ENKRIPSI DAN DEKRIPSI GAMBAR DENGAN MENGGUNAKAN PERPADUAN ALGORITMA BASE64 DAN RC4

Marta Darma Putra[1)], Mardhiya Hayaty[2)]

[1) 2)] Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta
Jl. Ring Road Utar, Condong Catur, Depok, Sleman, Yogyakarta 55281
Email : martadarmaputra@gmail.com[1)], mardhiya_hayati@amikom.ac.id[2)]

Abstrak

Data security is a thing everyone wants to maintain privacy. Both from the security of data in personal memory or even when sending the data. In this research, the encryption process is done by converting the image data into String and stored temporary file as temporary file, encryption method using RC4 cryptographic algorithm because RC4 algorithm operates with XOR method so XOR encryption operation is done very quickly assisted by Base64 encode algorithm in overcoming the data reading 0x00 [NUL] in image file when data is converted to string, this research is done by prototyping method, designed with SWOT method of analysis and system feasibility analysis and system design with flowchart and UML, implemented with flatform Responsive web application, and has been tested and feasible to implement. The results obtained from this research are RC4 can be operated on the image if assisted with base64, and successfully tested with 5 types of images Tiff, png, jpg, gif, bmp with a maximum limit of 2MB file size.

Keywords : *Encryption, RC4, Base64.*

1. Pendahuluan

1.1 Latar Belakang

Keamanan data adalah suatu hal yang diinginkan semua orang untuk menjaga privasi dengan menyem-bunyikan data menggunakan algoritma kriptografi.

Enkripsi dan dekripsi banyak diterapkan hanya pada saat pengiriman data, namun enkripsi dan dekripsi bisa juga diterapkan terhadap file tersimpan untuk men-gamankannya. Seperti permasalahan Pada *smartphone* yang menerapkan pengamanan file dengan folder dikunci, file gambar yang terdapat pada folder tersebut masih bisa diakses melalui aplikasi-aplikasi media sosial, sehingga untuk lebih mengamankannya sebaiknya file sudah terenkripsi.

File gambar saat diubah kedalam bentuk teks mempunyai jumlah character yang besar sehingga jika file gambar dienkripsi dengan mengenkripsi perkarakter harus menggunakan metode algoritma yang cepat dalam melakukan prosesnya, menurut Ir, Yusuf Kurniawan,MT pada bukunya yang berjudul kriptografi keamanan internet dan jaringan komunikasi, algoritma RC4 beroperasi dengan metode XOR maka operasi enkripsi dengan XOR berlangsung sangat cepat sehingga sering digunakan bila diinginkan kecepatan yang memadai.[1]

Dalam melakukan enkripsi gambar dengan algoritma RC4 perlu dibantu oleh algoritma encode radix base 64 karena adanya perbedaan pembacaan penulisan HTML dan File, sehingga saat melakukan dekripsi ada beberapa file yang berubah sesuai penulisan HTML yang mengakibatkan File asli tidak sama dengan file hasil dekripsi.

Berdasarkan latar belakang, maka diperlukan sebuah fasilitas yang dapat melakukan enkripsi dan dekripsi gambar untuk keamanannya pada penyimpanan maupun pengiriman. Oleh karena itu penulis bermaksud untuk merancang aplikasi berbasis web yang bisa melakukan enkripsi dan dekripsi gambar menggunakan algoritma Base64 dan RC4.

1.2 Rumusan Masalah

Perumusan masalah dalam skripsi ini adalah sebagai berikut:

a. Bagaimana cara membuat sebuah system enkripsi dan dekripsi gambar menggunakan algoritma base64 dan algoritma RC4?
b. Apa perbedaan jika proses tidak menggunakan algoritma base64?
c. Apakah ukuran hasil dekripsi sama dengan ukuran data sebelum enkripsi?

