Perancangan Aplikasi Steganografi Berbasis *Matrix Pattern* dengan Metode *Random Blocks*

Andri[1], Ali Akbar Lubis[2], Andy Angkasa[3], Haryono Angkasa[4]
STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789
[1]Jurusan Manajemen Informatika, STMIK Mikroskil, Medan
[2,3,4]Jurusan Teknik Informatika, STMIK Mikroskil, Medan
[1]andri@mikroskil.ac.id, [2]ali.akbar@mikroskil.ac.id, [3]101110141@students.mikroskil.ac.id,
[4]101110320@students.mikroskil.ac.id

Abstrak

Dengan perkembangan komputer yang digunakan dalam berbagai bidang kehidupan dan pekerjaan, masalah keamanan informasi yang dikirim menjadi semakin penting. Namun informasi yang dilindungi belum tentu bisa terproteksi dengan aman oleh sistem. Dapat terjadi pembobolan pada berbagai media pengiriman informasi. Berdasarkan masalah yang dihadapi tersebut dilakukan perancangan sebuah aplikasi steganografi guna untuk melindungi informasi yang akan dikirim. Steganografi merupakan seni penyimpanan pesan rahasia dengan menggunakan media digital seperti teks, citra, suara dan video. Rancangan aplikasi menggunakan sebuah algoritma yang proses penyisipannya dengan menentukan perkalian matriks dan hanya menggunakan beberapa blok untuk penyisipan yaitu Matrix Pattern. Pemilihan blok secara acak dengan menggunakan LCG. Hasil pengujian menunjukkan bahwa kecepatan, keamanan dan kualitas citra hasil steganografi bergantung pada dimensi citra, berapa banyak karakter yang diinput dan pemilihan ukuran matrix pattern. Kriteria untuk mendapatkan stego-image yang bagus yaitu pemilihan matrix pattern yang paling kecil, jumlah pesan yang disisipkan sedikit dan pemilihan media citra penyisipan yang memiliki dimensi yang besar. Perubahan warna atau pixel citra yang telah berisi pesan rahasia dapat merusak pesan sehingga tidak dapat diekstrak kembali.

Kata kunci— *Steganografi, Mattrix Pattern, Citra Digital*

Abstract

With the development of computers that are used in various fields of life and work, transmission of information security issues are becoming increasingly important. However, the information may not necessarily be protected safely protected by a system which can occur burglary in various information delivery media. Based on the problems, we design a steganography application in order to protect the information to be sent. Steganography is the art of secret message store using digital media such as text, images, sound and video. The design of the application uses an algorithm to determine the insertion process matrix multiplication and only use a few blocks to the insertion ie Matrix Pattern. The selection of random blocks using LCG.The results show that the speed, security and quality of output image depend on the dimensions of the image steganography, how many characters are inputted and the selection pattern matrix size. The criteria to get a good stego-image depends on the choice of most small pattern matrix, number of messages inserted slightly and selecting images for insertion has large dimensions. The change of color or pixel image that already contains a secret message can undermine the message that it can not be extracted back.

Keywords— *Steganography, Mattrix Pattern, Digital Image*

1. PENDAHULUAN

Dewasa ini seni mengirim dan menampilkan informasi yang tersembunyi terutama di tempat – tempat umum, telah menerima lebih banyak perhatian dan tantangan [1]. Dengan perkembangan komputer yang digunakan dalam berbagai bidang kehidupan dan pekerjaan, masalah keamanan informasi yang dikirim menjadi semakin penting. Namun, pesan rahasia yang dikirim bisa dibajak oleh orang yang ingin mengetahui isi dari pesan rahasia dan penyimpanan pesan rahasia harus memiliki tingkat transparansi secara visual yang tinggi agar tidak menimbulkan kecurigaan. Oleh karena itu, dibutuhkan aplikasi yang mendukung untuk mengamankan informasi atau pesan rahasia tersebut.

