39/2 =	19 sisa 1
19/2 =	9 sisa 1
9/2 =	4 sisa 1
4/2 =	2 sisa 0
2/2 =	1 sisa 0
1/2 =	0 sisa 1
79 =	01001111

T	
84/2 =	42 sisa 0
42/2 =	21 sisa 0
21/2 =	10 sisa 1
10/2 =	5 sisa 0
5/2 =	2 sisa 1
2/2 =	1 sisa 0
1/2 =	0 sisa 1
79 =	01010100

Selanjutnya key yang diubah ke biner 8 bit:

2	
50/2 =	25 sisa 0
25/2 =	12 sisa 1
12/2 =	6 sisa 0
6/2 =	3 sisa 0
3/2 =	1 sisa 1
1/2 =	0 sisa 1
97 =	00110010

a	
97/2 =	48 sisa 1
48/2 =	24 sisa 0
24/2 =	12 sisa 0
12/2 =	6 sisa 0
6/2 =	3 sisa 0
3/2 =	1 sisa 1
1/2 =	0 sisa 1
98 =	01100001

d	
100/2 =	50 sisa 0
50/2 =	25 sisa 0
25/2 =	12 sisa 1
12/2 =	6 sisa 0
6/2 =	3 sisa 0
3/2 =	1 sisa 1
1/2 =	0 sisa 1
99 =	01100100

c	
99/2 =	49 sisa 1
49/2 =	24 sisa 1
24/2 =	12 sisa 0
12/2 =	6 sisa 0
6/2 =	3 sisa 0
3/2 =	1 sisa 1
1/2 =	0 sisa 1
100 =	01100011

a	
97/2 =	48 sisa 1
48/2 =	24 sisa 0
24/2 =	12 sisa 0
12/2 =	6 sisa 0
6/2 =	3 sisa 0
3/2 =	1 sisa 1
1/2 =	0 sisa 1
100 =	01100001

Kemudian biner plaintext dan key kita lakukan operasi XOR, yaitu membandingkan bilangan biner 8 bit antara plaintext dengan key. Jika nilainya berbeda maka hasilnya bernilai 1 (True) dan jika nilainya sama hasilnya bernilai 0 (False). Rumus Operasi XOR sebagai berikut:

$C1 = (P1 + K1)\ Mod\ 2$

Operasi XOR antara plaintext 01010010 dan key 00110010 :

$C1 = (0 + 0)\ Mod\ 2 = 0$
$C2 = (1 + 1)\ Mod\ 2 = 0$
$C3 = (0 + 1)\ Mod\ 2 = 1$
$C4 = (1 + 0)\ Mod\ 2 = 1$
$C5 = (0 + 0)\ Mod\ 2 = 0$
$C6 = (0 + 0)\ Mod\ 2 = 0$
$C7 = (1 + 0)\ Mod\ 2 = 1$
$C8 = (0 + 1)\ Mod\ 2 = 1$

Hasil Biner 8 bit diurutkan dari yang paling atas maka hasil sebagai berikut:

$01010010 \oplus 00110010 = 00110011$

Lakukan juga XOR antar plaintext dan key yang lainnya, sehingga didapatkan hasil seperti tabel berikut:

Tabel 3. Hasil Enkripsi XOR Vernam Chiper

Plaintext	R	01010010	E	01000101	P	01010000	O	01001111	T	01010100
Key	2	00110010	a	01100001	d	01100100	c	01100011	a	01100001
XOR		01100000		00100100		00110100		00101100		00110101

Selanjutnya yaitu melakukan proses enkripsi menggunakan algoritma Base64, yaitu dengan mengelompokkan 3 bilangan biner 8 bit menjadi biner 24 bit. Selanjutnya dipecah-pecah menjadi biner 6 bit (sehingga menjadi 4 blok). Jika tidak mencukupi 4 blok, maka tambahkan angka biner '0' (nol) sehingga mencukupi untuk menjadi 4 blok dan nilai desimal untuk indeksnya adalah pad atau '='. Adapun hasil konversinya dapat dilihat pada gambar berikut ini:

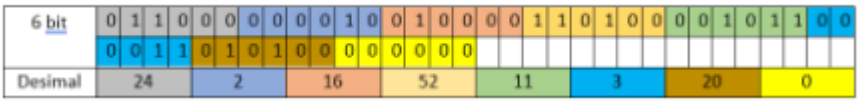
Tabel 4. Hasil Konversi XOR Biner 8 bit ke Biner 6 bit