1.3 Tinjauan Pustaka

Penelitian tentang algoritma RC4 telah banyak dilakukan sebelumnya, seperti penelitian yang dilakukan Hakim,Khairil,Utami (2014) dengan judul "Aplikasi Enkripsi Dan Dekripsi Data Menggunakan Algoritma RC4 Dengan Menggunakan Bahasa Pemrograman PHP"[2]. Penelitian ini bertujuan untuk mengenkripsi data berupa .txt (angka dan huruf) ke bentuk data plaintext yang tidak dapat dimengerti. Bahasa pemrograman yang digunakan adalah PHP. Dalam kesimpulannya, system yang telah dirancang dan dibangun dalam Pembuatan system enkripsi dan dekripsi data menggunakan Algoritma RC4 dengan meng-gunakan bahasa pemrograman PHP pada SMA Grakarsa Kota Bengkulu dapat memberikan kemudahan dalam proses enkripsi data sehingga dapat menjaga kerahasiaan data.

Penelitian lainnya dilakukan oleh Setiawan,Fiati,Listyorini (2014) dengan judul "Algoritma Enkripsi RC4 Sebagai Metode Obfuscation Source Code Php"[3]. Penelitian ini diharapkan dapat digunakan untuk melindungi source code PHP agar tidak mudah dimanipulasi dan dapat membantu para developer program web yang menggunakan bahasa PHP dalam menjaga hak cipta atas program yang telah dibuatnya. Bahasa yang diguakan adalah PHP dengan algoritma RC4 dan base64.

Penelitian selanjutnya dilakukan oleh Hendarsyahdan dan Wardoyo (2011) dengan judul "Implementasi Protokol Diffie-Hellman Dan Algoritma RC4 Untuk Keamanan Pesan SMS"[4]. Penelitian ini bertujuan untuk keamanan saat proses pengiriman pesan pada SMS dimana untuk menghindari adanya pesan yang dapat disadap atau disusupi oleh pihak-pihak yang tidak diinginkan menggunakan protocol diffie-hellman (diffie-hellman key exchange) dan algoritma RC4.

2. Pembahasan

2.1 Algoritma Base64

2.1.1 Pengertian Algoritma Base64

Algoritma base64 adalah teknik konversi pencodean radix-64, teknik ini merupakan pemetaan untuk merubah input numeric kebentuk karakter sebagai hasilnya. Berikut table pencodean radix 64 dapat dilihat pada table 2.[5]

Tabel 1 *Pencodean Radix Base 64*

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

2.1.2 Cara Penyandian Base64

Proses pengkodean BASE64 adalah sebagai berikut[6]:

a. Input data diubah kedalam Bilangan ASCII dan diambil nilai binernya
b. Nilai biner semua bilangan ASCII digabungkan dan dikelompokkan kedalam 1 kelompok mengandung 6 bit.
c. Setiap kelompok yang berisi 6 bit dipetakan ke 1 karakter yang dapat dicetak dan didasarkan pada nilai 6-bit menggunakan peta set karakter Base64.
d. Karakter padding "=" juga digunakan pada akhir teks yang dikodekan jika jumlah bit (atau jumlah karakter pada plaintext) tidak banyak dari 3. Jika jumlah bit dalam teks adalah 3n + 1, maka encoder menempatkan satu "=" pada akhir teks yang dikodekan, dan jika jumlah bit dalam teks adalah 3n + 2, maka akan menempatkan dua "=" pada akhir keluaran.

