STEGANOGRAFI CITRA DIGITAL MENGGUNAKAN ENKRIPSI BERDASARKAN PRINSIP KUBUS RUBIK DAN KODE BCH

Fista Monica Deswanti[1)], **Bambang Hidayat**[2)], **Suci Aulia**[3)]
[1]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University
email: fistamonica@gmail.com
[2]Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Telkom University
email: bhidayat@telkomuniversity.ac.id
[2]Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Telkom University
email: suciaulia@telkomuniversity.ac.id

Abstract

Steganography is a technique to hide a secret message into another message, so the existence of the message is not detected by human senses. The secret message could be a picture, audio, video or text and the cover could be a picture, audio, video or text too. To improve the quality and performance of steganography, has been done research that merge the encryption method and error correction method. In this research, it has been simulated steganography system using a secret information in the form of text and digital image as a cover. The encryption method is an encryption based on rubik's cube principle, the error correction method is BCH Code and the steganography is using Least Significant Bit (LSB) method. The accuration (without noise) is 100% and the PSNR is above 56 dB. The system's reliability also tested by adding Gaussian noise and Salt and Pepper noise. The result is a system that uses a BCH code is more resistant to noise than the system that do not use BCH code.

Keywords: steganography, digital image, LSB, rubik's cube, BCH Code

1. PENDAHULUAN

1.1 Latar Belakang

Steganografi merupakan suatu teknik untuk menyembunyikan atau menyisipkan pesan rahasia ke dalam pesan lainnya sehingga keberadaan pesan tidak terdeteksi oleh alat indera manusia [1]. Pesan rahasia tersebut dapat berupa gambar, audio, video atau tulisan dan media penyisipannya dapat berupa gambar, audio, video atau tulisan pula. Pada perkembangannya, steganografi banyak dikombinasikan dengan berbagai metode enkripsi untuk meningkatkan kualitas dan performansinya [2]. Pada penelitian sebelumnya [3], telah dilakukan pengkombinasian metode steganografi dengan metode enkripsi berdasarkan prinsip kubus rubik. Hasil yang diperoleh adalah pesan hasil enkripsi tidak mudah diserang oleh serangan secara statistik maupun serangan *Brute-Force*. Akan tetapi sistem tersebut masih memiliki kekurangan, yaitu rusak atau hilangnya pesan rahasia yang disisipkan akibat gangguan selama proses transmisi data. Hal ini disebabkan karena tidak adanya teknik deteksi dan koreksi *error* pada sistem tersebut.

Oleh karena itu dilakukan penelitian yang diharapkan dapat memperbaiki penelitian sebelumnya, yaitu dengan menggabungkan metode enkripsi dan metode *error control*. Metode enkripsi yang digunakan adalah metode enkripsi berdasarkan prinsip kubus rubik. Teknik deteksi dan koreksi *error* yang digunakan adalah BCH *code*. Sedangkan steganografi menggunakan metode *Least Significant Bit* (LSB). Pesan rahasia yang diamankan berupa tulisan dengan media penyisipannya berupa citra digital.

1.2 Tujuan

1. Menerapkan metode enkripsi dan dekripsi berdasarkan prinsip kubus rubik pada pesan rahasia yang disisipkan.
2. Mengimplementasikan kode BCH pada sistem steganografi yang telah dirancang untuk meningkatkan kualitas dan performansi steganografi.

3. Menganalisis kualitas dan performansi sistem steganografi yang telah dirancang.

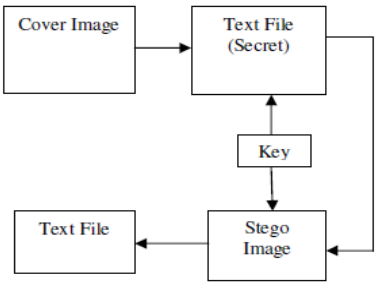
1.3 Citra Digital [4] [5]

Citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Agar dapat diolah dengan dengan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai *diskrit*. Representasi citra dari fungsi malar (kontinu) menjadi nilai-nilai diskrit disebut *digitalisasi* . Citra yang dihasilkan inilah yang disebut citra digital (*digital image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (atau lebar x panjang).