Steganografi merupakan seni penyimpanan pesan rahasia dengan menggunakan media digital seperti teks, citra, suara dan video. Data yang tersembunyi didalam steganografi merupakan hal yang sulit untuk dideteksi dan bila dikombinasikan dengan algoritma yang cocok maka akan lebih sulit untuk diuraikan [2]. Ada berbagai macam metode untuk steganografi, salah satunya adalah metode yang berdasarkan pada *Matrix Pattern* dengan *block – block* yang dipilih secara acak menggunakan *Pseudo-Random Number Generator* atau disebut metode *Random Blocks* [3], metode *Discrete Cosine Transformation (DCT)* yang mengubah gambar dari domain spasial ke domain frekuensi [4]. Beberapa algoritma yang mendukung pembentukan aplikasi steganografi yaitu algoritma *Least Significant Bit (LSB)* yang merupakan suatu algoritma yang memiliki paling sedikit perubahan pada suatu citra [5], algoritma LSB *Substitution by Minimize Detection* [6], dan algoritma F5 dengan menyisipkan bit data pesan kedalam bit koefisien [7]. Penelitian ini bertujuan untuk menghasilkan sebuah aplikasi steganografi dengan menggunakan metode *Random Blocks* berdasarkan pada *Matrix Pattern* dan pengunaan algoritma *Least Significant Bit* (LSB). Manfaatnya adalah aplikasi dapat mengamankan pesan rahasia yang telah disisipkan pada gambar.

Ruang lingkup penelitian ini sebagai berikut:

a. Pesan yang disisipkan berupa angka (0 - 9), teks yang berformat *lower case* (a - z), dan beberapa simbol yaitu koma (,), titik (.), petik ("), tambah (+), kurang (-), kali (*), bagi (/), tanda tanya (?), buka kurung, tutup kurung, simbol dan (&), dan spasi.
b. *Image cover* yang digunakan berbasis JPEG, PNG, GIF atau BMP dengan ukuran lebar dan tinggi kelipatan 60 pixel tetapi tidak diharuskan simetris.
c. Kunci yang digunakan berupa angka dengan minimal 1 digit dan maksimal 8 digit.
d. Ukuran blok dibatasi dengan ukuran 60x60 pixel.

Langkah-langkah untuk menyelesaikan penelitian ini antara lain:

a. Identifikasi Masalah
   Dilakukan identifikasi terhadap permasalahan yang ada. Kemudian mencari solusi yang mendukung pada permasalahan yang dihadapi.
b. Pengumpulan Data dan Sumber Pendukung (Literatur)
   Pengumpulan literatur yang mendukung penelitian dilakukan pada tahap ini. Literatur – literatur diambil dari jurnal – jurnal ilmiah.
c. Desain
   Pembuatan desain aplikasi steganografi.
d. Konstruksi Program
   Penulisan kode program dengan bahasa Visual Basic 2010.
e. Pengujian
   Dilakukan pengujian hasil dengan melihat *Peak Signal to Noise Ratio (PSNR)* untuk mengetahui kualitas citra setelah melakukan proses *embedding*.

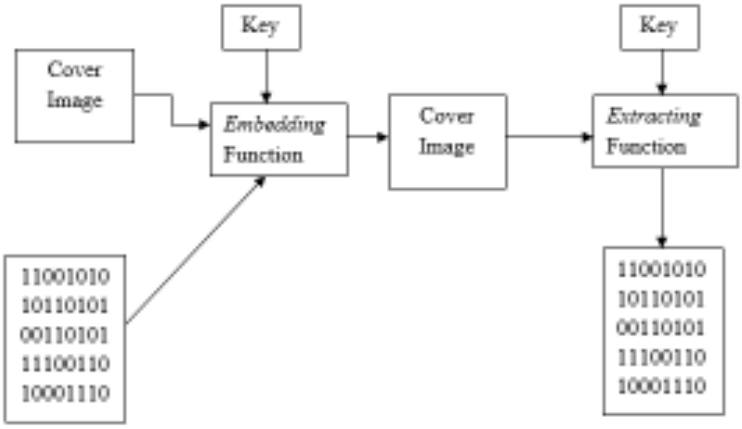
2. KAJIAN PUSTAKA

2.1 *Steganografi*

Steganografi merupakan seni menyembunyikan informasi penting atau pesan rahasia kedalam media digital seperti teks, citra, suara dan video. Kata steganografi (*steganography*) berasal dari

bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis, sehingga kurang lebih artinya adalah "*menulis tulisan yang tersembunyi atau terselubung*". Teknik ini meliputi banyak sekali metode komunikasi untuk menyembunyikan pesan rahasia. Pesan rahasia yang tersembunyi didalamnya memiliki perubahan yang sangat sedikit. Informasi atau pesan rahasia yang tersembunyi dapat berupa *plaintext*, *ciphertext*, gambar atau apapun yang dapat dimasukkan kedalam aliran bit [8]. Pesan rahasia yang berupa teks atau citra tersebut akan disimpan dalam citra yang disebut *image cover*.