Text	A				B				C			
ASCII value	65				66				67			
Bit pattern	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1
Index	16				20				9				3			
Encoded Text	Q				U				J				D			

Gambar 1. *Penyandian base64*

2.2 Konsep Dasar Algoritma RC4

2.2.1 Pengenalan Algoritma RC4

RC4 merupakan jenis dari stream cipher yang berarti operasi enkripsinya dilakukan per karakter 1 byte untuk sekali operasi. RC4 dibuat oleh Ron Rivest Massachusetts Institute if Technology (MIT). RC4 memiliki panjang kunci 2048 bit(256 byte) , namun yang biasa digunakan hanya 40 bit atau 128 bit,sisanya digunakan untuk perulangan kunci yang dipakai berarti 2048 – 40 = 2008 bit atau 2048 – 128 = 1920bit, jika kunci 16 byte(128 bit) berarti K=012345678abcdef dalam bentuk hexadecimal maka, byte yang ke 17 sampai byte yang ke 256 berisi K secara berulang.[5]

2.2.2 Penjadwalan Kunci RC4

System sandi RC4 menggunakan *state*, yaitu larik byte berukuran 256 yang termutasi dan tercampur oleh kunci. Kunci juga merupakan larik byte berukuran 256. Sebelum melakukan enkripsi, dan dekripsi, system sandi RC4 melakukan inisialisasi terhadap *state* dengan algoritma yang bisa dilihat pada gambar 2 Algoritma ini disebut dengan penjadwalan kunci (*Key scheduling*).[7]

```
Input: Kunci
Output: {S[1], ..., S[N]}
    For i = 0 → 255 do
        S[i] = i
    End for
    j = 0
    for i = 0 → 255 do
    j = (j + S[i] + kunci[i mod |kunci|]) mod 256
    swap(S[i],S[j])
    end For
```

Gambar 2. *Penjadwalan kunci*

2.2.3 Enkripsi RC4

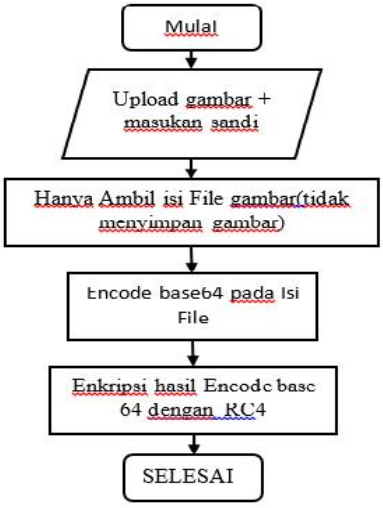
Setelah *state* S terinisialisasi oleh penjadwalan kunci setiap *byte* pada teks asli dikenakan operasi XOR dengan kunci *byte* untuk menghasilkan *byte* pada teks sandi.

Kunci *byte* yang digunakan pada enkripsi dibangkitkan dengan memanfaatkan *state* S. algoritma enkripsi RC4 dapat dilihat pada gambar 3.[7]

Input: P {stream teks asli}
Output: C{stream teks sandi}
$i - 0, j - 0$ {bisa diisi nilai lain}
While P masih memiliki *byte* do
$i - (i + 1) \bmod 256$
$j = (j = S[i]) \bmod 256$
Swap ($S[i]$, $S[j]$)
$k = S[S[i] + S[j]] \bmod 256$
$C - P$ xor k

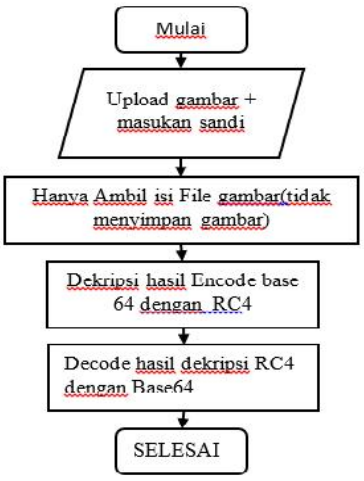
Gambar 3. ***Algoritma RC4***

2.3 Flowchart Sistem
Flowchart enkripsi dapat dilihat pada gambar 4.



Gambar 4. ***Flowchart enkripsi***

Serta flowchart proses dekripsi bisa dilihat pada gambar 5.