1.4 Steganografi [1] [6]

Steganografi (*steganography*) adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) sedemikian sehingga keberadaan (eksistensi) pesan tidak terdeteksi oleh indera manusia. Steganografi membutuhkan dua properti: wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video.

Proses steganografi terlihat seperti diagram dibawah ini



Gambar 1.*Proses steganografi*

1.5 Metode *Least Significant Bit* [1] [6]

Metode LSB adalah suatu teknik penyembunyian data yang dilakukan dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit data rahasia.

Pada susunan bit di dalam sebuah *byte* (1 *byte* = 8 bit), terdapat bit yang paling berarti *Most Significant Bit* atau MSB dan bit yang paling kurang berarti *Least Significant Bit* atau LSB. Bit yang cocok untuk diganti adalah bit LSB karena perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya.

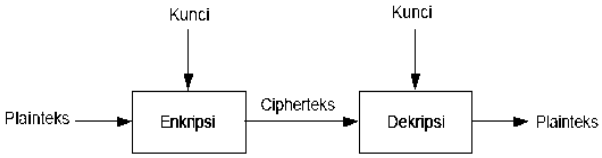
1.6 Kriptografi [1]

Kriptografi adalah ilmu dan seni yang menjaga kerahasiaan suatu pesan dengan menyandikan pesan tersebut menjadi suatu bentuk yang tidak dapat dimengerti lagi maknanya.

Berikut ini adalah istilah-istilah dalam kriptografi.

1. *Plaintext* adalah pesan asli sebelum dilakukan proses enkripsi.
2. *Ciphertext* adalah pesan hasil enkripsi.
3. Enkripsi adalah proses untuk mengubah *plaintext* menjadi *ciphertext*.
4. Dekripsi adalah proses untuk mengubah *ciphertext* menjadi *plaintext*.
5. *Key* adalah suatu bilangan yang dirahasiakan, digunakan untuk proses enkripsi dan dekripsi.

Skema enkripsi dan dekripsi mengunakan suatu kunci :



Gambar 2.*Skema enkripsi dan dekripsi*

1.7 Metode Kubus Rubik [7] [8]

1.7.1 Enkripsi Berdasarkan Prinsip Kubus Rubik

I_o merupakan citra *grayscale* berukuran M x N. Langkah-langkah algoritma berdasarkan prinsip kubus rubik adalah sebagai berikut.

1. Bangkitkan secara acak vektor kolom Kr dan vektor baris Kc dengan panjang M dan N. Elemen Kr(j) dan Kc(i) masing-masing bernilai acak dalam set A = {0,1,2,…,2^8-1}.
2. Tentukan jumlah iterasi, $ITER_{max}$ dan inisialisasi *counter* ITER di 0.
3. Tambahkan *counter* dengan 1 : ITER = ITER+1.

4. Untuk setiap baris i dari citra I_o,
   a. Hitung jumlah semua elemen dari baris i, dinotasikan dengan $\alpha(i)$.
   b. Hitung modulo 2 dari $\alpha(i)$, dinotasikan dengan $M_{\alpha(i)}$
   c. Baris i digeser secara sirkular ke kiri atau kanan sesuai Kr(j) (piksel-piksel citra dipindah posisi Kr(j) arah kiri atau kanan). Jika $M_{\alpha(i)}=0$, geser sirkular ke kanan, lainnya geser sirkular ke kiri.
5. Untuk setiap kolom j dari citra I_o,
   a. Hitung jumlah semua elemen dari kolom j, dinotasikan dengan $\beta(j)$.
   b. Hitung modulo 2 dari $\beta(j)$, dinotasikan dengan $M_{\beta(j)}$
   c. Kolom j digeser sirkular atas atau bawah, sesuai Kc(j), jika $M_{\beta(j)} = 0$, geser sirkular ke atas, lainnya geser sirkular ke bawah.

Langkah 4 dan 5 menghasilkan *scrambled image*, dinotasikan dengan I_{SCR}.

6. Menggunakan vektor Kc, operator *bitwise* XOR diaplikasikan ke setiap baris dari *scrambled image* I_{SCR}.
7. Menggunakan vektor Kr, operator *bitwise* XOR diaplikasikan ke setiap kolom dari *scrambled image* I_{SCR}.
8. Jika ITER=$ITER_{max}$, citra terenkripsi I_{ENC} berhasil dibuat dan proses enkripsi selesai. Jika tidak, kembali ke langkah 3.