Proses steganografi dapat dilihat pada Gambar 1 berikut. Secara garis besar, teknik penyembunyian data dengan steganografi adalah dengan cara menyisipkan sepotong demi sepotong informasi asli pada sebuah media, sehingga informasi tersebut tampak kalah dominan dengan media pelindungnya.



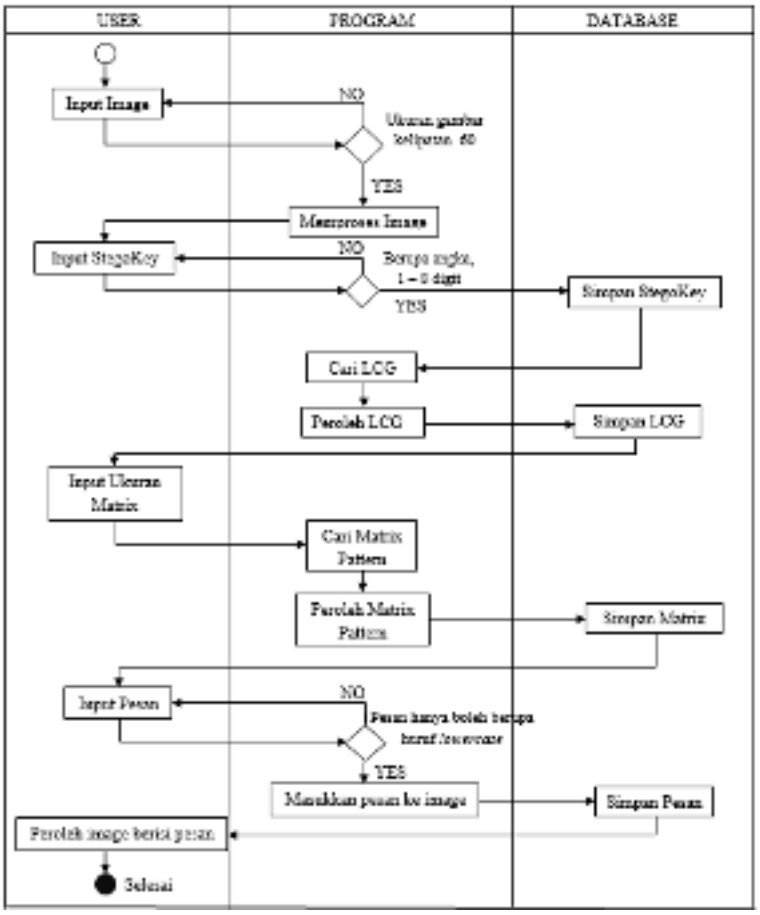
Gambar 1. Proses Steganografi [8]

