Implementation of Steganographic Algorithm Based on Pixel Value Differences (PVD) Method Using Matlab

Bayu Kumoro Yakti[1], Ragiel Hadi Prayitno[1] dan Sunny Arief Sudiro[2]

[1]Fakultas Teknologi Industri Universitas Gunadarma, Jakarta Indonesia
[2]STMIK Jakarta STI&K, Jakarta Indonesia
E-mail : {bayuyakti, ragielhp}@staff.gunadarma.ac.id, sunnyariefsudiro@ieee.org

Abstrak

Gambar digital adalah gambar dalam bentuk format digital atau media digital seperti hard drive. Gambar digital terdiri dari bit (0 atau 1) yang disebut piksel dan memiliki kapasitas tinggi untuk menyimpan data dan informasi. Keamanan itu penting, terutama saat mengirim data dari satu tempat ke tempat lain. Salah satu cara untuk mengamankan data adalah melalui steganografi. Steganografi adalah teknik yang digunakan untuk menyembunyikan keberadaan informasi rahasia di dalam suatu objek. Teknik Steganografi dengan sempurna menutup pesan rahasia dalam gambar pembawa keamanan tinggi. Informasi dan data akan dimanipulasi sehingga tidak dapat dideteksi oleh mata manusia. Teknik Pixel Value Differences (PVD) merupakan salah satu metode steganografi. Metode ini menyisipkan pesan rahasia berdasarkan perbedaan bit antara dua piksel yang bertetangga. Pesan rahasia dalam bentuk teks akan disebar ke seluruh gambar dalam tingkat keabu-abuan. Hasil penelitian menunjukkan bahwa pesan yang berjumlah 6 karakter dan 37 karakter memiliki hasil: 0,0459 untuk MSE dan 61,5113 untuk PSNR (6 karakter) dan 0,4099 untuk MSE dan 52.004 untuk PSNR (37 karakter).

Kata kunci : algorithm, MATLAB, PVD, steganographi

Abstract

Digital images are images in the form of digital formats or digital media such as hard drives. Digital images consist of bits (0 or 1) called pixels and have a high capacity to store data and information. Security is important, especially when sending data from one place to another. One way to secure data is through steganography. Steganography is a technique used to hide the existence of confidential information inside an object. The Steganography Technique perfectly closes the secret message in a high-security carrier image. Information and data will be manipulated so that it cannot be detected by the human eye. Pixel Value Differences (PVD) technique is a method of steganography. The method inserts a secret message based on bit differences between two neighboring pixels. Secret messages in the form of text will be spread out onto the grayscale image. The results showed that the messages amounted to 6 characters and 37 characters had results: 0.0459 for MSE and 61.5113 for PSNR (6 characters) and 0.4099 for MSE and 52.004 for PSNR (37 characters).

Keywords : algorithm, MATLAB, PVD, steganography

DOI : http://dx.doi.org/10.32409/jikstik.19.4.357

Introduction

The internet has developed so rapidly that most individuals often use the internet as the main media to transfer data from a place to another. there are a lot of ways to send data using the internet, including via e-mail, chat, cloud. Data transmission is made simple, fast, and accurate using the internet. However, one of the main problems with sending data over the internet is a security threat, i.e. private or confidential data can be stolen or hacked in various ways. Therefore,it is very important to consider the method that will be used to secure the following data transmitted data output, because data security is one of the most important factors that need to be considered in the data transfer process.

Data security basically means protecting data from unauthorized users or hackers and attempts to prevent data modification. This area of data security has received a lot of attention over the last period of time due to the large increase in data transfer speeds over the internet. To improve security features in data transfer via the internet, many techniques have been developed, such as Cryptography, Steganography, and digital watermarking. Cryptography is a method for hiding information by encrypting information into "coded text" and transmitting it to the intended recipient, using an unknown key. Steganography provides further security by hiding ciphertext into invisible images or in other formats. Cryptography uses unreadable data output. Because of this output, cryptography is easily suspected by hackers, while the output of steganography does not look like a cipher so steganography is used more often. [1]

Pixel Value Differences

Steganography is a secret message data method that is inserted in the cover data. So steganography has two inputs namely secret message data and cover data. Secret messages can be in the form of text, images, sound or video [2]. Cover data can also be the same thing as a secret message in the form of text, images, sound or video. In steganography systems, secret message data and cover data can be done with different data such as: secret text messages inserted in the image cover data, audio secret messages inserted in the image cover data, secret picture messages inserted into the image cover data and so on.