Selanjutnya adalah mengubah biner 6 bit menjadi desimal. Untuk biner 6 bit yang dibuat dengan maksud melengkapi 4 blok maka tidak tidak perlu dihitung dan nilai indeksnya adalah pad atau '='. Cara mengubah biner menjadi desimal adalah dengan mengalikan tiap-tiap angka biner dengan 2 lalu memangkatkan hasilnya dengan indeks bilangan biner, dimulai dari angka dari paling kanan. Berikut adalah proses mengubah biner menjadi desimal:

a) $011000 = (0*2)^5+(1*2)^4 + (1*2)^3 + (0*2)^2 + (0*2)^1 + (0*2)^0$
   $011000 = (0) + (16) +(8) +(0) + (0) + (0) = 24$
b) $000010 = (0*2)^5+(0*2)^4 + (0*2)^3 + (0*2)^2 + (1*2)^1 + (0*2)^0$
   $000010 = (0) + (0) +(0) +(0) + (2) + (0) = 2$
c) $010000 = (0*2)^5+(1*2)^4 + (0*2)^3 + (0*2)^2 + (0*2)^1 + (0*2)^0$
   $010000 = (0) + (16) +(0) +(0) + (0) + (0) = 16$
d) $110100 = (1*2)^5+(1*2)^4 + (0*2)^3 + (1*2)^2 + (0*2)^1 + (0*2)^0$
   $110100 = (32) + (16) +(0) +(4) + (0) + (0) = 52$
e) $001011 = (0*2)^5+(0*2)^4 + (1*2)^3 + (0*2)^2 + (1*2)^1 + (1*2)^0$
   $001011 = (0) + (0) +(8) +(0) + (2) + (1) = 11$
f) $000011 = (0*2)^5+(0*2)^4 + (0*2)^3 + (0*2)^2 + (1*2)^1 + (1*2)^0$
   $000011 = (0) + (0) +(0) +(0) + (2) + (1) = 3$
g) $010100 = (0*2)^5+(1*2)^4 + (1*2)^3 + (0*2)^2 + (0*2)^1 + (0*2)^0$
   $010100 = (0) + (16) +(0) +(4) + (0) + (0) = 20$
h) $000000 = (0*2)^5+(0*2)^4 + (0*2)^3 + (0*2)^2 + (0*2)^1 + (0*2)^0$
   $000000 = (0) + (0) +(0) +(0) + (0) + (0) = 0$

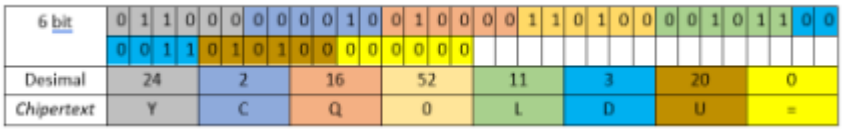
Jika disajikan dalam bentuk tabel maka hasilnya sebagai berikut:

Tabel 5. Hasil Konversi Biner 6 Bit ke Desimal



Selanjutnya yaitu mengubah nilai desimal menjadi karakter sesuai dengan indeks Base64 yang dapat dilihat pada tabel berikut:

Tabel 6. Ciphertext Hasil Enkripsi



Maka Ciphertext yang dihasilkan dari proses enkripsi data adalah YCQ0LDU=, Begitupu hasil enkripsi yang dilakukan oleh sistem di website menggunakan plaintext dan key yang sama adalah sebagai berikut:

$key="2adca0a4efb4c1696f04dd83fe357f8c08ef31a1c095c195c291cd188d95fe51";

Perlu diperhatikan bahwa dalam metode Vernam Cipher, panjang key dengan plaintext harus sama. Karena plaintext memiliki Panjang 5 karakter, maka key yang digunakan adalah 5 karakter awal.