2.2 Metode *Matrix Pattern*

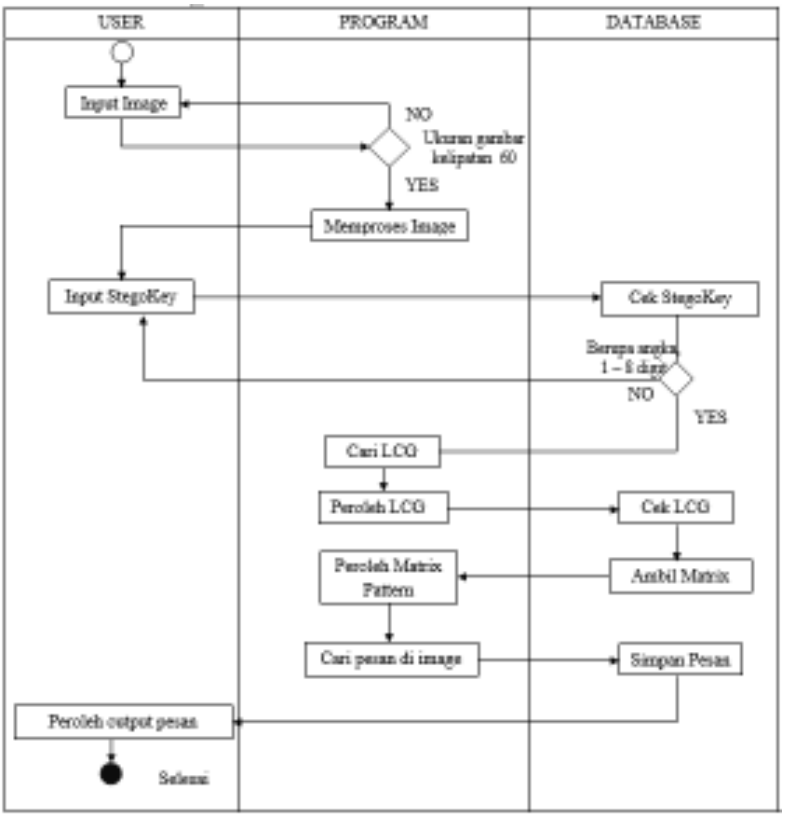
Setelah diperoleh blok - blok yang telah dipilih secara acak berdasarkan kunci, pesan rahasia akan diterjemahkan ke "*Matrix Pattern*" [3]. Berbeda dengan algoritma LSB yang biasa yaitu mengubah seluruh isi dari gambar sehingga tidak akan terlihat kejanggalan secara visual, pemanfaatan *matrix pattern* akan menghasilkan pola citra yang memiliki perbedaan warna yang terlihat secara visual. Untuk mengatasi masalah ini, akan ditetapkan 49 Blok *matrix pattern* yang acak tetapi bersifat unik. Blok yang dihasilkan akan dipilih berdasarkan tekstur blok yang telah terpilih berdasarkan informasi citra, sehingga akan berubah dari blok ke blok. 48 dari 49 Blok *matrix pattern* di atas akan ditugaskan sebagai karakter *keyboard* yang terdiri dari 26 karakter inggris, 10 angka dan 12 karakter *keyboard* khusus. 1 Blok *Matrix Pattern* yang tersisa akan dijadikan sebagai penanda akhir dari pesan rahasia.

3. METODE PENELITIAN

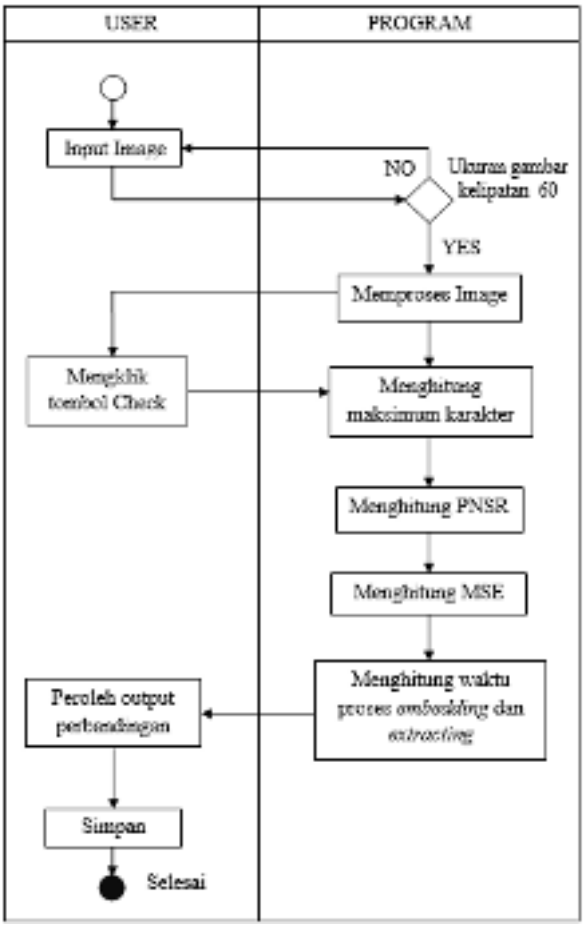
Cara kerja desain aplikasi meliputi proses *embedding* (Gambar 2), *extracting* (Gambar 3) dan pengujian PSNR (Gambar 4). Analisis kebutuhan fungsional aplikasi ini dapat dilihat pada Gambar 5.