In this study, the method used for data transmission security is Pixel Value Differences (PVD). This PVD method uses the value of the difference between two consecutive pixels blocked to determine how many secret bits must be embedded. There are two types of quantization range tables in Wu and Tasi's methods. The first is based on choosing a wide range (8, 8, 16, 32, 64, 128), to provide large capacity. The second is based on choosing a wide range (2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, 64), to provide high imperceptibility. Most of the related research focuses on capacity building using LSB and the adjustment process, so that their approach is too aligned with the LSB approach [3].

There is little research that focuses on reach design. In addition, it is intuitive to design using two power widths. This work designs a new quantization range table based on the perfect square number to decide on the payload with the value of the difference between consecutive pixels. Our research provides a new perspective that if we choose the right width for each range and use the proposed method, we can get a better number of images and a higher capacity. [3] In this study, the input used is a secret message in the form of text and images as a cover.

The purpose of this study is the analysis of the PVD method. In this study, the basic algorithm in PVD will be explained and implemented using MATLAB 2019a software. in this study will be done by inputting text characters and grayscale images. The following steps will be taken.

Based on Figure 1, the input in this study is a secret message in the form of text and cover images in various formats. In secret messages, each character in the message is changed to 8 ASCII bits for insertion. Figure 2 and 3 are is the ASCII table used in this study.

Based on Figure 2, the character used is the character "!" with decimal 33 up to the character "~" with decimal 126. While figure 3 is an extension of the ASCII table. The characters in ASCII will be converted to decimal values and converted to binary values to be inserted in the image. For example, if the secret message used is 'wind' then the decimal result obtained is 98 97 121 117. Then, the decimal value is converted to binary, then the result obtained is 01100010 01100001 01111001 01110101.

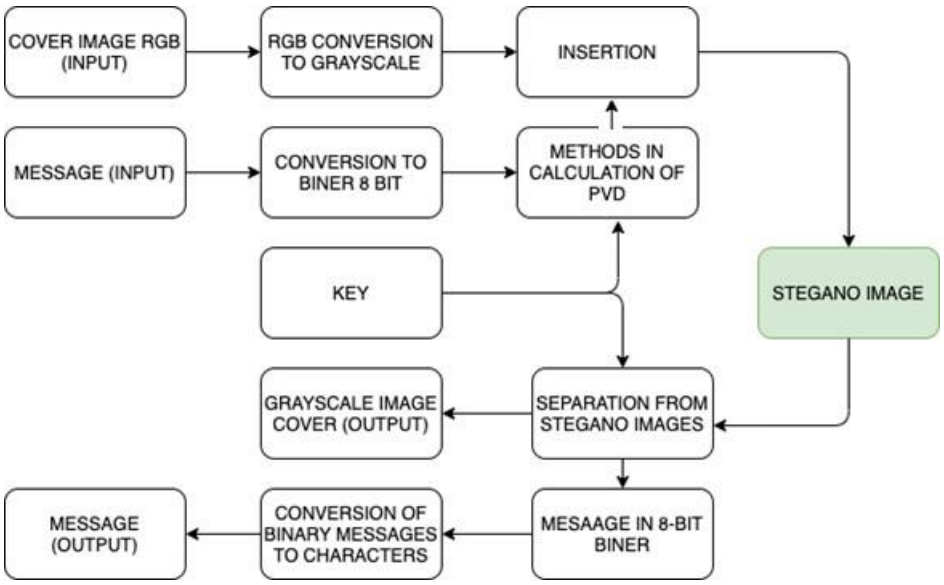

Figure 1: Research Flow

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Figure 2: ASCII Table [4]