Gambar 2. Hasil Uji Coba Enkripsi

Langkah awal dekripsi data menggunakan algoritma Base64 yaitu menentukan ciphertext dan key yang akan digunakan untuk dekripsi serta akan digunakan hasil dari enkripsi sebelumnya yaitu ciphertext YCQ0LDU= dan key 2adca. Selanjutnya adalah mengubah karakter, menjadi desimal sesuai dengan indeks Base64 yang dapat dilihat pada tabel berikut:

Tabel 7. Konversi Ciphertext Menjadi Desimal

Chipertext	Y	C	Q	0	L	D	U	=
Desimal	24	2	16	52	11	3	20	Pad

Selanjutnya mengubah nilai desimal yang diperoleh menjadi biner 6 bit, dan karakter pad atau '=' tidak perlu dihitung.

a) Mengkonversi desimal 24
   24 = 24/2 = 12 sisa 0
   12/2 = 6 sisa 0
   6/2 = 3 sisa 0
   3/2 = 1 sisa 1
   1/2 = 0 sisa 1
   24 = 011000

b) Mengkonversi desimal 2
   2 = 2/2 = 1 sisa 0
   2 = 000010

c) Mengkonversi desimal 16
   16 = 16/2 = 8 sisa 0
   8/2 = 4 sisa 0
   4/2 = 2 sisa 0
   2/2 = 1 sisa 0
   1/2 = 0 sisa 1
   16 = 010000

d) Mengkonversi desimal 52
   52 = 52/2 = 26 sisa 0
   26/2 = 13 sisa 0
   13/2 = 6 sisa 1
   6/2 = 3 sisa 0
   3/2 = 1 sisa 1
   1/2 = 0 sisa 1
   52 = 110100

e) Mengkonversi desimal 11
   11 = 11/2 = 5 sisa 1
   5/2 = 2 sisa 1
   2/2 = 1 sisa 0
   1/2 = 0 sisa 1
   11 = 001011

f) Mengkonversi desimal 3
   3 = 3/2 = 1 sisa 1

1/2 = 0 sisa 1
3 = 000011

g) Mengkonversi desimal 20
20 = 20/2 = 10 sisa 0
10/2 = 5 sisa 0
5/2 = 2 sisa 1
2/2 = 1 sisa 0
1/2 = 0 sisa 1
20 = 010100

Jika disajikan dalam bentuk tabel maka hasilnya sebgai berikut:

Tabel 8. Konversi Desimal ke Biner 6 bit

Chipertext	Y	C	Q	O	L	D	U	-
Desimal	24	2	16	52	11	3	20	Pad
Biner 6 bit	011000	000010	010000	110100	001011	000011	010100	001000

Selanjutnya adalah mengubah biner 6 bit menjadi biner 8 bit dengan cara mengelompokan 8 angka biner. Padding yang sebelumnya diciptakan untuk menggenapkan 4 blok tidak perlu disertakan. Adapun hasilnya dapat dilihat pada tabel berikut:

Tabel 9. Biner 6 bit ke Biner 8 bit

Chipertext	Y	C	Q	O	L	D	U	-
Desimal	24	2	16	52	11	3	20	Pad
Biner 6 bit	011000	000010	010000	110100	001011	000011	010100	001000
biner 8 bit	01100001	00100000	11010000	10110000	11010000	10010100	00110101	00000000

Selanjutnya mengubah key ke bentuk desimal sesuai dengan kode ASCII, maka didapatkan hasil sebagai berikut:

Tabel 10. Konversi Key ke Bentuk Desimal

Key	2	a	d	c	a
Desimal	50	97	100	99	97

Kemudian kita ubah desimal key ke bentuk biner 8 bit, seperti berikut:

2		a		d	
50/2 =	25 sisa 0	97/2 =	48 sisa 1	100/2 =	50 sisa 0
25/2 =	12 sisa 1	48/2 =	24 sisa 0	50/2 =	25 sisa 0
12/2 =	6 sisa 0	24/2 =	12 sisa 0	25/2 =	12 sisa 1
6/2 =	3 sisa 0	12/2 =	6 sisa 0	12/2 =	6 sisa 0
3/2 =	1 sisa 1	6/2 =	3 sisa 0	6/2 =	3 sisa 0
1/2 =	0 sisa 1	3/2 =	1 sisa 1	3/2 =	1 sisa 1
		1/2 =	0 sisa 1	1/2 =	0 sisa 1
97 =	00110010	98 =	01100001	99 =	01100100

c		a	
99/2 =	49 sisa 1	97/2 =	48 sisa 1
49/2 =	24 sisa 1	48/2 =	24 sisa 0
24/2 =	12 sisa 0	24/2 =	12 sisa 0
12/2 =	6 sisa 0	12/2 =	6 sisa 0
6/2 =	3 sisa 0	6/2 =	3 sisa 0
3/2 =	1 sisa 1	3/2 =	1 sisa 1
1/2 =	0 sisa 1	1/2 =	0 sisa 1
100 =	01100011	100 =	01100001

