

MIPA

ABSTRACT & EXECUTIVE SUMMARY

HIBAH BERSAING



**Sistem Pengkodean File Image Kedalam Citra Foto
Menggunakan Teknik Steganografi**

Oleh :

Kiswara Agung Santoso, M.Kom NIDN : 0007097202

Kusbudiono, M.Si NIDN : 0030047703

Ahmad Kamsyakawuni, M.Kom NIDN : 0029117202

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER**



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS JEMBER
LEMBAGA PENELITIAN

Alamat : Jl. Kalimantan No. 37 Jember Telp. 0331-337818, 339385 Fax. 0331-337818
e-Mail : penelitian.lemlit@unej.ac.id

Sistem Pengkodean File Image Kedalam Citra Foto Menggunakan Teknik Steganografi

Peneliti : Kiswara Agung Santoso¹, Kusbudiono², Ahmad Kamsyakawuni³

Mahasiswa : Agnes Ika Nurvitaningrum⁴, Agustina Muharomah⁵, Dwi Agustin Retno Wardani⁶,
Novita Sana Susanti⁷

Sumber Dana : DP2M

¹Jurusan Matematika Fakultas MIPA Universitas Jember

²Jurusan Matematika Fakultas MIPA Universitas Jember

³Jurusan Matematika Fakultas MIPA Universitas Jember

⁴Jurusan Matematika Fakultas MIPA Universitas Jember

⁵Jurusan Matematika Fakultas MIPA Universitas Jember

⁶Jurusan Matematika Fakultas MIPA Universitas Jember

⁷Jurusan Matematika Fakultas MIPA Universitas Jember

ABSTRAK

Setiap harinya banyak orang bekerja dengan menggunakan jaringan komputer dan internet. Perkembangan teknologi internet yang sangat pesat ini mampu menghubungkan hampir semua komputer yang ada di dunia. Internet dengan segala fasilitas di dalamnya sudah menjadi bagian kehidupan sehari-hari dan menjadi begitu di andalkan oleh banyak orang. Kebutuhan akan internet tersebut tentunya menuntut internet menjadi sebuah dunia yang aman selayaknya kita menghendaki keamanan di dunia nyata. Keamanan dan kerahasiaan data sangat penting terutama dalam pertukaran informasi. Untuk menjaga kerahasiaan suatu pesan tersebut, baik pesan yang akan dikirimkan maupun pesan yang telah disimpan di dalam media penyimpanan, telah dikembangkan teknik steganografi yang bisa menyamarkan pesan tersebut pada suatu media digital berupa gambar yang dapat di akses oleh setiap orang namun mereka tidak menyadari bahwa media gambar tersebut telah disisipi pesan rahasia.

Pada penelitian ini dilakukan implementasi pengkodean *file* foto dengan melakukan manipulasi nilai bit pada *depth color* nya. Pada penelitian ini juga digunakan parameter untuk memandingkan

kualitas citra sebelum dan sesudah proses pengkodean dengan melihat nilai MSE (*Mean Squared Error*) dan PSNR (*Peak Signal to Noise Ratio*). Selain itu juga digunakan perbandingan bit per bit agar dapat diketahui seberapa besar perubahan citra oleh proses steganografi.

Kata kunci : *Steganografi, MSE, PSNR, image, depth color*



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS JEMBER
LEMBAGA PENELITIAN

Alamat : Jl. Kalimantan No. 37 Jember Telp. 0331-337818, 339385 Fax. 0331-337818
e-Mail : penelitian.lemlit@unej.ac.id

Sistem Pengkodean File Image Kedalam Citra Foto Menggunakan Teknik Steganografi

Peneliti : Kiswara Agung Santoso¹, Kusbudiono², Ahmad Kamsyakawuni³

Mahasiswa : Agnes Ika Nurvitaningrum⁴, Agustina Muharomah⁵, Dwi Agustin Retno Wardani⁶,
Novita Sana Susanti⁷