1.7.2 Dekripsi Berdasarkan Prinsip Kubus Rubik

Citra terdekripsi I_o didapatkan dari citra terenkripsi I_{ENC} dengan kunci rahasia Kr, Kc dan $ITER_{max}$.

1. Inisialisasi ITER=0
2. Tambahkan *counter* ITER dengan 1 : ITER=ITER+1.
3. Operasi *bitwise* XOR diaplikasikan pada vektor Kr dan setiap kolom dari citra terenkripsi I_{ENC}.
4. Menggunakan vektor Kc, operasi *bitwise* XOR diaplikasikan pada setiap baris citra I_1
5. Untuk setiap kolom j dari *scrambled image* I_{SCR},
   a. Hitung jumlah semua elemen dari kolom j, dinotasikan dengan $\beta_{SCR}(j)$.
   b. Hitung modulo 2 dari $\beta_{SCR}(j)$, dinotasikan dengan $M_{\beta SCR(j)}$
   c. Kolom j digeser sirkular atas atau bawah sesuai Kc(i). Jika $M_{\beta SCR(j)}=0$, geser sirkular ke bawah, lainnya geser sirkular atas.
6. Untuk setiap baris I dari scrambled image I_{SCR},
   a. Hitung jumlah semua elemen dari baris i, dinotasikan dengan $\alpha_{SCR}(i)$.
   b. Hitung modulo 2 dari $\alpha_{SCR}(i)$, dinotasikan dengan $M\alpha_{SCR(i)}$
   c. Baris i digeser secara sirkular ke kiri atau kanan sesuai Kr(j). Jika $M\alpha_{SCR(i)}=0$, geser sirkular ke kiri, lainnya geser sirkular ke kanan.
7. Jika ITER=$ITER_{max}$, citra I_{ENC} terdekripsi menjadi citra asli dan proses dekripsi selesai. Jika tidak, kembali ke langkah 2.

1.8 Kode BCH [9] [10]

Kode BCH mempunyai kemampuan untuk mengkoreksi semua bentuk acak dari "t" *error*. BCH *decoder* menyediakan informasi mengenai jumlah *bit error* yang terjadi pada blok yang didekodekan. Jika jumlah *error* ini masih dalam kapasitas kemampuan koreksi dari skema BCH yang dipilih, maka *error* ini masih akan diperbaiki. Sedangkan jika jumlah *error* yang dideteksi melebihi kapasitas kemampuan dari kode BCH, maka kode BCH hanya menunjukkan informasi mengenai jumlah *error* yang ada tanpa kemampuan mengoreksinya.

Untuk semua integer positif m ($m \geq 3$) dan t ($t < 2^{m-1}$) serta panjang data k, terdapat kode BCH biner dengan parameter berikut:

Panjang blok : $n = 2^m - 1$ .....(1)
Jumlah digit *parity check*: $n - k \leq mt$ .....(2)
Jarak minimum : $d_{min} \geq 2t + 1$ .....(3)
Dimana m adalah *parity check bit* , k adalah bit informasi, dan t : *correctable error*.

1.8.1 BCH *encoding*

Langkah-langkah *encoding* BCH adalah sebagai berikut :

1. Ubah bit informasi ke dalam polinomial m(x).
2. Cari nilai generator polinomial g(x).

3. Lakukan perkalian antara x^{n-k} dan m(x).
4. Lakukan pembagian antara $x^{n-k}.m(x)$ terhadap g(x). Didapatkan hasil pembagian v(x) dan sisa pembagian h(x).
5. Didapatkan *codeword* $c(x) = h(x) + x^{n-k}$ dan m(x). Lalu diubah lagi ke dalam bit.