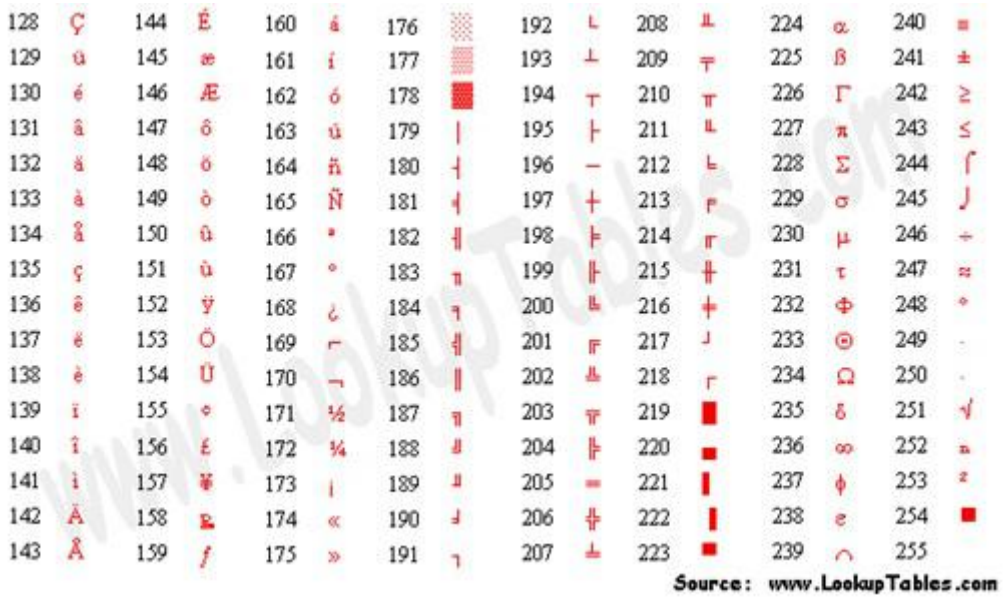

Figure 3: ASCII Table extended [4]

Each character will be converted to binary so the secret message, which originally consisted of 4 characters, turned into 32 binary data (the number of characters multiplied by 8). Because of the large number of binaries, in this study the secret message is limited to 25 characters so that the maximum binary value obtained is 200 data. Because of the size of the data, the minimum size of the image used in this study was 200 pixels [5].

After the secret message process is converted into finished bits, proceed with the cover image process which has a resolution of 1280x720 pixels. The cover image used is the RGB image as shown in Figure 3.

Figure 4: Cover image

From the RGB image is converted into a grayscale image so that it has a value of 0 to 255 where the value of 0 is black and the value of 255 is white. Grayscale images use 8bit integers [3]. From the value results of the cover image (grayscale), the image is made as a 1-row matrix. Then the insertion process continues. Following is the insertion algorithm used [6]:

1. Enter the message in text form and declaration as a string
2. Change the secret message to ASCII decimal
3. Change the message to 8 binary bits
4. Insert the cover image
5. Change the cover Figure to grayscale so that the pixels obtained range from 0-255
6. Change the cover Figure into a row vector
7. The key used in this study is the gray range values proposed by Wu and Tsai
8. Calculate the difference between 2 neighboring pixels (gi, g (i + 1): with the equation

$$di = g(i+1) – gi \quad (1)$$

9. Determine the lower limit (ik) with the number of bits n, with the equation:

$$ik \leq di < i(k+1) \quad (2)$$

10. Take the message as many as n bits, then change it to decimal (b)

11. Calculate the new value difference using the equation:

$$d' = ik + b \quad jika \quad d \geq 0 \tag{3}$$

$$d' = -(ik + b) \quad jika \quad d < 0 \tag{4}$$

12. Calculate using the equation:

$$m = d'–d \tag{5}$$

13. Calculate the new pixel value using the equation:

    If m is odd, then:

$$[gi - m/2], [g(i + 1) + m/2] \tag{6}$$

    If m is even, then:

$$[gi + m/2], [g(i + 1) - m/2] \tag{7}$$

14. At$[gi-m/2]or[g(i+1)-m/2], m/2$ rounding up, for example: 5,3=6

15. $At[g(i+1)+m/2]or[g(i+1)+m/2]$, m/2 rounding down, for example: 5,3 = 5

16. The Image row vector is returned to the image matrix

17. Show the stegano image

18. Perform MSE and PSNR calculations

For example, the calculation is as follows:

1. A message '154' will be inserted into the 8-bit grayscale 8-bit cover Figure

2. Figure 5 is the matrix used

105	200	57	28
178	145	18	23
143	211	54	68
153	174	58	98

Figure 5: Example of a Cover iamge matrix

3. Message = 154 = 1 0 0 1 1 0 1 0

4. d = 200-105 = 95

5. $64 \leq d \leq 127$(refer to table 1), ik = 64, n = 6

6. Message = 1 0 0 1 1 0 1 0, Takes 6 bit, b = 1 0 0 1 1 0 = 38 (decimal)

7. $d \geq 0$, then the new difference: d' = ik + b = 64 + 38 = 102

8. Calculate: m = d' – d = 102 – 95 = 7

9. Because m is odd, carried out the equation (6) then: [gi - m/2], [g(i+1) + m/2] = [105 - 7/2], [200 + 7/2] = [105 - 4], [200 + 3] = 101, 203

10. Inserted at the 1st and 2nd pixels, so Figure 6 is the result:

11. For further insertion in the remaining binary i.e. 0 (can be seen in steps 3 and 6) the same calculation is performed and inserted at pixels 3 and 4, so it can be seen the results in Figure 7.

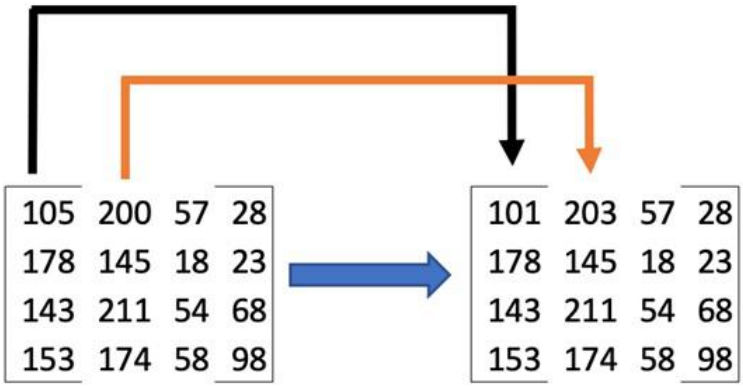

Figure 6: Result of pixel insertion 1 and 2

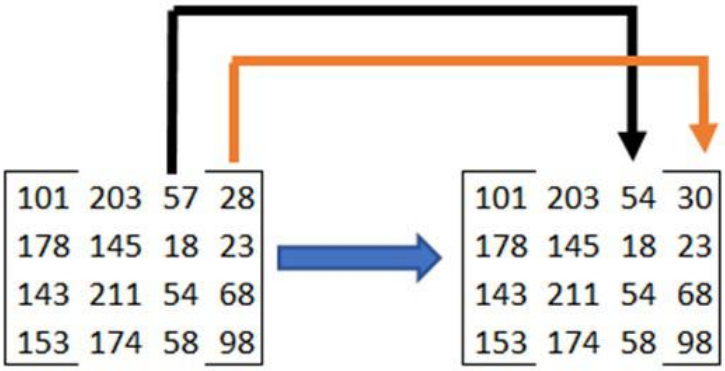

Figure 7: Result of pixel insertion 3 and 4

After the insertion process is completed the MSE and PSNR calculations are performed.Peak Signal to Noise Ratio (PSNR) is the ratio between the maximum value of thesignal measured by the amount of noise that affects the signal. PSNR is usually measured in decibels(db). PSNR is used to compare the quality of the cover image before and after the message is entered. To determine the PSNR, The MSE (Mean Square Error) value must be known in advance.

MSE is the average error value between the original image and the manipulated image (in the case of steganography; MSE is the average error value between the cover image and the stego image). PSNR is often expressed on a logarithmic scale in decibels (dB). PSNR values that fall below 30 dB indicate a relatively low quality, where distortion due to insertion is clearly visible. However, high stego image quality is found at values of 40dB and above. [7]

In ordinary view, the best value in an image shows that the human eye barely recognizes the original image and steganographic image. A higher PSNR value means a closer similarity between the results of the reconstruction and the original image. PSNR is defined as:

$$PSNR = 10log10((C^2_{MAX})/mse) \quad (8)$$

The lower the MSE, the lower the error produced. Where MSE is stated as:

$$MSE = \frac{1}{MN}\sum_{X=1}^{M}\sum_{X=1}^{N}(S_{XY} - C_{XY})^2 \quad (9)$$

Following is the extraction algorithm used [6]:

1. Insert the stegano image
2. Change the stegano image into a 1 row matrix
3. Calculate the difference between 2 neighboring pixels (gi, g (i + 1)) so that: di = g (i + 1) – gi
4. Determine the lower limit (ik) with the number of bits n, by: ik< di <i (k + 1) (refer to table 1)
5. Calculate: b = | d | - ik
6. Change b (decimal) to binary n bits
7. Take the Message n bit