Sumber Dana : DP2M

¹Jurusan Matematika Fakultas MIPA Universitas Jember

²Jurusan Matematika Fakultas MIPA Universitas Jember

³Jurusan Matematika Fakultas MIPA Universitas Jember

⁴Jurusan Matematika Fakultas MIPA Universitas Jember

⁵Jurusan Matematika Fakultas MIPA Universitas Jember

⁶Jurusan Matematika Fakultas MIPA Universitas Jember

⁷Jurusan Matematika Fakultas MIPA Universitas Jember

LATAR BELAKANG

Dewasa ini teknologi informasi telah berkembang pesat seiring dengan berkembangnya hardware dan software komputer. Salah satu teknologi informasi yang sangat menonjol adalah teknologi citra digital (*image*/foto, video, suara) dimana dengan teknologi tersebut kita dapat mensimulasikan dari suatu keadaan nyata kedalam bentuk visual, baik itu berupa suara, gambar(*image*) maupun gambar bergerak (video). Dalam melakukan transfer citra digital sering kali diperlukan spesifikasi hardware yang tinggi dan waktu yang lama untuk memprosesnya. Seiring dengan berkembangnya ilmu pengetahuan, software maupun hardware yang sangat cepat memungkinkan seseorang untuk memanipulasi citra digital menjadi lebih mudah dan cepat. Salah satu perkembangan citra digital yang paling populer adalah *image*.

Dengan adanya teknologi internet, proses manipulasi *image* dapat dilakukan oleh orang lain secara *mobile* dari tempat terpisah dan diluar sistem dimana *image* itu berada. Adanya kemudahan dalam memanipulasi *image* dari jarak jauh (memakai internet) bukan berarti selalu bersifat positif atau baik, tetapi hal tersebut juga memunculkan permasalahan baru bagi dunia informasi. Adapun permasalahan yang paling besar saat menggunakan internet adalah kecepatan akses yang lambat dan keamanan data saat melakukan transfer. Komunikasi melalui internet sangat rentan terhadap hilangnya atau berubahnya data yang disebabkan oleh kesalahan sistem maupun disengaja oleh adanya oknum yang tidak bertanggung jawab.

Image processing adalah suatu pengolahan sinyal dengan input berupa gambar (*image*) dan ditransformasikan menjadi gambar lain sebagai keluarannya dengan teknik tertentu. *Image processing* dilakukan untuk memperbaiki data sinyal gambar yang terjadi akibat kesalahan transmisi dan untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh sistem penglihatan

manusia. *Image processing* dapat pula digunakan untuk membuat gambar menjadi sulit diinterpretasikan. Hal ini sengaja dilakukan untuk keperluan pengkodean atau mengecilkan ukuran file (Jumain, 2012). Ukuran file suatu image tergantung dari dua komponen yaitu ukuran *image* dan *color depth*. *Color depth* adalah jumlah warna yang ditampilkan di monitor oleh *video card*. Semakin banyak warna yang digunakan, semakin realistis tampilan yang dapat dilihat. Tidak selamanya perubahan *color depth* suatu *image* dapat meningkatkan maupun menurunkan kualitas gambar 8

secara signifikan. Hal ini terjadi apabila *image* itu terbatas pada jumlah warna tertentu (misalkan hitam dan putih saja). Dengan mengubah *color depth* dari setiap *pixel image* maka secara keseluruhan *image* tersebut juga akan berubah. Dari sini pengkodean *image* dapat dilakukan dengan cara mengubah *color depth* dari setiap *pixel image*.

Dewasa ini perkembangan teknik pengkodean sangat pesat terutama yang diterapkan dalam bidang komputer guna menjaga keamanan data saat berkomunikasi melalui internet. Adapun bentuknya dalam bidang komputer adalah pengkodean data yang berupa data teks, suara, video maupun gambar atau *image*. Khusus untuk pengkodean *image* ada beberapa cara yang pernah dilakukan seperti merubah *color depth pixel*, mengacak posisi pixel dan sebagainya. Pada tahun 2008, Muhammad Ali Bani Younes membuat teknik pengkodean *image* dengan cara membagi *image* asli beberapa blok, kemudian blok-blok tersebut diacak dengan cara kombinasi dan permutasi. Kesimpulan dari penelitian tersebut adalah semakin besar ukuran bloknya atau semakin sedikit jumlah bloknya maka hasil dekripsi makin baik. Selanjutnya algoritma ini dikenal dengan nama algoritma blowfish.

Amitava Nag (2011) memperkenalkan algoritma pengkodean *image* dengan menggunakan transformasi Affine. Dalam proses enkripsi dan dekripsi algoritma ini melalui dua tahap, dimana tahap pertama adalah mengkodekan *image* menggunakan operasi XOR dengan kunci 4 bit. Selanjutnya hasil pengkodean tahap pertama ini dikodekan lagi menggunakan transformasi Affine sebagai tahap kedua. Kesimpulan dari penelitian tersebut menyebutkan bahwa korelasi nilai pixel menurun secara signifikan setelah dikodekan menggunakan transformasi Affine

Shocif(2013) mengkodekan *image* dengan cara merubah nilai pixel RGB. Adapun cara merubahnya yaitu dengan mengalikan nilai pixel *image* asli dengan suatu matriks kunci yang invertibel kemudian menjumlahkan tiap elemen matriks hasil perkalian dengan jumlah diagonal utama matriks kunci.