1.8.2 BCH *decoding*

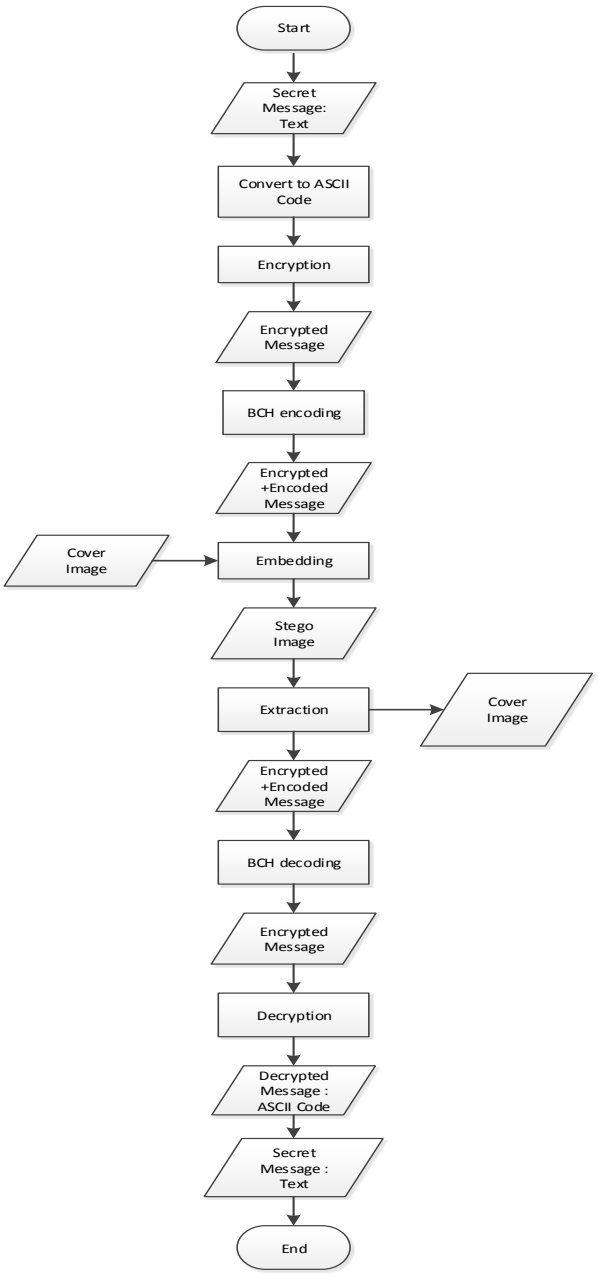
Langkah-langkah *decoding* BCH adalah sebagai berikut:

1. Didapatkan r(x) disisi penerima.
   $r(x) = c(x) + e(x)$
   dimana c(x) adalah bit-bit *codeword* yang dikirim dan e(x) adalah pola *error* yang terjadi selama proses transmisi.
2. Hitung sindrom $S = (S_1,S_2,\ldots,S_{2t})$ dari polinomial terimaan r(x).
3. Tentukan *error location* polinomial $\sigma(x)$ dari komponen sindrom $S_1,S_2,\ldots,S_{2t}$.
4. Tentukan *error location numbers* $\beta_1,\beta_2,\ldots\beta_v$ dengan mencari akar-akar $\sigma(x)$.
5. Perbaiki *error* dengan mengganti bit 0 menjadi bit 1, atau sebaliknya.

2. METODE PENELITIAN

Secara umum, sistem yang dibuat digambarkan seperti berikut.

1. Pesan rahasia yang berupa tulisan diubah terlebih dahulu ke dalam bentuk biner dengan menggunakan kode ASCII.
2. Kode ASCII tersebut dienkripsi dengan metode enkripsi berdasarkan prinsip kubus rubik.
3. Pesan terenkripsi di-*encode* dengan menggunakan teknik BCH *encoding*. Proses ini bertujuan untuk mengurangi kesalahan data yang diterima. Kode BCH yang digunakan pada penelitian ini adalah BCH(15,5), yang artinya setiap 5 bit masukan diproses menjadi 15 bit keluaran, sehingga dapat memperbaiki maksimal 3 kesalahan.
4. Pesan hasil *encoding* disisipkan ke dalam *cover image* yang berupa citra RGB dengan menggunakan metode *Least Significant Bit* (LSB), dimana penentuan piksel tempat penyisipan pesan menggunakan syarat MSB ≥ 1, dengan kata lain pesan disisipkan pada piksel yang memiliki nilai diatas 128. Hasil penyisipan berupa *stego image*.
5. *Stego image* siap dikirimkan ke penerima.
6. Di sisi penerima, *stego image* diekstraksi untuk memisahkan pesan hasil *encoding* dengan *cover image*.
7. Pesan hasil *encoding* di-*decode* menggunakan teknik BCH *encoding*, sehingga menghasilkan pesan terenkripsi.
8. Pesan terenkripsi didekripsi menggunakan metode dekripsi berdasarkan prinsip kubus rubik, sehingga pesan kembali menjadi kode ASCII. Kemudian kode ASCII diubah menjadi pesan rahasia seperti semula.