For example, the calculation is as follows:

1. It is known that the stego shown in figure 8:

101	203	54	30
178	145	18	23
143	211	54	68
153	174	58	98

Figure 8: Pixel of Stegano image

2. d = 203 – 101 = 102
3. $64 \leq d \leq 127$(refer to table 1), ik = 64, n = 6
4. b = |102| - 64 = 38
5. b = 38 = 1 0 0 1 1 0
6. Message 6 bit = 1 0 0 1 1 0
7. Message still has 2 bit, then proceed to pixels 3 and 4
8. d = 30 – 54 = -24
9. $16 \leq |d| \leq 31$(refer to table 1), ik = 16, n = 4
10. b = |-24| - 16 = 24 – 16 = 8
11. b = 8 = 1 0 0 0
12. take the 2 bit = 1 0
13. Message = Message 6 bit and Message 2 bit = 1 0 0 1 1 0 1 0 = 154

Results and Discussion

In this section, the results of the research conducted will be explained. The object to be compared in this study is the image format using the same method as described in the introduction. The following are the provisions carried out in this study: The cover image you are going to use has a resolution of 1280x720 pixels The messages used are "gundar" and "Sphinx of black quartz, judge my vow!"

Figure 9 is the source code used in the research. Figure 9 shows the program input process message and picture input. The process starts from inputting message variables. These variables can be entered in the form of text messages and entered as a string. After that the message length is calculated. The next variable is the cover image input variable. The cover image is inputted and then converted to a grayscale image.

```
%-----------------------------------
% PROGRAM STEGANOGRAFI METODE PVD
%-----------------------------------

clc;
clear all;

Pesan = input('Masukkan Pesan = ', 's');
Pesan=uint8(Pesan);              %Pesan teks dijadikan angka
panjang_Pesan = length(Pesan);   %hitung panjang pesan

citra = imread('1280x720.jpg');      %baca citra penampung
if size (citra, 3)==3            %jika citranya RGB, jadikan grayscale
    citra=rgb2gray(citra);
end

[baris ,kolom]=size(citra);
stego=citra(:);                  %matrik citra dijadikan 1 baris
panjang_stego=length(stego);

%pesan dijadikan biner
bit_pesan=[];
for i=1:panjang_Pesan
    biner=dec2bin(Pesan(i), 8);
    bit_pesan=[bit_pesan biner];
end

panjang_bit_pesan=length(bit_pesan);
ambil_bit_pesan=[];
n=0;
```

Figure 9: Message and image input process

```
31
32      %penyisihan pesan
33 -    for i=1:2:panjang_stego
34 -        d=stego(i+1)-stego(i);
35 -        if 0<=d<=7; Lk=0; n=3; end
36 -        if 8<=d<=15; Lk=8; n=3; end
37 -        if 16<=d<=31; Lk=16; n=4; end
38 -        if 32<=d<=63; Lk=32; n=5; end
39 -        if 64<=d<=127; Lk=64; n=7; end
40 -        if 128<=d<=255; Lk=128; n=7; end
41 -        if n> length(bit_pesan)
42 -            n=length(bit_pesan);
43 -            break
44 -        end
45 -        ambil_bit_pesan=bit_pesan(1:n);
46 -        bit_pesan=bit_pesan(n+1:end);
47
48 -        b=bin2dec(ambil_bit_pesan);
49 -        if d>=0; d1=Lk+b; else d1=-(Lk+b); end
50 -        m=d1-d; bawah=floor(m/2); atas=ceil(m/2);
51 -        if mod(m,2)==1, stego(i)=stego(i)-bawah;
52 -            stego(i+1)=stego(i+7)+bawah;end
53 -    end
54 -    stego=reshape(stego, [baris kolom]); %citra baris dijadikan matriks
55
56 -    figure,
57 -    imshow(citra),title('Asli');
58
59 -    figure,
60 -    imshow(stego),title('Stego');
61
```