Kesimpulan dari penelitian tersebut adalah ukuran file hasil enkripsi lebih kecil daripada file aslinya. Nilai elemen matriks kunci sebaiknya lebih besar dari 100 agar hasil enkripsi sulit untuk diinterpertasikan.

Mengingat pentingnya sistem pengkodean dalam dunia informatika dan melihat beberapa teknik pengkodean yang telah dipublikasikan, maka penulis tertarik untuk membuat teknik pengkodean yang baru dimana hasil pengkodean nantinya selain sulit untuk diinterpertasikan juga mempunyai ukuran file yang lebih kecil. 9

Berdasarkan beberapa hal yang telah disampaikan di atas, penulis ingin melakukan penelitian tentang *image processing* yaitu mengkodekan suatu *image* tetapi sekaligus mengecilkan ukuran filenya.

Dalam memperkecil ukuran file peneliti tidak melakukan pengecilan ukurannya tetapi memanipulasi layer RGB sehingga dapat memperkecil ukuran filenya. Secara umum suatu metode pengkodean menggunakan kunci untuk mengkodekan maupun membuka kode, dimana kunci tersebut dimasukkan sebelum pengkodean dilakukan. Pada penelitian ini peneliti akan membuat suatu metode pengkodean dengan menggunakan algoritma CSO dimana algoritma tersebut selama ini digunakan untuk menyelesaikan masalah optimasi. Selain itu yang membedakan dengan algoritma ini dengan lainnya adalah untuk mengkodekan suatu *image* tidak memerlukan kunci, tetapi saat membuka kode tersebut membutuhkan kunci yang dibangkitkan oleh sistem saat proses pengkodean dilakukan. Selanjutnya algoritma pengkodean pada penelitian ini akan dibandingkan dengan algoritma lain yang telah ada sebelumnya.

Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah :

- a. Membuat Teknik Pengkodean suatu *image* dengan cara memanipulasi nilai pixelnya

- b. Membuat Teknik pengkodean image sehingga ukuran file encrypt image lebih kecil daripada image aslinya.
- c. Menyembunyikan image pada image lain menggunakan teknik steganografi

METODE PENELITIAN

Manipulasi Pixel

Dalam penelitian ini akan dibuat suatu pengkodean image dengan cara memanipulasi pixelnya. Adapun metode manipulasinya dapat disajikan sebagai berikut :

1. Membagi image kedalam beberapa blok yang berukuran 200 pixel x 200 pixel
2. Dalam tiap blok, dibangkitkan pixel pada layer *Blue* secara acak antara 0 sampai 200, dengan syarat nilai pixel dalam satu baris tidak ada yang sama
3. Misalkan nilai pixel posisi (x,y) pada layer *Blue* adalah z. Menukarkan pixel pada posisi (x,y) pada layer *Red* dengan pixel posisi (z,y) pada layer green.
4. Melakukan pertukaran ini hingga semua pixel dalam satu blok dipertukarkan.
5. Terapkan langkah 3 dan 4 pada semua blok yang ada.
6. Untuk proses dekripsi lakukan hal yang sama seperti langkah 3 dan 4. Disini proses membangkitkan bilangan random tidak diperlukan lagi karena nilai tersebut sudah ada pada layer *Blue*
7. Membuat program simulasi menggunakan Matlab 7,8 (2012 a)

Menurunkan Ukuran File Encrypt Image

Berikut ini adalah metode pengkodean image dimana nantinya hasil dari pengkodeannya mempunyai ukuran file yang lebih kecil dari file aslinya.

1. Membagi nilai pixel pada semua layer dengan suatu bilangan bulat
2. Mengganti nilai setiap pixel pada kolom terakhir dengan rata-rata sisa pembagian yang dilakukan pada no 1 dari semua pixel pada kolom yang sama
3. Untuk proses dekripsi adalah mengalikan tiap pixel dengan suatu bilangan bulat yang telah ditentukan pada no. 1, dan menjumlahkannya dengan nilai pixel pada baris terakhir.
4. Membuat program simulasi menggunakan Matlab 7,8 (2012 a)

HASIL DAN PEMBAHASAN

5.1. Pembentukan Kunci

Pada pembahasan hasil penelitian ini dimulai dengan skema pembentukan kunci yang digunakan untuk mengkodekan image. Disini kunci dibagi menjadi tiga bagian yaitu :