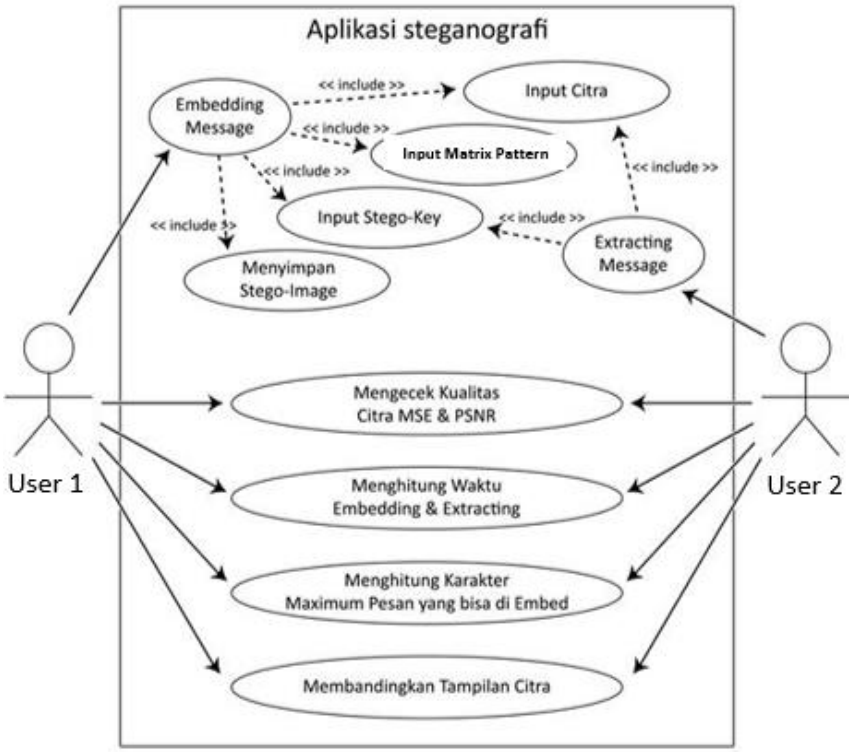
Gambar 2. Proses *Embedding*



Gambar 3. Proses *Extracting*



Gambar 4. Proses Pengujian PSNR

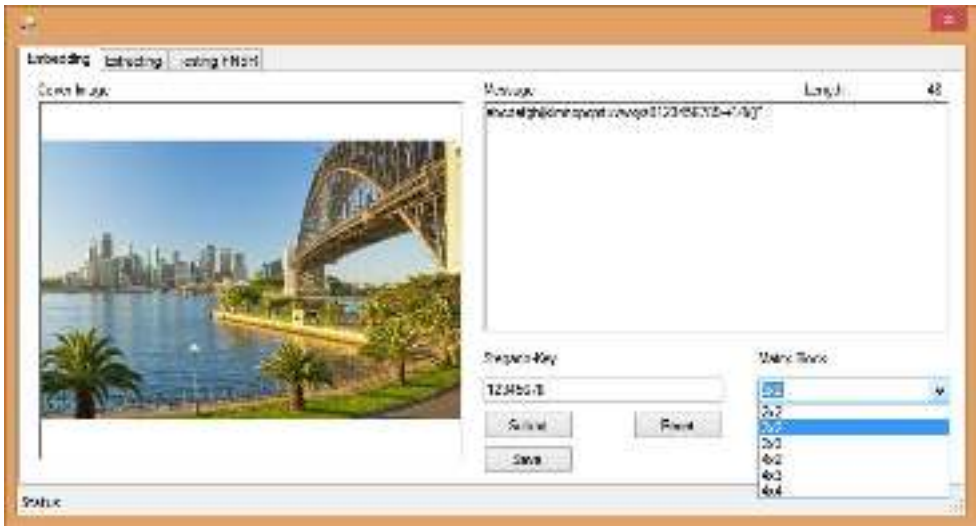


Gambar 5. Analisis Kebutuhan Aplikasi

4. HASIL DAN PEMBAHASAN

4.1. Hasil

a. Form *Embedding*
Citra dimasukkan dengan klik ganda pada bagian "*Cover Image*" seperti terlihat pada Gambar 6.