Gambar 3.*Diagram alir sistem*

3. HASIL DAN PEMBAHASAN

3.1 Pengaruh Jumlah Karakter Terhadap Nilai Akurasi Pesan Terekstraksi dan Nilai PSNR Citra Stego

Tabel 1.*Hasil enkripsi dekripsi kubus rubik*

Pesan	Pesan Terenkripsi	Pesan Hasil Dekripsi	Pesan Terekstraksi
stegano grafi	r®ë\,ÛñV-KÇ	stegano grafi	steganografi
citra	P»$þÈà8	citra	citra
digital	P b1È\	digital	digital
enkripsi 123	ý,wÍ¯²8iKÊ	enkripsi 123	enkripsi123
kubusR ubik	«ó.þ¯ê8i-KÃ	kubusR ubik	kubusRubik

Sistem steganografi yang digabungkan dengan enkripsi berdasarkan prinsip kubus rubik memiliki tingkat akurasi 100%. Hal ini terlihat dari tabel 1, bahwa pesan terenkripsi yang disisipkan ke dalam suatu citra *cover* kemudian dapat didekrip dan diekstrak secara sempurna menjadi pesan semula.

Tabel 2.*Pengaruh jumlah karakter terhadap akurasi dan PSNR*

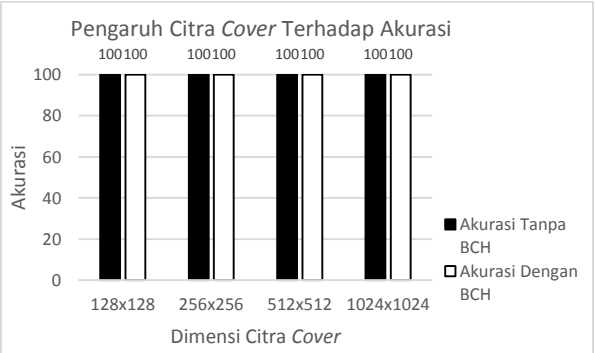
Juml	Akurasi		PSNR	
	Tanpa BCH	Dgn BCH	Tanpa BCH	Dgn BCH
278	100	100	76.387	71.648
572	100	100	73.496	68.690
1229	100	100	69.956	65.181
2180	100	100	67.630	62.849
3319	100	100	65.732	61.034
4568	100	100	64.459	59.639
6209	100	100	63.126	58.343
8675	100	100	61.596	56.841
9043	100	-	61.409	-
11490	100	-	60.433	-

- artinya bit yang disisipkan sudah melebihi kapasitas citra *cover* sehingga nilai akurasi dan PSNR tidak terdeteksi.

Pada tabel 2 terlihat bahwa berapapun jumlah karakter masukan, nilai akurasi tetap 100% baik pada sistem yang menggunakan kode BCH maupun yang tanpa BCH. Jumlah karakter masukan tidak mempengaruhi tingkat akurasi pesan terekstraksi. Hal ini membuktikan bahwa dalam keadaan tanpa gangguan, sistem steganografi ini memiliki tingkat akurasi sebesar 100%.