Figure 10: Insertion process

```
62      %ekstraksi
63 -    bit_pesan=[];
64 -    ambil_bit_pesan=[];
65 -    for i=1:2:panjang_stego
66 -        if length(bit_pesan)==panjang_bit_pesan,
67 -            break, end
68 -        d=stego(i+1)-stego(i);
69 -        if 0<=d<=7; Lk=0; n=3; end
70 -        if 8<=d<=15; Lk=8; n=3; end
71 -        if 16<=d<=31; Lk=16; n=4; end
72 -        if 32<=d<=63; Lk=32; n=5; end
73 -        if 64<=d<=127; Lk=64; n=7; end
74 -        if 128<=d<=255; Lk=128; n=7; end
75 -        b=abs(d)-Lk;
76 -        ubah_b=dec2bin(b);
77 -        bit_pesan=[bit_pesan ubah_b];
78 -    end
79 -    pesan2=[];
80 -    for i=1:8:panjang_bit_pesan
81 -        a=char(bin2dec(bit_pesan(1:8)));
82 -        pesan2=[Pesan a];
83 -    end
84 -    pesan2=pesan2(1:end-1)
85
86 -    citra = double(citra); stego = double(stego);
87 -    mse = sum((citra(:)-stego(:)).^2) / prod(size(citra));
88 -    psnr = 10*log10(255*255/mse);
```

Figure 11: Extraction process, MSE and PSNR

Furthermore, the matrix is made into 1 row to simplify the PVD method process. The message that has been typed will be converted to 8 binary bits. Figure 10 shows the PVD method insertion process according to Wu and Tsa. Then, the image is returned to the image matrix. The resulting cover image and stegano image are displayed.

Figure 11 shows the process of extracting the PVD method between stegano and key images to produce cover images and secret messages, which are :

1. Research results on the "Gundar" message When the program starts, the user will be asked to enter a secret message. The secret message entered is "gundar". Figure 12 is the data or results that appear in the MATLAB command window:

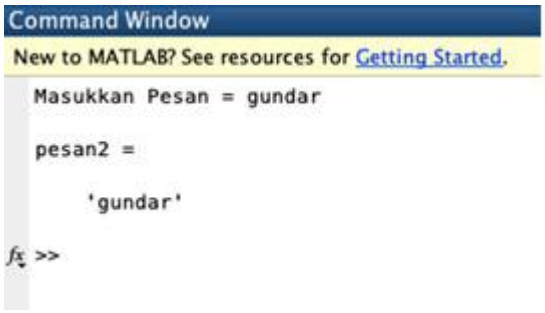

Figure 12: Message for “gundar”

Figure 13 is the result of the image in MATLAB:

Figure 13: Cover Image for “gundar”

Figure 14 is a stegano image with the message "Gundar".

Figure 14: Stegano Image for “gundar”

Figure 15 is the results of the data obtained.

Figure 15: Results on the MATLAB workspace for “gundar”

In Figure 15 is the result of input and output in MATLAB. The results obtained if the secret message "Gundar" is 0.0459 for MSE and 61.5113 for PSNR. The results obtained are good because the lowest limit on the PSNR is 30.

2. Research results on the message "Sphinx of black quartz, judge my vow!" When the program starts, the user will be asked to enter a secret message. The secret message entered was "Sphinx of black quartz, judge my vow!". Figure 16 is the data or results that appear in the MATLAB command window:

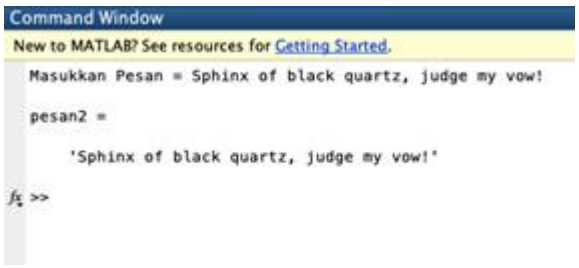

Figure 16: Message

Figure 17: Cover Image

Figure 18: Stegano Image

Figure 17 is the result of the image in MATLAB. Figure 18 is a stegano image with the message "Sphinx of black quartz, judge my vow!"

