Kriptografi Visual Sederhana Berbasis Nilai Modulo pada Intensitas Warna Gambar RGB

Ignatius Evan Daryanto (13511019)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
ignatius.evan.d@s.itb.ac.id

***Abstract*—Pada makalah ini akan dijelaskan mengenai implementasi Kriptografi Visual berdasarkan nilai Modulo masing-masing intensitas warna pada gambar RGB. Kombinasi nilai modulo ini kemudian dikonversi dalam heksadesimal yang kemudian diterjemahkan berdasarkan table ASCII. Metode ini tidak akan menjadikan seseorang curiga terhadap gambar yang kita kirim karena perubahan warna yang dilakukan tidak signifikan, sehingga tidak terlihat sebagai *noise*.**

***Index Terms*—Kriptografi Visual, ASCII, modulo, RGB**

I. PENDAHULUAN

Kriptografi merupakan salah satu cabang matematika yang merupakan ilmu penulisan pesan rahasia. Kriptografi memiliki anak cabang keilmuan, yaitu kriptoanalisis yang merupakan ilmu untuk memecahkan kriptografi. Masalah utama yang ditangani kriptografi adalah keamanan suatu data/pesan. Aplikasi komputer perlu untuk melindungi data mereka dari akses yang tidak sah. Anda tidak ingin orang mengintip data Anda (Anda ingin kerahasiaan), dan Anda tidak ingin seseorang mengubah data tanpa sepengetahuan Anda (Anda ingin meyakinkan integritas data Anda).[1]

Perkembangan kriptografi kini sudah menghasilkan enkripsi yang beragam algoritma dan metodenya, seperti Kriptografi Caesar, Autokey, One-Time Pad, MD5, sha1, dan sha256. Namun enkripsi-enkripsi tersebut memiliki kelemahan, yaitu seseorang akan tau bahwa teks yang dikirim merupakan hasil enkripsi atau tidak. Teks hasil enkripsi tentu saja terdiri dari huruf-huruf acak yang tidak dapat terbaca. Hal ini tentu saja menjadikan para *cracker* penasaran dan berusaha untuk memecahkan (mendekripsi) kode tersebut.

Oleh karena itu, dibutuhkan enkripsi yang benar-benar tidak terlihat, yaitu Kriptografi Visual. Dengan Kriptografi visual, kita dapat menyembunyikan pesan yang ingin kita sampaikan ke dalam sebuah gambar tanpa mengubah gambar tersebut secara signifikan, sehingga orang tidak dengan mudah mengetahui bahawa di dalam gambar tersebut terselip sebuah kode yang mempresentasikan sebuah pesan.

II. DASAR TEORI

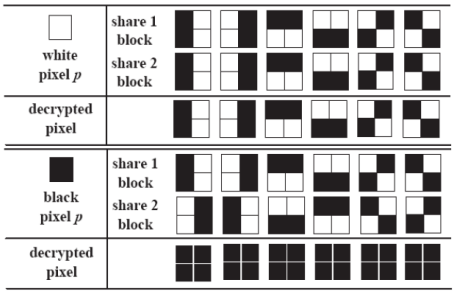
A. Kriptografi

Kriptografi adalah ilmu sekaligus seni untuk menjaga keamanan pesan. Keamanan pesan diperoleh dengan menyandikannya menjadi pesan yang tidak mempunyai makna. Zaman sekarang ini kerahasiaan informasi menjadi sesuatu yang penting. Informasi yang rahasia perlu disembunyikan agar tidak diketahui oleh orang yang tidak berhak. [2]

Pesan yang dirahasiakan disebut ***plainteks*** (*plaintext*, artinya teks jelas yang dapat dimengerti), sedangkan pesan hasil penyandiannya disebut ***cipherteks*** (*ciphertext*, artinya teks tersandi). Pesan yang telah tersandikan dapat dikembalikan lagi ke pesan aslinya hanya oleh orang yang berhak (orang yang berhak adalah orang yang mengetahui metode penyandian atau memiliki kunci penyandian).[2]