Dekripsi data menggunakan algoritma Vernam Cipher a. Kumpulkan semua hasil dari 8 bit biner ciphertext dan key dan lakukan operasi XOR, yaitu membandingkan bilangan biner 8 bit antara ciphertext dan key. Jika nilainya berbeda maka hasilnya bernilai 1 (True) dan jika nilainya sama hasilnya bernilai 0 (False) .

Rumus Operasi XOR sebagai berikut:
$C1 = (K\ 1 + P\ 1\ )\ Mod\ 2$
Operasi XOR antara Chipertext 01100000 dan key 00110010 :
$C1 = (0 + 0)\ Mod\ 2 = 0$
$C2 = (1 + 0)\ Mod\ 2 = 1$
$C3 = (1 + 1)\ Mod\ 2 = 0$
$C4 = (0 + 1)\ Mod\ 2 = 1$
$C5 = (0 + 0)\ Mod\ 2 = 0$
$C6 = (0 + 0)\ Mod\ 2 = 0$
$C7 = (0 + 1)\ Mod\ 2 = 1$
$C8 = (0 + 0)\ Mod\ 2 = 0$
Hasil Biner 8 bit diurutkan dari yang paling atas maka hasil sebagai berikut:
01100000 ⊕ 00110010 = 01010010. Lakukan juga XOR antar plaintext dan key yang lainnya, sehingga didapatkan hasil seperti tabel berikut:

Tabel 11. Hasil Dekripsi XOR Vernam Chiper

Chipertext	01100000	00100100	00110100	00101100	00110101
Key	00110010	01100001	01100100	01100011	01100001
XOR	01010010	01000101	01010000	01001111	01010100

Tahap Selanjutnya yaitu mengubah hasil operasi XOR ke bentuk desimal sesuai kode ASCII. Seperti Langkah-langkah berikut:

a) $01010010 = (0*2)^7 + (1*2)^6 + (0*2)^5 + (1*2)^4 + (0*2)^3 + (0*2)^2 + (1*2)^1 + (0*2)^0$
$01010010 = (0)+ (64)+ (0)+ (16)+ (0)+ (0)+ (2)+ (0) = 82$

b) $01000101 = (0*2)^7 + (1*2)^6 + (0*2)^5 + (0*2)^4 + (0*2)^3 + (1*2)^2 + (0*2)^1 + (1*2)^0$
$01000101 = (0)+ (64)+ (0)+ (0)+ (0) + (4) + (0)+ (1) = 69$

c) $01010000 = (0*2)^7 + (1*2)^6 + (0*2)^5 + (1*2)^4 + (0*2)^3 + (0*2)^2 + (0*2)^1 + (0*2)^0$
$01010000 = (0)+ (64)+ (0)+ (16)+ (0)+ (0)+ (0)+ (0) = 80$

d) $01001111 = (0*2)^7 + (1*2)^6 + (0*2)^5 + (0*2)^4 + (1*2)^3 + (1*2)^2 + (1*2)^1 + (1*2)^0$
$01001111 = (0)+ (64)+ (0)+ (0)+ (8)+ (4)+ (2)+ (1) = 79$

e) $01010100 = (0*2)^7 + (1*2)^6 + (0*2)^5 + (1*2)^4 + (0*2)^3 + (1*2)^2 + (0*2)^1 + (0*2)^0$
$01010100 = (0)+ (64)+ (0)+ (16)+ (0)+ (4)+ (0)+ (0) = 84$

Tahap terakhir yaitu ubah Hasil Desimal menjadi char menggunakan Kode ASCII, sehingga didapatkan hasil sebagai berikut:

Tabel 12. Hasil Konversi 8 bit ke Bentuk Desimal

Desimal	82	69	80	79	84
Plaintext	R	E	P	O	T

Maka plaintext yang dihasilkan dari proses dekripsi data adalah REPOT. Hasil sesuai seperti Plaintext awal, adapun dekripsi yang dilakukan oleh sistem menggunakan plaintext dan key yang sama adalah sebagai berikut:
$key="2adca0a4efb4c1696f04dd83fe357f8c08ef31a1c095c195c291cd188d95fe51";
Perlu diperhatikan bahwa dalam metode Vernam Cipher, sistem hanya menggunakan key sepanjang plaintext. Karena plaintext memiliki Panjang 5 karakter begitu pula dengan key memilliki 5 karakter dari awal.