Gambar 6. Form *Embedding*

Setelah itu masukkan pesan, *Stegano-key* dan pemilihan *Matrix Block*. Lalu lakukan proses *Embedding* dengan klik tombol "*Submit*". Setelah proses selesai, citra hasil *embedding* dapat disimpan dengan klik tombol "*Save*".

b. Form *Extracting*
Digunakan untuk melakukan proses ekstraksi pesan dari citra yang berisi pesan rahasia. Agar pesan bisa diekstrasi dengan baik, kunci yang dimasukkan harus benar seperti yang terlihat pada Gambar 7.

Gambar 7. Form Ekstraksi

c. Form Pengujian PSNR
Form ini (Gambar 8) digunakan untuk membandingkan antara citra yang sebelum disisipkan pesan dengan citra setelah disisipkan pesan. Pada form ini, terdapat dua perhitungan untuk membandingkan kedua citra, yaitu MSE dan PSNR. Setelah itu ditampilkan karakter maksimum yaitu berapa banyak karakter yang dapat diisi kedalam citra dan waktu proses.

Gambar 8. Form Pengujian PSNR

4.2 Pembahasan

Pengujian dilakukan terhadap berbagai citra dengan ukuran bervariasi. Citra uji ini disisipkan dengan sejumlah karakter dengan jumlah yang bervariasi mulai dari 50, 100, 200, 400, 600 dan 1000 karakter. Pengujian yang dilakukan adalah pengaruh *matrix pattern* yang berbeda (Tabel 1), dimensi citra yang berbeda (Tabel 2), tipe file yang berbeda (Tabel 3) dan *stego key* yang berbeda (Tabel 4). Selain itu diuji juga waktu pemrosesan (Tabel 5) dan keberhasilan ekstraksi gambar hasil *embed* yang telah mengalami perubahan.

Tabel 1. Pengujian pada *Matrix Pattern* yang Berbeda

File Name	*Image Size(pixel)*	*Matrix Pattern Size*	*Message Length*	*Maximum Message*	MSE	PSNR	*Embedding Time(s)*	*Extracting Time(s)*
bridge 600x420.jpg	600x420	2x2	100	61200	4,317	54,69	0,002	0,273
bridge 600x420.jpg	600x420	3x2	100	40800	11,144	45,206	0,003	0,007
bridge 600x420.jpg	600x420	3x3	100	27200	18,788	39,982	0,004	0,008
bridge 600x420.jpg	600x420	4x2	100	30600	2,839	58,88	0,004	0,231
bridge 600x420.jpg	600x420	4x3	100	20400	44,539	31,351	0,006	0,008
bridge 600x420.jpg	600x420	4x4	100	15300	37,729	33,01	0,007	0,009

Tabel 2. Pengujian untuk Gambar dengan Dimensi yang Berbeda

File Name	*Image Size(pixel)*	*Matrix Pattern Size*	*Message Length*	*Maximum Message*	MSE	PSNR	*Embedding Time(s)*	*Extracting Time(s)*
bridge 300x180.jpg	300x180	3x2	50	7800	29,129	35,597	0,002	0,006
bridge 300x180.jpg	300x180	3x2	100	7800	78,443	25,691	0,002	0,003
bridge 300x180.jpg	300x180	3x2	200	7800	111,18	22,203	0,004	0,003
bridge 300x180.jpg	300x180	3x2	400	7800	242,967	14,385	0,007	0,004
bridge 300x180.jpg	300x180	3x2	600	7800	529,552	6,594	0,019	0,005

bridge 300x180.jpg	300x180	3x2	1000	7800	862,73	1,714	0,015	0,003
bridge 600x420.jpg	600x420	3x2	50	40800	3,005	58,312	0,006	0,006
bridge 600x420.jpg	600x420	3x2	100	40800	20,303	39,207	0,006	0,007
bridge 600x420.jpg	600x420	3x2	200	40800	33,776	34,117	0,005	0,009
bridge 600x420.jpg	600x420	3x2	400	40800	46,588	30,901	0,007	0,008
bridge 600x420.jpg	600x420	3x2	600	40800	53,845	29,454	0,009	0,009
bridge 600x420.jpg	600x420	3x2	1000	40800	147,553	19,373	0,015	0,013
bridge 900x600.jpg	900x600	3x2	50	88800	0,267	82,508	0,002	0,012
bridge 900x600.jpg	900x600	3x2	100	88800	4,893	53,438	0,003	0,012
bridge 900x600.jpg	900x600	3x2	200	88800	14,63	42,484	0,005	0,013
bridge 900x600.jpg	900x600	3x2	400	88800	30,914	35,003	0,007	0,015
bridge 900x600.jpg	900x600	3x2	600	88800	36,296	33,398	0,01	0,016