B. Perkembangan Kriptografi Visual

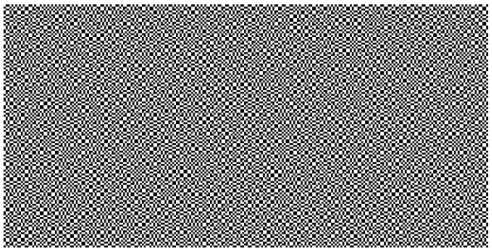
Kriptografi visual adalah teknik kriptografi yang memungkinkan informasi visual (gambar, teks, dll) yang akan dienkripsi sedemikian rupa sehingga dekripsi dapat dilakukan oleh sistem visual manusia, tanpa bantuan komputer. Teknik kriptografi visual untuk gambar hitam dan putih (Basic) diperkenalkan oleh Naor dan Shamir. Setiap informasi rahasia visual (gambar, teks, dll) dianggap sebagai gambar dan enkripsi ini dilakukan dengan menggunakan algoritma sederhana untuk menghasilkan salinan n *shares* tergantung pada jenis skema struktur akses.[3]



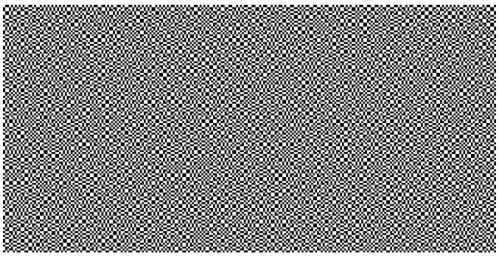
Gambar 2.1 Skema *2 to* 2 pada Kriptografi Visual pertama yang dibuat Naor dan Shamir[1]

[1] Idem.

Teknik kriptografi visual untuk gambar hitam dan putih (Basic) diperkenalkan oleh Naor dan Shamir[4]. Setiap informasi rahasia visual (gambar, teks, dll) dianggap sebagai gambar dan enkripsi ini dilakukan dengan menggunakan algoritma sederhana untuk menghasilkan salinan n *shares* tergantung pada jenis skema struktur akses. Struktur sederhana adalah akses. 2 dari 2 skema di mana citra rahasia dienkripsi menjadi 2 *shares* dan keduanya diperlukan untuk dekripsi sukses. *Shares* tersebut adalah titik acak tanpa mengungkapkan informasi rahasia.[3]



(a)



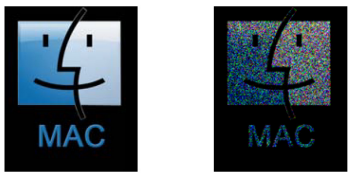
(b)

Gambar 2.2 (a) Kunci dan (b)hasil enkripsi pada Kriptografi Visual pertama yang dibuat Naor dan Shamir[2]

Threshold Visual Cryptography Scheme for Color Images with No Pixel Expansion

VCS (*Visual Cryptography Scheme*) jenis ini sudah mendukung gambar berwarna. Skema ini juga mendukung beberapa properti yang diinginkan, yaitu :[3]

1. Mendukung gambar dengan banyak warna
2. Tidak ada ekspansi piksel
3. Tidak ada *preprocessing* dari gambar asli (seperti *dithering* dan *block averaging*)
4. Mendukung *k-out-of-n*, *threshold setting,* dan
5. Mendukung sejumlah tingkat warna dalam proses pembagian rahasia.



(a) (b)

Gambar 2.3 (a) Gambar original dan (b)hasil enkripsi pada Threshold Visual Cryptography Scheme for Color Images with No Pixel Expansion

Selama transformasi, digunakan teknik probabilistik agar tidak ada ekspansi pixel. Selain itu, sistem ini juga memungkinkan pengguna dari VCS untuk memilih jumlah warna yang dimiliki pada gambar yang direkonstruksi. Kita akan melihat bahwa fitur ini memungkinkan pengguna untuk mengontrol kualitas gambar.[5]

Namun pada enkripsi ini menghasilkan *noise* yang cukup ekstrim pada gambar, sehingga dapat memunculkan kecurigaan seseorang terhadap gambar tersebut.