Bagian 1 : berisi suatu bilangan antara 0..255

Bagian 2 : berisi karakter R, G atau B. R mengindikasikan layer Red, G mengindikasikan layer green dan B mengindikasikan layer blue

5.2. Sistem Pengkodean Berdasarkan Manipulasi Pixel

Dalam melakukan pengkodean ini karena terkait langsung dengan kunci maka perlu diketahui hubungan antara kunci dan cara pengkodeannya. Karakter yang muncul di kunci mengindikasikan layer yang akan diisi dengan bilangan acak antara 1 sampai 200. Misalkan kuncinya adalah "25B" selanjutnya sistem pengkodeannya dilakukan melalui beberapa tahap yaitu :

- 1) Import data dari *image* yang akan dikodekan (dalam hal ini data berupa matriks berukuran M x N)
- 2) Bagilah image kedalam blok berukuran 200 pixel x 200 pixel
- 3) Ekstrak pixel dari layer yang tidak ada pada kunci bagian 2, dalam hal ini adalah layer Red ($R_1(x,y)$) dan layer Green ($G_1(x,y)$), dimana (x,y) adalah posisi pixel

- 4) Bangkitkan nilai random antara 1 sampai 200 untuk setiap bloknya pada layer Blue ($B_1(x,y)$) dengan syarat tidak ada nilai yang sama dalam satu kolom.
- 5) Tukarkan $R_1(x,y)$ dengan $G_1(B_1(x,y),y)$
- 6) Penggabungan $R_1(x,y)$, $G_1(x,y)$ dan $B_1(x,y)$ menghasilkan *encrypt image* hasil pertukaran pixel
- 7) Untuk melakukan decoding dapat langsung dilakukan seperti langkah no. 6, tanpa harus membangkitkan bilangan random pada layer *Blue*

Sebagai ilustrasi berikut adalah contoh matriks yang diambil dari sebagian image dimana setiap elemen matriks merupakan representasi dari nilai pixel image asli. 24

Selanjutnya dibangkitkan bilangan random pada layer Blue dengan syarat tidak boleh ada nilai yang sama dalam satu kolom, untuk menggantikan nilai pixel yang telah ada.

Sistem Pengkodean Berdasarkan Penurunan Intensitas Cahaya

Pada dasarnya yang mempengaruhi besar file suatu image ada dua faktor yaitu ukuran (dimensi) dan besar intensitas cahaya dari image tersebut. Pada penelitian ini untuk mengkodekan sekaligus menurunkan besar file dari suatu image yaitu dengan cara mengurangi intensitas cahaya di setiap pixel pada setiap layer.

Adapun caranya adalah sebagai berikut :

- 1) Membagi pixel dengan bagian pertama dari kunci (misalnya : 25)

$$R_1(x,y)=[R(x,y)/25] \quad G_1(x,y)=[G(x,y)/25] \quad B_1(x,y)=[B(x,y)/25]$$

- 2) Menambah satu kolom pada pixel pada *image*

$$R_1((x+1),y)=\Sigma(R(x,y)-R_1(x,y))/MM_{x=1} \quad G_1((x+1),y)=\Sigma(G(x,y)-G_1(x,y))/MM_{x=1}$$

$$B_1((x+1),y)=\Sigma(B(x,y)-B_1(x,y))/MM_{x=1}$$

- 8) Penggabungan $R_1(x,y)$, $G_1(x,y)$ dan $B_1(x,y)$ merupakan hasil akhir dari *encrypt image*

- 9) Untuk melakukan decoding dilakukan dengan cara meningkatkan intensitas cahaya yang telah diturunkan. Adapun formulanya dapat dituliskan seperti berikut ini :

$$R(x,y)=R_1(x,y)*25+R_1(M,y) \quad G(x,y)=G_1(x,y)*25+G_1(M,y)$$

$$B(x,y)=B_1(x,y)*25+B_1(M,y)$$

dimana : $x = 1..(M-1)$ dan $R_2(x,y)$, $G_2(x,y)$, $B_2(x,y)$ = Komponen *encrypt image* 26



Sebagai ilustrasi berikut adalah contoh matriks yang diambil dari sebagian image dimana setiap elemen matriks merupakan representasi dari nilai pixel image asli. Misalkan untuk mengkodekan menggunakan kunci 25

23018649265821 1274154265226256 533524912658577
 717439157 93626 4139100

Layer Red (R(x,y)) Layer Green (G(x,y)) Layer Blue (B(x,y))

Selanjutnya bagilah tiap pixel dengan kunci, dalam hal ini adalah 25. Hasil pembagian merupakan pembulatan kebawah Untuk lebih jelasnya flowchart untuk pengkodean ini dapat dilihat pada lampiran 1. Sebagai ilustrasi berikut adalah contoh matriks yang diambil dari sebagian image dimana setiap elemen matriks merupakan representasi dari nilai pixel image asli.