Tabel 3. Pengujian untuk Tipe File yang Berbeda

File Name	***Image Size(pixel)***	***Matrix Pattern Size***	***Message Length***	***Maximum Message***	**MSE**	**PSNR**	***Embedding Time(s)***	***Extracting Time(s)***
bridge 600x420.jpg	900x600	3x3	100	59200	5,73	51,857	0,007	0,02
bridge 600x420.bmp	900x600	3x3	100	59200	3,584	56,549	0,007	0,016
bridge 600x420.gif	900x600	3x3	100	59200	3,873	55,775	0,007	0,016
bridge 600x420.png	900x600	3x3	100	59200	7,967	48,562	0,007	0,017

Tabel 4. Pengujian untuk Panjang *Stego-Key* yang Berbeda

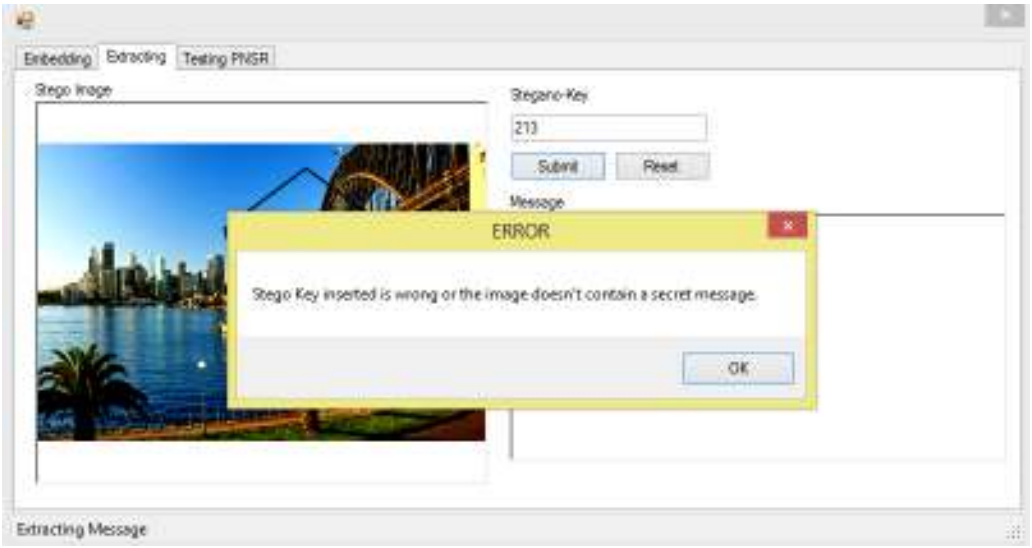
File Name	***Stego-Key Length***	***Matrix Pattern Size***	***Message Length***	***Maximum Message***	**MSE**	**PSNR**	***Embedding Time(s)***	***Extracting Time(s)***
bridge 600x420.jpg	1	3x3	100	27200	11,989	44,475	0,024	0,011
bridge 600x420.jpg	2	3x3	100	27200	17,129	40,907	0,008	0,011
bridge 600x420.jpg	3	3x3	100	27200	5,25	52,733	0,008	0,01
bridge 600x420.jpg	4	3x3	100	27200	5,149	52,926	0,007	0,011
bridge 600x420.jpg	5	3x3	100	27200	5,233	52,764	0,008	0,011
bridge 600x420.jpg	6	3x3	100	27200	9,196	47,127	0,007	0,011
bridge 600x420.jpg	7	3x3	100	27200	8,387	48,048	0,007	0,011

bridge 600x420.jpg	8	3x3	100	27200	8,387	48,048	0,007	0,011

Tabel 5. Pengujian untuk Waktu *Embedding* dan *Extracting*

File Name	***Image Size (pixel)***	***Matrix Pattern Size***	***Message Length***	***Maximum Message***	**MSE**	**PSNR**	***Embedding Time(s)***	***Extracting Time(s)***
bridge 300x180.jpg	300x180	3x2	200	7800	111,18	22,203	0,004	0,003
bridge 600x420.jpg	600x420	3x2	200	40800	33,776	34,117	0,005	0,009
bridge 900x600.jpg	900x600	3x2	200	88800	14,63	42,484	0,005	0,013
bridge 1200x840.jpg	1200x840	3x2	200	166800	0,467	76,934	0,014	0,03