III. KONVERSI KARAKTER ASCII KE PERUBAHAN WARNA PIKSEL

Pada bab ini akan dibahas rancangan awal tentang Kriptografi Visual berdasarkan nilai modulo pada intensitas warna. Pada kriptografi ini kita memanfaatkan nilai modulo pada masing-masing nilai warna RGB nya. Kombinasi nilai-nilai modulo ini akan merepresentasikan suatu huruf tertentu. Selain itu sistem ini juga dapat dikombinasikan dengan enkripsi sederhana lainnya, seperti enkripsi *Caesar, Autokey,* dan *One-Time Pad.*

Misalkan saja kita akan membangun sebuah sistem kriptografi visual berbasis nilai modulo pada intensitas warna gambar dengan spesifikasi input pesan dibaca berdasarkan kode karakter ASCII dan dipecah menjadi 2, sehingga masing-masing karakter dipresentasikan dengan 2 piksel dengan memproses 4 bit masing-masing karakter ASCII. Masing-masing warna piksel akan dirubah (dengan menambahkan atau mengurangkan nilai intensitas warnanya) sehinggi nilai modulo nya sesuai dengan tabel 3.1

Tabel 3.1 Tabel konversi Nilai Hex dengan nilai intensitas warna

Nilai Hex	R mod 4	G mod 4
0	0	0
1	1	0
2	2	0
3	3	0
4	0	1
5	1	1
6	2	1
7	3	1
8	0	2
9	1	2
A	2	2
B	3	2
C	0	3
D	1	3
E	2	3
F	3	3

Sebagai contoh warna *Spring Green* pada *Standart RGB Palette* akan kita konversi warnanya sehingga dapat mewakili nilai Hexadecimal tertentu (Lihat gambar 3.1). Warna aslinya memiliki komposisi warna Red = 54, Green = 206, dan Blue = 51. Warna ini kemudian dikonversi nilai komposisi warnanya sesuai dengan nilai

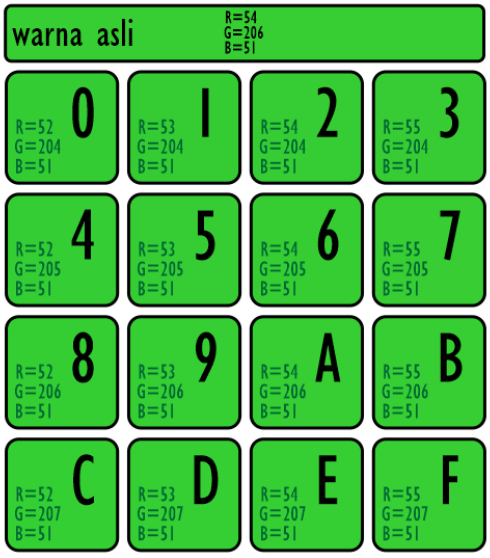
[2] Moni Naor dan Adi Shamir, "*Visual Cryptography*", Eurocrypt, 1994

[3] Xiaoyu Wu1, Duncan S. Wong2, and Qing Li, "*Threshold Visual Cryptography Scheme for ColorImages with No Pixel Expansion*", Huangsan, China, Desember 2009, hal 1.

Heksadesimal yang akan dimasukan. Karena pergeseran nilai warnanya kecil, menyebabkan perubahannya tidak terlihat dengan mata telanjang. Konversi warna ini dapat dirumuskan menjadi :

$$C' = C - C \bmod 4 + i, 0 \leq i \leq 3$$

Dengan C' = intensitas warna baru
C = intensitas warna lama
i = nilai yang ditambahkan sesuai dengan nilai heksadesimal yang akan dimasukan



Gambar 3.1 Salah satu contoh konversi warna

Sedangkan warna *blue* dapat dijadikan sebagai *synchronizer*. Setiap karakter ASCII membutuhkan 2 piksel, sehingga untuk mensinkronkan sisa bagi 4 kedua piksel harus sama, sehingga secara matematis,

$$C_B^{2\prime} = C_B^2 - C_B^2 \bmod 4 + C_B^1 \bmod 4$$

dengan C_B^1 : intensitas warna biru pada piksel pertama
C_B^2 : intensitas warna lama biru pada piksel kedua
$C_B^{2\prime}$: intensitas warna baru biru pada piksel kedua

Sebagai contoh misalkan kita akan mengkonversi dua buah piksel dengan warna *Banana Yellow* (#CCCC33) dan *Dusty Rose* (#CC6699) dengan menyisipkan huruf O (x4F). Sehingga untuk piksel pertama, R mod 4 = 0 dan G mod 4 = 1, sedangkan pada piksel kedua, R mod 4 = 3 dan G mod 4 = 3. Sehingga warna piksel tersebut diproses sebagai berikut,

$$\begin{aligned} C_R^{1\prime} &= C_R^1 - C_R^1 \bmod 4 + i \\ &= 204 - 204 \; div \; 4 + 0 \\ &= 204 \\ C_G^{1\prime} &= C_G^1 - C_G^1 \bmod 4 + i \\ &= 204 - 204 \; div \; 4 + 1 \\ &= 205 \end{aligned}$$

$$\begin{aligned} C_R^{2\prime} &= C_R^2 - C_R^2 \bmod 4 + i \\ &= 204 - 204 \; div \; 4 + 3 \\ &= 207 \\ C_G^{1\prime} &= C_G^1 - C_G^1 \bmod 4 + i \\ &= 102 - 102 \; div \; 4 + 3 \\ &= 103 \end{aligned}$$