Jumlah karakter masukan mempengaruhi nilai PSNR citra stego. Semakin banyak jumlah karakter masukan, maka nilai PSNR akan semakin kecil. Hal ini disebabkan karena semakin banyak jumlah karakter masukan menyebabkan semakin banyak pula jumlah piksel yang bit LSB-nya diganti, sehingga kualitas citra stego akan semakin berkurang. Selain itu nilai PSNR sistem yang menggunakan BCH lebih kecil daripada sistem yang tanpa BCH karena kode BCH akan memperpanjang bit pesan. Pada penelitian ini digunakan BCH (15,5), yang artinya setiap 5 bit masukan akan diproses menjadi 15 bit keluaran. Semakin panjang bit pesan yang akan disisipkan maka semakin banyak pula piksel dari citra *cover* yang akan berubah nilainya, sehingga nilai PSNR akan semakin kecil.

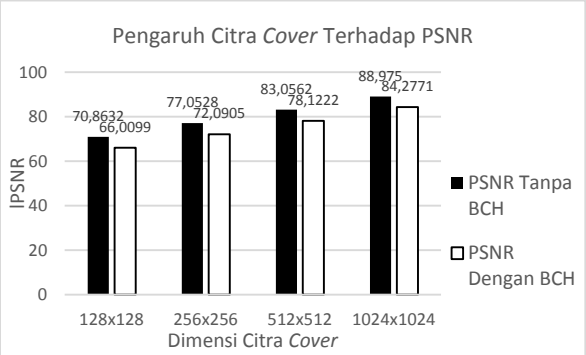
3.2 Pengaruh Citra *Cover* Terhadap Nilai Akurasi Pesan Terekstraksi dan Nilai PSNR Citra Stego



Gambar 4.*Pengaruh citra cover terhadap nilai akurasi*

Pada gambar 4 terlihat bahwa nilai akurasi pesan terekstraksi pada sistem yang menggunakan kode BCH dan yang tidak menggunakan kode BCH tetap 100% walaupun dimensi citra *cover* berbeda-beda. Semakin besar dimensi citra, semakin banyak pula tempat untuk pesan yang akan disisipkan.

Namun hal tersebut tidak mempengaruhi tingkat akurasi pesan terekstraksi.

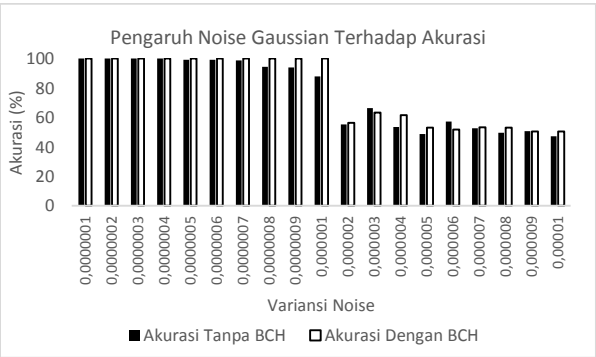


Gambar 5.*Pengaruh citra cover terhadap nilai PSNR*

Dari gambar 5 dapat terlihat bahwa semakin besar dimensi dari citra *cover*, semakin besar pula nilai PSNRnya. Nilai PSNR sistem yang menggunakan kode BCH lebih kecil daripada sistem yang tanpa BCH. Hal ini disebabkan karena penggunaan kode BCH dapat memperpanjang bit pesan yang akan disisipkan.

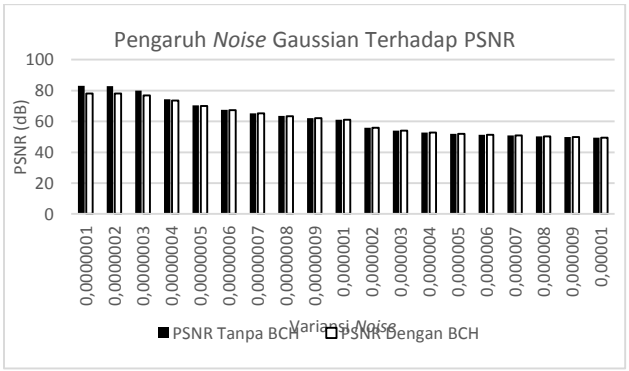
3.3 Pengaruh Serangan *Noise* Terhadap Nilai Akurasi Pesan Terekstraksi dan Nilai PSNR Citra Stego