Gambar 9 menunjukkan proses ekstraksi yang dilakukan pada gambar yang telah mengalami perubahan pada tingkat keabuan citra berupa penurunan *contrast*. Proses ekstraksi gagal dilakukan pada gambar karena *pixel* yang menentukan identitas dari *Matrix Pattern* dan pesan rahasia yang disisipkan telah berubah sehingga tidak dapat dibaca oleh program.



Gambar 9. Pengujian pada Gambar yang Telah Mengalami Perubahan

5. KESIMPULAN

Kesimpulan yang bisa diambil adalah:

a. Hasil *stego-image* yang baik bergantung pada ukuran *Matrix Pattern* yang kecil, pesan rahasia yang pendek dan ukuran citra yang besar untuk menyisipkan pesan rahasia.
b. Dalam proses *embedding* dan *extracting stego-image* yang cepat tergantung pada ukuran citra, *matrix pattern*, dan pesan yang disisipkan. Semakin kecil *matrix pattern* dan semakin pendek pesan akan semakin cepat proses *Embedding* dan *Extracting*.
c. Jumlah karakter maksimum yang bisa disimpan pada *stego-image* tergantung pada ukuran citra itu sendiri dan ukuran *matrix pattern* yang dipilih. Makin besar ukuran citra dan makin kecil *matrix pattern* maka karakter yang bisa dinput lebih banyak.
d. Panjang *stego-key* yang dipakai hanya berpengaruh pada tingkat keamanan steganografi agar tidak mudah dibobol.

e. *Stego-image* yang mengalami perubahan warna atau nilai *pixel* karena efek *brightness* atau *contrast* tidak dapat diekstrak kembali pesan rahasianya.

6. SARAN

Saran untuk penelitian berikutnya:

a. Melakukan pengembangan pada aplikasi agar dapat melakukan penyimpanan citra dan suara digital selain teks di dalam gambar.
b. Melakukan pengembangan agar ukuran citra yang ingin disisipkan tidak memiliki batasan ukuran perkalian lebar dan tinggi, pemilihan blok bisa dipilih oleh pengguna, serta memperbanyak ukuran perkalian *matrix pattern*.

DAFTAR PUSTAKA

[1] Channalli, S., 2009, Steganography An Art Of Hiding Data, *International Journal on Computer Science and Engineering*, Vol 1, hal 137-141.
[2] Dickman, S. D., 2007, *An Overview of Steganography*, James Madison University Infosec Techreport Department of Computer Science, http://www.infosec.jmu.edu/documents/jmu-infosec-tr-2007-002.pdf, diakses tgl 06 April 2015.
[3] Nilizadeh, A. F. dan Nilchi, A. R., 2013, Steganography on RGB Images Based on a "Matrix Pattern" using Random Blocks, *Internatonal Journal of Modern Education and Computer Science*, Vol 5, hal 8-18.
[4] Patel, H. dan Dave, P., 2012, Steganography Technique Based on DCT Coefficients, *International Journal of Engineering Research and Applications*, Vol 2, hal 713-717.
[5] Reddy, R. dan Ramani, R., 2012, The Process of Encoding and Decoding of Image Steganography using LSB Algorithm, *International Journal on Computer Science Engineering and Technology*, Vol 2, hal 1488-1492.
[6] Sharma, V. K. dan Shrivastava, V., 2012, A Steganography Algorithm for Hiding Image in Image by Improved LSB Substitution by Minimize Detection, *Journal of Theoretical and Applied Information Technology*, Vol 36, hal 1-8.
[7] Westfeld, A., 2001, F5 – *A Steganographic Algorithm High Capacity Despite Better Stegannalysis,* Institute for System Architecture, hal 289-302, Springer, Berlin Heidelberg.
[8] Kumar, A. dan Pooja, K., 2010, Steganography – A Data Hiding Technique, *International Journal of Computer Applications*, Vol 9, hal 19-23.