Sychronizer

$$\begin{aligned} C_B^{2\prime} &= C_B^2 - C_B^2 \bmod 4 + C_B^1 \bmod 4 \\ &= 153 - 153 \bmod 4 + 51 \bmod 4 \\ &= 155 \end{aligned}$$

Sehingga setelah warna tersebut diproses dan dikonversi ke dalam bentuk heksadesimal menghasilkan warna #CCCD33 dan #CF679B.(Perhatikan gambar 3.2)

Gambar 3.2 Hasil konversi warna Banana Yellow (#CCCC33) dan Dusty Rose (#CC6699) (atas) dengan menyisipkan data huruf O (bawah)

IV. PENYISIPAN PESAN KE DALAM GAMBAR

Seperti yang sudah kita bahas pada bab 3, penyisipan data setiap karakter membutuhkan 2 piksel. Sehingga gambar dengan ukuran m x n piksel dapat memuat (m x n)/2 karakter. Orientasi pembacaan data dapat dilakukan secara horizontal maupun vertical. Sedangkan dalam makalah ini akan dibahas mengenai orientasi horizontal.

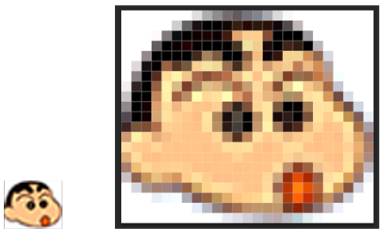
Jika jumlah karakter kurang dari kapasitas maksismum pada gambar, maka nilai *synchronize* harus salah, yang berarti $C_B^n \bmod 4 \neq C_B^{n+1} \bmod 4$ (dengan n bilangan ganjil).

Sehingga jika kita rangkum derikut *pseudocode* dari algoritma sistem ini.

```
w = width(img)
h = height(img)
if ((w*h)>(length(msg)*2)) then
 wpos = 1
 hpos = 1
 i traversal [0..(length(msg)-1)]
  ir1 = SetIR(GetFirstHex(msg[i]))
  ig1 = SetIG(GetFirstHex(msg[i]))
  ir2 = SetIR(GetSecHex(msg[i]))
  ig2 = SetIG(GetSecHex(msg[i]))
  img.Rcolor[wpos,hpos]=img.Rcolor[wpos,hpos]-img.Rcolor[wpos,hpos]mod4+ir1
  img.Gcolor[wpos,hpos]=img.Gcolor[wpos,hpos]-img.Gcolor[wpos,hpos]mod4+ig1
  sync = img.Bcolor[wpos,hpos]mod4
  if wpos=w then
   wpos = 1
   hpos++
  else
   wpos++
  img.Rcolor[wpos,hpos]=img.Rcolor[wpos,hpos]-img.Rcolor[wpos,hpos]mod4+ir2
  img.Gcolor[wpos,hpos]=img.Gcolor[wpos,hpos]-img.Gcolor[wpos,hpos]mod4+ig2
  img.Bcolor[wpos,hpos]=img.Bcolor[wpos,hpos]-img.Bcolor[wpos,hpos]mod4+ sync
  if wpos=w then
   wpos = 1
   hpos++
  else
   wpos++
 if (((w*h)-(length(msg)*2))>2) then
  sync = img.Bcolor[wpos,hpos]mod4
  if wpos=w then
   wpos = 1
   hpos++
  else
```

```
  wpos++
 if img.Bcolor[wpos,hpos] mod 4 = sync then
  if Bcolor[wpos,hpos]=255 then
   img.Bcolor[wpos,hpos]--
  else
   img.Bcolor[wpos,hpos]++
else
 output("Text Overflow")
```

Misalkan saja pada gambar 4.1 akan disisipkan sebuah pesan "STRUKTUR DISKRIT". Hal pertama yang kita analisis adalah apakah gambar tersebut cukup untuk menyisipkan pesan tersebut. Gambar 4.1 berukuran 25x21px = 525 px. Sedangkan pesan yang akan ditulis berjumlah 16 karakter, sehingga hanya diperlukan 32 piksel untuk menyisipkan pesan ini ke dalam gambar.