3.3.1 *Noise* Gaussian Variansi 10^{-7} sampai 10^{-5}



Gambar 6.*Pengaruh noise Gaussian terhadap nilai akurasi*

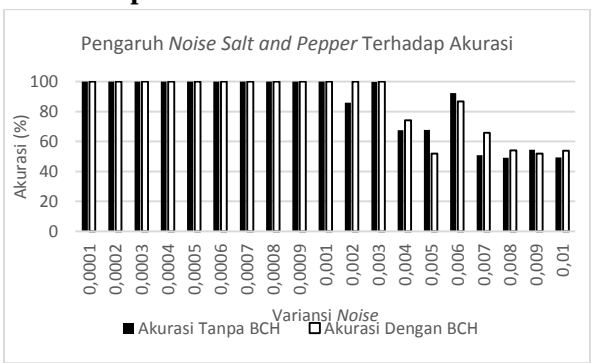
Pada gambar 6 terlihat bahwa sistem yang menggunakan kode BCH tahan terhadap *noise* Gaussian hingga variansi $1x10^{-6}$, sedangkan pada sistem yang tidak menggunakan kode BCH hanya tahan terhadap *noise* Gaussian hingga variansi $5x10^{-7}$ saja. Hal ini membuktikan bahwa kode BCH bekerja secara efektif untuk memperbaiki *error* yang ada. Akan tetapi pada saat variansi sudah melebihi batas maksimal ketahanannya, kode BCH sudah tidak efektif untuk mengatasi *error* yang ada. Hal ini terjadi karena *error* yang disebabkan oleh *noise* tersebut sudah melebihi kapasitas kemampuan koreksi kode BCH sehingga nilai akurasi pada sistem yang menggunakan kode BCH bisa jadi lebih buruk daripada sistem yang tidak menggunakan kode BCH. Selain itu, nilai akurasi mengalami fluktuasi naik turun (tidak stabil) pada kedua sistem ketika variansi diatas $1x10^{-6}$. Hal ini dapat disebabkan karena persebaran *noise* Gaussian bersifat acak sehingga bisa jadi variansi kecil tetapi *error* besar atau sebaliknya.



Gambar 7. *Pengaruh noise Gaussian terhadap nilai PSNR*

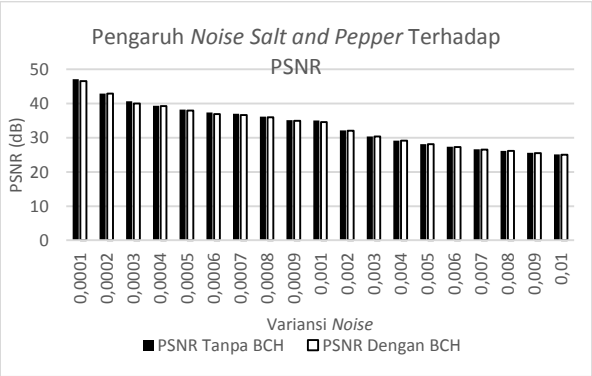
Dari gambar 7 terlihat bahwa semakin besar nilai variansi *noise* Gaussian, maka nilai PSNR akan semakin kecil. jangkauan *noise* tersebut. Hal ini disebabkan karena semakin besar *noise* Gaussian yang diberikan maka piksel yang berubah akan semakin banyak, sehingga nilai PSNR akan semakin kecil.

3.3.2 *Noise* Salt and Pepper Variansi 10^{-4} sampai 10^{-2}



Gambar 8. *Pengaruh noise Salt and Pepper terhadap nilai akurasi*

Akurasi pada sistem yang menggunakan kode BCH mencapai 100% pada batas maksimal variansi $3x10^{-3}$, sedangkan pada sistem yang tidak menggunakan kode BCH mencapai akurasi 100% pada batas maksimal variansi $1x10^{-3}$. Hal tersebut membuktikan bahwa sistem yang menggunakan kode BCH lebih tahan terhadap *noise* Salt and Pepper dibandingkan sistem yang tanpa BCH. Ketika jumlah *error* melebihi kapasitas kemampuan koreksinya, bisa jadi sistem yang menggunakan kode BCH memliki nilai akurasi yang lebih buruk dibandingkan dengan sistem yang tanpa kode BCH, misalnya pada saat variansi $5x10^{-3}$. Dari grafik tersebut terlihat bahwa nilai akurasi yang diperoleh tidak stabil atau mengalami fluktuasi naik turun ketika variansi melebihi batas maksimal ketahanannya. Hal ini disebabkan karena *noise* Salt and Pepper termasuk dalam *noise* impuls dimana kehadirannya bersifat acak sehingga bisa jadi variansi besar tetapi akurasi kecil atau sebaliknya.