Gambar 5. ***Flowchart proses dekripsi***

2.4 Proses Perubahan Bentuk Gambar
2.4.1 Gambar Asli

Gambar asli yang digunakan adalah gambar yang bernama data.png dengan ukuran 587byte, berikut gambarnya bisa dilihat pada gambar 6

Gambar 6. ***Gambar asli***

2.4.2 Hasil encode base64

Gambar yang telah diambil diubah kedalam bentuk teks dan diubah lagi dengan base64 encode, berikut hasil pengubahannya pada gambar 7

iVBORw0KGgoAAAANSUhEUgAAABoAAAAaCAYAAACpSkzOAAAAAX
NSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7
DAcdvqGQAAAHgSURBVEhL7ZRPSwJBGMZnhUXs5LfoG3TuEHVMiog8h
ZSpaQlRGF2CDh2K/ABWXrp3K7CDEOWfEqSDFzv251CQFKiX4G2eWUa
W2V2XlegQDjww7Mw7v533ed9h7U6X/kJD0MByBRWqDYrsn9FoeI9GJjaE
MMc3rNnF2MkR9PreotmdHAWmtkkP58kXuyK2fs91R75ogfSFU76WEXuw1
+4Ms2xBCBxbPiD/TJa0VIlYGgADwta4UlCVtMQN+UNHYq8bzBaEvwSEpW
s9yO7lCxUfv4TOHz4ouMXXkhViqxUBQ4zdWVIWEPKOdKk3wQCk/tw25s1
PAYG02DUFJjN9PbOAYDI8UdMVyjVFuliySvUnAyZAiTKxeJn0+RMRq54n
ZQGhoszGmz0BJLhZo1bnW8AkhMVL5ItciFj1PCkLCOXbDyJvM55t9CAsZgi
x6nlSDiArBMajCHAbOwhbufUGEqnjfaJC4AcKIF9+s4X4Fj2mThQDb0YVYj
ZehUD63LG3YjDKOyOaUYXIoUK0paL38oZEw/ImVG8ihxnCovx1mD703rB
Q7wniMDSjU7pwE0AGfoIgBOIvkRI0I/pEQmA8PMEa9rhBIEeQFPIOk1FRKF
8Ic3zr54kqV9BvaQgaWP8N1KUfcJ/2aeVyBN4AAAAASUVORK5CYII=

Gambar 7. *Pengubahan Base64 encode*

2.4.3 Hasil enkripsi RC4

Hasil encode base 64 di enkripsi dengan algoritma RC4, berikut gambar hasil enkripsi RC4 pada gambar 8

3f1e4e9190a71d82a46f993be10072ddfaff23a8482b90e88527e4490cacbe1e819e
9fc3e8ef48ed07d0d045e246da33de74d3f102b86b9743c449996ce0de5aa75d375a
b58b40060aecd8536d6effcb9bc665e199adb0e05edfd6268ceab767b2bc257464bd
40c352a09653a60480786fe85031f1f740af25220ca20b36d622811609e75f6f0d83
0540d9da895f41708fbcc3a50f28164f8a2786c5a78988f1b0555c3371bd24fee718
9bd651e850a97c861aecb7407b6edd79bbc655427c7beedcd259b8e88fd9ea671cf8
77da694454ab430f39b9d9323771674075f128ccbf41df2b52bbd1dab12e9af3ec98
ba1eba87f5e8d7060ddc2c4b8f404bc2c146ca6aa57cf2cd9a49c6990b33a49b46f9
7517587a0659e7cc69c92b82d509d6191e444328885d552bd8688d035de680af97
c2b06906ae233e76d187f70f7bb37bd75e2e95369b73e39d51b89e83a58c84d23e9
d0c23944655e88893e0c853b77901ae206f0fb40b74ddea1b80938eb41e25efc1d3
9b7fd0a6f79a183eda550707df51472c8849cfb163fc0c7c11a59b2b87a8bddc8d58
a1a43ea4aae265163acd3caa31ea30805d1947fe8028d08f3b8786620fe4530c3247
1eaa7e5cf332af0b7e5e8811eb001a6f2f371eef0aee8278628e3c5886853d402786c
664e7b4e33320297a49f59190370104da825953ae715afa28f51885d46734dbee20
71768eb1f1f446f883c992cf2da01ddaffe303cb71a86c15fc2e9daf5809ea64441d5
29e4e09f2d7245a85e796ae62902fe89dcdcdc0f19a1aeb6b5d22e2f3530fa2f91cbe
c5b129ec5446e31a608208a47aab96ecd8bb8161db2e91aecfdd3638d18c8e368bd
7f64c032e343003b93d7fe27d0f7ff1758e3852b778f4534be54586aaf6f16158b6d1
6912813d724971b4ba52c8c9b674b917f890afb8a78cdf89787fb6cf316d701bcf9a
2de2d86f4c64241f77544308d4b5bb62920296db536b7d0f8a62d8ab55657f26559
deee767e424776a513b75b124120fa55affb4243d527372ea2d6f485d2b61193518f
ae79e41a852471bc3f67e12b346fd72be6a9bd4a4bc86a646652aa979a08b815274
1ed610af