(a) (b)

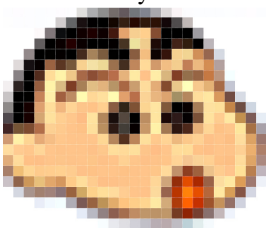
Gambar 4.1 Contoh gambar yang akan disisipkan suatu pesan didalamnya

Setelah diproses berdasarkan algoritma yang sudah dibuat sebelumnya, maka terjadi perubahan data warna pada gambar seperti pada tabel 4.1

Tabel 4.1 Tabel konversi nilai intensitas Warna untuk menyisipkan pesan "STRUKTUR DISKRIT"

Posisi Piksel (x,y)	Input Hex	Warna Asal (#RRGGBB)	Warna Keluaran (#RRGGBB)
(1,1)	5	#FFFFFF	#FDFDFF
(2,1)	3	#FFFFFF	#FFFCFF
(3,1)	5	#FFFFFF	#FDFDFF
(4,1)	4	#FFFFFF	#FCFDFF
(5,1)	5	#FFFFFF	#FDFDFF
(6,1)	2	#FFFFFF	#FEFCFF
(7,1)	5	#FFFFFF	#FDFDFF
(8,1)	5	#E0DDDE	#E1DEDE
(9,1)	4	#AFAEAF	#ACADAF
(10,1)	B	#8F8D8D	#8F8E8F
(11,1)	5	#776C6C	#756D6C
(12,1)	4	#645A5E	#64595C
(13,1)	5	#74767B	#73767B
(14,1)	5	#81868E	#81858F
(15,1)	5	#A9A9AD	#A9A9AD
(16,1)	2	#DCD9DA	#DDD8D9
(17,1)	2	#F9F7F7	#FAF4F7
(18,1)	0	#FFFFFF	#FCFCFF
(19,1)	4	#FFFFFF	#FCFDFF
(20,1)	4	#FFFFFF	#FCFDFF
(21,1)	4	#FFFFFF	#FCFDFF
(22,1)	9	#FFFFFF	#FDFEFF
(23,1)	5	#FFFFFF	#FDFDFF
(24,1)	3	#FFFFFF	#FFFDFF
(25,1)	4	#FFFFFF	#FCFDFF
(1,2)	B	#FFFFFF	#FFFEFF
(2,2)	5	#FFFFFF	#FDFDFF
(3,2)	2	#FFFFFF	#FDFDFF
(4,2)	4	#FFFFFF	#FCFDFF
(5,2)	9	#FFFFFF	#FCFEFF
(6,2)	5	#D1CDCE	#D1CDCE
(7,2)	4	#656465	#646566
(8,2)	-	#191112	#191112
(9,2)	-	#020000	#020001

Setelah gambar 4.1 berdasarkan tabel 1.1, dihasilkan gambar seperti pada gambar 4.2. Jika kita lihat dengan mata telanjang, tidak ada perbedaan antara keduanya, namun data intensitas warnanya berbeda.



Gambar 4.2 Hasil proses gambar ketika sudah disisipkan pesan

V. PROSES *DECODING*

Proses *decoding* dari kriptografi ini sebenarnya sangat mudah. Kita dapat membaca nilai modulo 4 dari masing-masing piksel untuk menentukan nilai heksadesimal yang sudah dimasukan sebelumnya berdasarkan tabel 3.1. Pembacaan ini dilakukan hingga akhir piksel ataupun hingga $C_B^n \ mod\ 4 \neq C_B^{n+1} \ mod\ 4$ (dengan n bilangan ganjil). Kemudian masing nilai heksadesimal ini dapat disusun menjadi huruf ASCII.

Namun salah satu kelemahan dari kriptografi ini, gambar yang sudah disisipkan pesan tidak dapat dikembalikan seperti semula. Namun hal ini tidak terlalu bermasalah karena pengembalian gambar ke semula tidak terlalu perlu.