Here are the results of the data obtained:

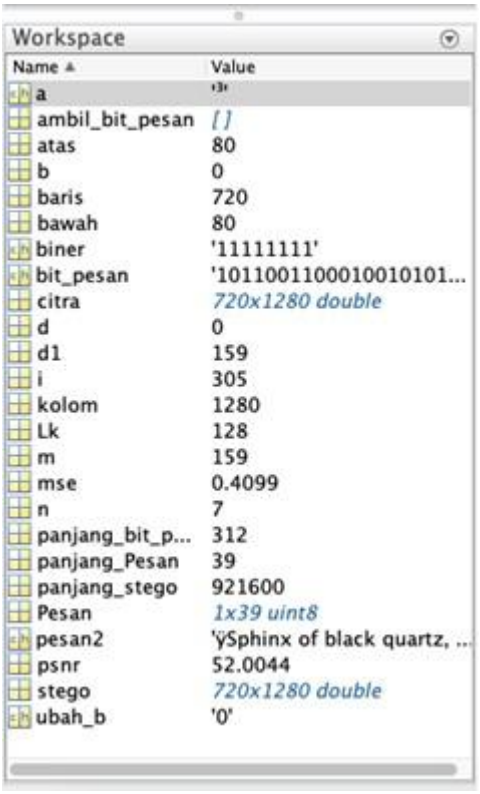

Figure 19: Results on workspace MATLAB

In Figure 19 is the result of input and output in MATLAB. The results obtained if the secret message "Sphinx of black quartz, judge my vow!" Is 0.4099 for MSE and 52,004 for PSNR. The results obtained are good because the lowest limit on the PSNR is 30.

The results of studies that have been made are better than previous studies [8]in terms of PSNR. Figure 20 are the results of previous researchers.

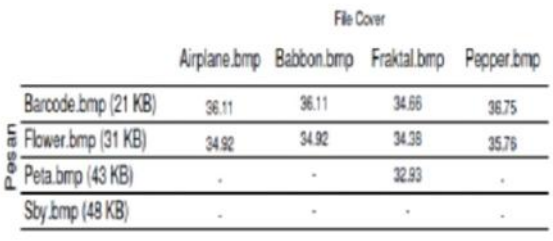

Figure 20: PSNR results of previous researchers [6]

Conclusion

In this study steganography has been processed using the PVD method. In this study conducted with RGB images which are then converted to grayscale images. The image used has a resolution of 1280x720 pixels. For secret messages, two experiments were carried out, namely secret messages totaling 6 characters "gundar" and secret messages totaling 37 characters "Sphinx of black quartz, judge my vow!".

The results on the secret message totaling 6 characters are 0.0459 for MSE and 61.5113 for PSNR. While secret messages totaling 37 characters have a result of 0.4099 for MSE and 52,004 for PSNR. The results of the 6 characters message are better because the MSE is lower and the PSNR is higher than the 37 characters message. The 6 characters message is better because the fewer characters inserted the less the pixel changes in the image. Even though, in software data in MATLAB a 6 characters message is better, the human eye still cannot see the difference between a 6 characters message and a 37 characters message that has been inserted.

This research can be done by adding variations of images or different image sizes to get output variations to be analyzed.

References

[1] Indu Nehra and Rakesh Sharma, " Review Paper On Image Based Steganography", in International Journal of Scientific & Engineering Research, Vol: 6, Issue: 6, June 2015.

[2] Jaslen Kour and Deepankar Verma, "Steganography Techniques –A Review Paper", in International Journal of Emerging Research in Management &Technology, Vol: 3, Issue : 3, May 2014.

[3] R. Rahim, "Penyisipan Pesan Dengan Algoritma Pixel Value Differencing Dengan Algoritma Caesar Cipher Pada Proses Steganografi", Jurnal TIMES, Vol : 5, No: 1, Februari, 2016.

[4] Injosoft, "Ascii Table," Injosoft, [Online]. Available: http://www.asciitable.com/. [Accessed 12 April 2020].

[5] E. Sinduningrum dan Anton Supriyanto, "Perancangan Aplikasi Steganografi Berbasis Android dengan Metode Pixel Value Differencing (PVD)", in JURNALMULTINETICS, Vol: 2, No: 2, November 2016.

[6] Pulung Nurtantio Andono, "Pengolahan Citra Digital", Penerbit ANDI, 2017.

[7] M. Azzami, "Pengembangan Aplikasi Steganografi Pixel Value Differences (PVD)", in Prosiding Seminar Sains dan Teknologi, Universitas Muhammadiyah Jakarta, November 2014.

[8] R. Kumar, G. Sharma, G. and V. Sanduja, "A Real Time Approach to Compare PSNR and MSE Value of Different Original Images and Noise (Salt and Pepper, Speckle, Gaussian) Added Images", International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), Vol: VII, Issue: I, Punjab, India, January 2018.