Gambar 9. *Pengaruh noise Salt and Pepper terhadap nilai PSNR*

Pada gambar 9 dapat dilihat bahwa semakin besar *noise* yang berikan, maka kualitas citra stego akan semakin berkurang. Hal ini disebabkan karena semakin besar *noise* yang diberikan, maka piksel yang berubah akibat serangan *noise* tersebut akan semakin banyak, sehingga nilai PSNR semakin kecil.

4. KESIMPULAN

Hasil perancangan sistem steganografi dengan menambahkan enkripsi berdasarkan prinsip kubus rubik menghasilkan nilai akurasi 100% dalam keadaan tanpa gangguan. Sistem yang tidak menggunakan kode BCH memliki kualitas citra stego yang lebih baik dibandingkan dengan sistem yang menggunakan kode BCH, karena kode BCH akan memperpanjang bit pesan.

Ketika diserang dengan *noise* Gaussian, sistem yang menggunakan kode BCH tahan hingga variansi $1x10^{-6}$, sedangkan pada sistem yang tidak menggunakan kode BCH hanya tahan hingga variansi $5x10^{-7}$ saja. Kemudian pada saat diserang dengan menggunakan *noise* Salt and Pepper, sistem yang menggunakan kode BCH tahan hingga $3x10^{-3}$ dan sistem yang tanpa BCH hanya tahan hingga variansi $1x10^{-3}$ saja. Sedangkan nilai PSNR akan menurun seiring dengan bertambahnya nilai variansi *noise*. Dari hasil tersebut dapat disimpulkan bahwa sistem yang menggunakan kode BCH lebih tahan terhadap serangan *noise* Gaussian dan Salt and Pepper dibandingkan dengan sistem yang tanpa BCH. Namun ketika jumlah *error* melebihi kapasitas kemampuan koreksinya, bisa jadi sistem yang menggunakan kode BCH memliki performansi yang lebih buruk dibandingkan dengan sistem yang tanpa kode BCH Sistem yang menggunakan kode BCH dan yang tanpa kode BCH sama-sama memiliki ketahanan terhadap *noise* Salt and Pepper yang lebih baik dibandingkan dengan *noise* Gaussian.

5. REFERENSI

[1] R. Munir, Kriptografi, Bandung: Informatika, 2006.

[2] N. M. L. D. Aristia, Simulasi dan Analisis Steganografi Citra Digital Menggunakan Metode AES dan BCH Code, Bandung: Institut Teknologi Telkom, 2013.

[3] S. Raniprima, Simulasi dan Analisis Steganografi Citra Digital Dengan Enkripsi Berdasarkan Prinsip Kubus Rubik, Bandung: Universitas Telkom, 2014.

[4] R. Munir, Pengolahan Citra Digital Dengan Pendekatan Algoritmik, Bandung: Informatika, 2004.

[5] M. Hery, Konsep Pengolahan Citra Digital dan Ekstraksi Fitur, Surabaya: Graha Ilmu, 2010.

[6] K. Devi dan G.Sudha, "An analysis of LSB Based Image Steganography Techniques," 2014.

[7] K. Loukhaoukha, J.-Y. Chouinard dan A. Berdai, "A Secure Image Encryption Algorithm Based on Rubik's Cube Principle," *Journal of Electrical and Computer Engineering Volume 2012*, 2012.

[8] K. Loukhaoukha, M. Nabti dan K. Zebbiche, "An Efficient Image Encryption Algorithm Based on Blocks Permutation and Rubik's Cube Principle for Iris Image," 2013.

[9] J. C. Moreira dan P. G. Farrel, Essentials of Error Control Coding, England: John Wiley and Sons, Ltd, 2006.

[10] F. Caroline, Simulasi dan Analisis Steganografi Citra Digital Menggunakan Metode Sudoku Puzzle Acak dan kode BCH, Bandung: Universitas Telkom, 2014.