Dari penjelasan itu, dapat ditulis ke dalam *pseudocode* menjadi :

```
w = width(img)
h = height(img)
wpos = 1
hpos = 1
i = 0
while ((hpos<=h) and (sync)) do
 HEX1 = ConvHex (img.color[wpos,hpos])
 B1 = img.colorB[wpos,hpos]
 if wpos=w then
  wpos = 1
  hpos++
 else
  wpos++
 sync = (B1==img.colorB[wpos,hpos])
 HEX2 = ConvHex(img.color[wpos,hpos])
 if (sync) then
  msg[i] = ConvASCII(HEX1,HEX2)
  i++
 if wpos=w then
  wpos = 1
  hpos++
 else
  wpos++
```

VI. KEKURANGAN SISTEM

Sistem ini memiliki beberapa kekurangan, yaitu :

- Setelah gambar disisipi dengan kode tertentu, gambar tidak dapat dikompresi karena menyebabkan perubahan warna pada gambar, sehingga ukuran gambar akan membesar.
- Setelah gambar disisipi dengan kode tertentu, gambar tidak dapat kembali ke asal.

VII. KESIMPULAN

Dalam menyipkan suatu pesan ke dalam gambar, ita dapat memanfaatkan nilai modulo intensitas warna pada gambar. Kita dapat menggeser warna pada masing-masing sehingga nilai modulonya merepresentasikan nilai heksadesimal yang kemudian digabungkan dan dikonversi menjadi karakter ASCII. Karena penggeseran warna ini sangat kecil, perbedaan yang muncul tidak cukup signifikan, sehingga tidak memunculkan kecurigaan seseorang bahwa ada pesan dibalik gambar tersebut.

VII. LAMPIRAN

A. *ASCII Table*

Hex	Glyph	Hex	Glyph	Hex	Glyph
20	␣	40	@	60	`
21	!	41	A	61	a
22	"	42	B	62	b
23	#	43	C	63	c
24	$	44	D	64	d
25	%	45	E	65	e
26	&	46	F	66	f
27	'	47	G	67	g
28	(	48	H	68	h
29	)	49	I	69	i
2A	*	4A	J	6A	j
2B	+	4B	K	6B	k
2C	,	4C	L	6C	l
2D	-	4D	M	6D	m
2E	.	4E	N	6E	n
2F	/	4F	O	6F	o
30	0	50	P	70	p
31	1	51	Q	71	q
32	2	52	R	72	r
33	3	53	S	73	s
34	4	54	T	74	t
35	5	55	U	75	u
36	6	56	V	76	v
37	7	57	W	77	w
38	8	58	X	78	x
39	9	59	Y	79	y
3A	:	5A	Z	7A	z
3B	;	5B	[	7B	{
3C	<	5C	\	7C	|
3D	=	5D	]	7D	}
3E	>	5E	^	7E	~
3F	?	5F	_		

VII. UCAPAN TERIMA KASIH

Dalam penulisan makalah ini penulis menyampaikan ucapan terima kasih yang tak terhingga kepada pihak-pihak yang membantu dalam menyelesaikan penelitian ini, khususnya kepada :

1. Ibu Dra. Harlili S., M.Sc. dan Dr. Ir. Rinaldi Munir selaku dosen mata kuliah Struktur Diskrit
2. Rekan-rekan semua Teknik Informatika angkatan 2011 yang telah membantu memberikan dorongan dan ide dalam pembuatan makalah ini.
3. Secara khusus penulis menyampaikan terima kasih kepada keluarga tercinta yang telah memberikan dorongan dan bantuan serta pengertian yang besar kepada penulis, baik selama mengikuti perkuliahan maupun dalam menyelesaikan makalah ini
4. Semua pihak yang tidak dapat disebutkan satu persatu, yang telah memberikan bantuan dalam penulisan makalah ini.

PUSTAKA

[1] Jonathan B. Knudsen, *Java Cryptography*, California : O'Reilly, 1998

[2] Munir, Rinaldi.2009.*Matematika Diskrit*. Bandung : Informatika

[3] Ms. Kiran Kumari dan Prof. Shalini Bhatia, "*Multi-pixel Visual Cryptography for color images with Meaningful Shares*", International Journal of Engineering Science and Technology, Volume II No. 6, 2010

[4] Moni Naor dan Adi Shamir, "*Visual Cryptography"*, Eurocrypt, 1994

[5] Xiaoyu Wu1, Duncan S. Wong2, and Qing Li, "*Threshold Visual Cryptography Scheme for ColorImages with No Pixel Expansion*", Huangsan, China, Desember 2009.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 18 Desember 2012

Ignatius Evan Daryanto (13511019)