Gambar 8. *Hasil enkripsi RC4*

2.4.4 Tampilan file tersimpan saat dibuka
Tampilan file tersimpan bisa diliat pada gambar 9.

Gambar 9. ***Tampilan File tersimpan***

2.4.5 Hasil dekripsi RC4

Saat melakukan dekripsi data gambar yang disimpan diubah kedalam bentuk String dan diproses dengan dekripsi RC4, berikut gambar hasil dekripsi RC4 terdapat pada gambar 10.

iVBORw0KGgoAAAANSUhEUgAAABoAAAAaCAYAAACpSkzOAAAAAX
NSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAADsMAAA7
DAcdvqGQAAAHgSURBVEhL7ZRPSwJBGMZnhUXs5LfoG3TuEHVMiog8h
ZSpaQlRGF2CDh2K/ABWXrp3K7CDEOWfEqSDFzv251CQFKiX4G2eWUa
W2V2XlegQDjww7Mw7v533ed9h7U6X/kJD0MByBRWqDYrsn9FoeI9GJjaE
MMc3rNnF2MkR9PreotmdHAWmtkkP58kXuyK2fs91R75ogfSFU76WEXuw1
+4Ms2xBCBxbPiD/TJa0VIlYGgADwta4UlCVtMQN+UNHYq8bzBaEvwSEpW
s9yO7lCxUfv4TOHz4ouMXXkhViqxUBQ4zdWVIWEPKOdKk3wQCk/tw25s1
PAYG02DUFJjN9PbOAYDI8UdMVyjVFuliySvUnAyZAiTKxeJn0+RMRq54n
ZQGhoszGmz0BJLhZo1bnW8AkhMVL5ItciFj1PCkLCOXbDyJvM55t9CAsZgi
x6nlSDiArBMajCHAbOwhbufUGEqnjfaJC4AcKIF9+s4X4Fj2mThQDb0YVYj
ZehUD63LG3YjDKOyOaUYXIoUK0paL38oZEw/ImVG8ihxnCovx1mD703rB
Q7wniMDSjU7pwE0AGfoIgBOIvkRI0I/pEQmA8PMEa9rhBIEeQFPIOk1FRKF
8Ic3zr54kqV9BvaQgaWP8N1KUfcJ/2aeVyBN4AAAAASUVORK5CYII=

Gambar 10. *Hasil dekripsi RC4*

3.1.5 Hasil Decode Base64

Gambar 11. *Hasil Decode Base 64*

2.5 Pengujian ukuran data asli dan hasil dekripsi

Ukuran data asli dan hasil dekripsi seharusnya sama jika tidak ada penambahan data atau pengurangan data. Bisa dilihat pada table 3.