Gambar 3. Hasil Uji Coba Dekripsi

4. KESIMPULAN

Setelah keseluruhan proses dilakukan, yaitu dari tahapan studi literatur sampai pengujian program, maka dapat diambil kesimpulan sebagai berikut: 1. Proses pengamanan data pernikahan pada KUA Kecamatan Harjamukti menggunakan algoritma Vernam Chiper dan Base64 adalah sebagai berikut: langkah pertama adalah proses enkripsi menggunakan metode Vernam Chiper, Langkah kedua enkripsi menggunakan metode Base64, Langkah ketiga dekripsi menggunakan metode Vernam Chiper, dan Langkah keempat dekripsi menggunakan metode Base64. 2. Implementasi algoritma Vernam Cipher dan Base64 mencegah orang yang tidak berhak untuk membaca data pernikahan. Meskipun data tersebut diambil atau direkam, namun karena semua data pernikahan sudah dilakukan enkripsi maka data tersebut menjadi sulit dibaca

5. SARAN

Adapun saran-saran yang penulis berikan untuk pengembangan dan perbaikan sistem ini adalah sebagai berikut:1. Disarankan untuk mengembangkan metode yang penulis terapkan ke tahap yang lebih lanjut. Terutama algoritma Base64 yang memiliki susunan karakter tetap yang diketahui oleh umum dan tidak menggunakan kunci. Solusinya dapat dilakukan perubahan susunan karakter untuk membuatnya sulit ditebak dan menyulitkan proses penyadapan. 2. Disarankan untuk dapat disambungkan dengan web hosting online sehingga website berserta database dapat diakses melalui internet tanpa terbatas akses Wi-Fi lokal.

DAFTAR PUSTAKA

[1] E. T. Rante, M. Alnando, M. Heru, and K. Junior, “Analisis Keamanan Data Pelanggan Menggunakan,” vol. 2, no. 1, pp. 24–29, 2023.

[2] Z. Shabrizqi, “Penerapan Algoritma Vegenere Cipher Dan Vernam Cipher Dalam Penggamanan File Text,” JURIKOM (Jurnal Ris. Komputer), vol. 6, no. 3, pp. 326–332, 2019, ISSN 2407-389X.

[3] M. F. Arif and Muhammad Misdram, “Implementasi Enkripsi Url Pada Website Menggunakan Metode Base64 Dan Rotation13,” Jurnal Spirit, vol. 12, no. 1, pp. 20–25, 2020, P- ISSN : 2085 – 3092, E- ISSN : 2721 – 057X.

[4] P. T. I. M. Kriptografi, K. Rc, D. Base, and R. C. D. Base, “Sistem Keamanan Basis Data Klien,” Jurnal Nasional Teknologi dan Sistem Informasi, Vol.06 No. 01 (2020) 009-018, ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812.

[5] Wina Mariana Br Purba, “Implementasi Algoritma Knapsack Dan Base64 Pada Pengamanan File Teks,” Jurnal Pelita Informatika, Volume 7, Nomor 4, April 2019, ISSN 2301-9425 (Media Cetak), Hal: 552-563.

[6] N. A. Nanda, S. M. S. Silalahi, D. Fatricia Nasution, M. Sari, and I. Gunawan, "Kriptografi dan Penerapannya Dalam Sistem Keamanan Data," J. Media Inform., vol. 4, no. 2, pp. 90–93, 2023, doi: 10.55338/jumin.v4i2.428.

[7] Budiyanto, R. Primananda, and F. A. Bakhtiar, "Implementasi Enkripsi Vernam Cipher dan Distribusi Kunci Three-Pass Protocol untuk Mengamankan Data Chatting pada ATmega328," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 5, no. 3, pp. 1119–1125, 2021, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/8744