Tabel 2. *Pengujian ukuran gambar*

No	Nama file	Sandi	Ukuran Asli	Ukuran enkripsi	Ukuran Dekripsi	STATUS(sama/beda)
1	a.bmp	marta	48 KB	128 KB	48 KB	Sama
2	b.jpg	m	64 KB	172 KB	64 KB	Sama
3	c.png	8096	78.8 KB	210 KB	78.8 KB	Sama
4	d.gif	8097	843 KB	2.19 MB	843 KB	Sama
5	e.tiff	darma	87 KB	233 KB	87 KB	Sama

2.6 Pengujian Pengubahan Sandi yang digunakan

Sandi yang digunakan diubah kedalam base64 dan digabungkan dengan hasil enkripsi dengan SHA.

Tabel 4. *Perubahan Sandi*

No	Sandi	Base64	SHA	Hasil
1	marta	bWFydGE=	54401d296cf92 05c850efc8886 9109c0054506f8	bWFydGE=54401d2 96cf9205c850efc88 869109c0054506f8
2	m	bQ==	6b0d31c0d56322 3024da4569158 46 43ac78c96e8	bQ==6b0d31c0d563 223024da45691584 643ac78c96e8
3	8096	ODA5Ng==	6a3dfbcfc1e4e023 c345366eed bf289eb991038b	ODA5Ng==6a3df bcfc1e4e023c 345366eedbf289eb991038b
4	8097	ODA5Nw==	02f6929d64925285 146a5905c 840433cc056d1b2	ODA5Nw==02f6 929d649252851 46a5905c840433cc056d1b2
5	darma	ZGFybWE	05fa0ef8720539a 61d4f653e63 0bb6d27ea56d3a	ZGFybWE=05fa0ef87 20539a61d4 f653e630bb6d27ea56d3a

2.7 Pengujian masing – masing ekstensi gambar

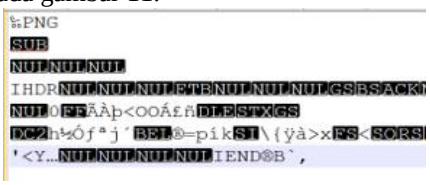
Pengujian dilakukan terhadap 5 type gambar, yaitu bmp, jpg, png, gif, dan tiff, hasil pengujian bisa dilihat pada table 5.

Tabel 5. *Pengujian ekstensi gambar*

No	Nama File	Type	Status (sukses /gagal)
1	a.bmp	BMP	Sukses
2	b.jpg	JPG	Sukses
3	c.png	PNG	Sukses
4	d.gif	GIF	Sukses
5	e.tiff	TIFF	Sukses

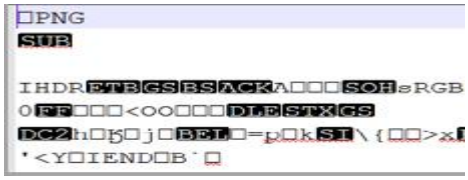
2.8 Pengujian jika tidak menggunakan base64

Pengujian dilakukan dengan melakukan pengamatan terhadap hasil dekripsi menggunakan base64 dan tanpa base64, pada hasil dekripsi terdapat perbedaan pembacaan data ASCII 0x00 atau /0 atau data NULL. Bentuk data file asli saat di buka dengan notepad++ dilihat pada gambar 11.



Gambar 11. *String data asli*

Hasil dekripsi tanpa menggunakan base64 saat dibuka dengan notepad++ dapat dilihat pada gambar 12.



Gambar 12. *Hasil dekrip tanpa base64*

Gambar 13. *Hasil Dekripsi tanpa base64*

Berdasarkan pengamatan terdapat perbedaan hasil yaitu karakter [NUL], data asli dan data hasil dekripsi dengan base64 masih terdapat karakter [NUL] namun saat melakukan proses yag sama tanpa base64 karakter [NUL] hilang, perbedaan tersebut terjadi dikarenakan pembacaan karakter 0x00 pada String.
Semua String pasti diakhiri karakter null ("\0","0x00") dimana nilai ASCII bernilai nol. Saat menggunakan string dengan "HELLO" Dalam array panjangnya dibaca 6 karakter. 5 karakter untuk Hello dan 1 karakter untuk null "\0".[8]
Contohnya adalah " HELLO", saat dialokasikan kedalam memory dengan alamat 2000 yang bisa dilihat pada gambar 15.

2000	H	E	L	L	O	\0	

Gambar 13. *Alokasi Hello pada memory*

Pengamatan juga dilakukan pada perubahan ukuran data saat sebelum dan sesudah proses dekripsi. Perbedaan data bisa dilihat pada table 6.

Tabel 6. *Perbedaan Data tanpa base64*

Ukuran data asli	ukuran data hasil dekripsi
168 bytes	**214 Bytes**

3. Kesimpulan

Berdasarkan penjelasan sebelumnya hingga akhir kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Pengimplementasian algoritma base64 dan RC4 pada enkripsi gambar telah berhasil dibuat. Aplikasi yang dibangun tersebut dapat melakukan enkripsi dan dekripsi dengan baik. Aplikasi ini dibuat dengan Bahasa pemrograman web dan hasilnya terdapat pada halaman web enkmage.cf.
2. [NULL],0x00 adalah tanda akhiran pada String.
3. Menggunakan algoritma base 64 untuk menangani pembacaan String pada data 0x00 atau [NULL] dengan diubah ke dalam bentuk base64.
4. Data gambar yang diubah kedalam string sangat banyak, sehingga diperlukan algoritma yang prosesnya cepat seperti RC4 karena menggunakan metode XOR.
5. Pengamanan sandi pada RC4 sudah diamankan dengan base64 dan SHA
6. Perbedaan ukuran data sebelum dan sesudah dekripsi menandakan adanya pengurangan atau penambahan data.
7. Aplikasi dapat mengirimkan email dengan email *default* enkmages.gmail.com ke email tujuan masing – masing

Daftar Pustaka

[1] M. Ir Yusuf Kurniawan, KRIPOGRAFI Keamanan Internet dan JaringanKomunikasi, Bandung: Informatika, 2004.

[2] E. L. Hakim, Khairil and f. H. Utami, "Aplikasi Enkripsi Dan Dekripsi Data Menggunakan Algoritma RC4 Dengan Menggunakan Bahasa Pemrograman Php.," *Media Informatika,* vol. 10, p. 1, 2014.

[3] O. Setiawan, R. Fiati and T. Listyorini, "Algoritma Enkripsi RC4 sebagai metode Obfuscation Source Code Php.," *Prosiding SNATIF*, vol. 1, 2014.

[4] D. Hendarsyah and R. Wardoyo, "Implementasi Protokol Diffie-Hellman Dan Algoritma RC4 untuk Keamanan Pesan SMS," *IJCCS*, vol. 5, p. 1, 2011.

[5] D. Arius, Kriftografi Keamanan Data dan Komunikasi, Yogyakarta: Graha Ilmu, 2006.

[6] G. Singh and Supriya, "Modified Vigenere Encryption Algorithm and its Hybrid Implementation with Base64 and AES," *Second International Conference on Advanced Computing, Networking and Security*, 2013.

[7] S. Rifki, Kriftografi Untuk Keamanan Jaringan, Yogyakarta: ANDI, 2012.

[8] D. S. 2. S. D. 3, Chapter 3: Characters and Strings.

Biodata Penulis

Marta Darma Putra, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Informatika UNIVERSITAS AMIKOM Yogyakarta, lulus tahun 2017.

Mardhiya Hayaty, memperoleh gelar Sarjana Teknik (ST), Jurusuan Teknik Informatika Universitas Achmad Dahlan Yogyakarta, lulus tahun 2003. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Teknologi Informasi Fakultas Ilmu Komputer Universitas Amikom Yogyakarta, lulus tahun 2012, saat ini menjadi Dosen di Universitas Amikom Yogyakarta.

ISSN : 2302-3805