971128616 02610111211 219522254
 Layer Red (R₁(x,y)) Layer Green (G₁(x,y)) Layer Blue (B₁(x,y))

Setiap layer matriks ditambah satu baris yang berisi rata-rata sisa hasil bagi dari kunci dalam satu kolom Sehingga menghasilkan

10111697112861 1512202610111211 Layer Blue (B₁(x,y))
 6 Layer Green (G₁(x,y))

Layer Red (R₁(x,y)) 91110219522254

Dimensi dari encrypt image menjadi lebih besar daripada image asli karena adanya penambahan satu baris pixel yang berisi rata-rata sisa hasil bagi.

Untuk proses dekripsi kalikan setiap pixel pada baris kedua hingga terakhir dengan kunci (dalam hal ini 25) dan jumlahkan hasilnya dengan pixel pada baris pertama.

23518641356621

616036166

Layer Red (R₁(x,y))

156215226528727

653727

Layer Green

(G(x,y))

593623513461605

9136110

Layer Blue (B₁(x,y))

27

Jika dilihat dari hasil dekripsinya memang tidak sama persis dengan image aslinya, tetapi perubahan ini masih bisa diterima karena tidak mengubah gambar aslinya atau dengan kata lain kita masih dapat menginterpretasikan hasil dekripsi sama dengan gambar aslinya. Berikut ini adalah contoh gambar yang dikodekan dengan cara menurunkan intensitas cahaya dengan kunci “25B”.



Ukuran file : 3,21 Kb

Gambar 4. Hasil Pengkodean

Ukuran file : 14,8 Kb

Gambar 3. Gambar asli

Dari gambar di atas terlihat bahwa terjadi penurunan ukuran file dari 14,8 Kb menjadi 3,21 Kb atau mengalami penurunan sebesar 78.3 %. Hal ini merupakan suatu nilai penurunan yang sangat signifikan. Tetapi hasil pengkodean dengan menurunkan intensitas cahaya ini dapat dikatakan kurang baik karena dengan menambah intensitasnya masih dapat ditebak image asli yang dikodekannya

5.4. Sistem Pengkodean Berdasarkan Penurunan Intensitas dan Manipulasi Pixel

Karena sistem pengkodean berdasarkan manipulasi pixel mempunyai keunggulan pada hasil encrypt imagenya yang sulit untuk dipecahkan, sedangkan sistem pengkodean berdasarkan penurunan intensitas cahaya mempunyai keunggulan dalam menurunkan ukuran file, maka akan lebih baik bila kedua metode ini digabung. Adapun proses penggabungannya dapat dijelaskan sebagai berikut

- 1) Import data dari *image* yang akan dikodekan (dalam hal ini data berupa matriks berukuran $M \times N$)
- 2) Bagilah image kedalam blok berukuran 200 pixel x 200 pixel
- 3) Ekstrak pixel dari layer yang tidak ada pada kunci bagian 2, dalam hal ini adalah layer Red ($R_1(x,y)$) dan layer Green ($G_1(x,y)$), dimana (x,y) adalah posisi pixel

4) Bangkitkan nilai random antara 1 sampai 200 untuk setiap bloknya pada layer Blue ($B_1(x,y)$) dengan syarat tidak ada nilai yang sama dalam satu kolom.

5) Tukarkan $R_1(x,y)$ dengan $G_1(B_1(x,y),y)$

6) Membagi pixel dengan bagian pertama dari kunci (misalnya : 25)

$$R_2(x,y)=[R_1(x,y)/25] \quad G_2(x,y)=[G_1(x,y)/25] \quad B_2(x,y)=[B_1(x,y)/25]$$

7) Menambah satu kolom pada pixel pada *image*

$$R_2((x+1),y)=\Sigma(R_1(x,y)-R_2(x,y))/MM_{x=1} \quad G_2((x+1),y)=\Sigma(G_1(x,y)-G_2(x,y))/MM_{x=1}$$

$$B_2((x+1),y)=\Sigma(B_1(x,y)-B_2(x,y))/MM_{x=1}$$

8) Penggabungan $R_2(x,y)$, $G_2(x,y)$ dan $B_2(x,y)$ merupakan hasil akhir dari *encrypt image*

Sebagai ilustrasi berikut adalah contoh matriks yang diambil dari sebagian image dimana setiap elemen matriks merupakan representasi dari nilai pixel image asli.

23018649265821717439157

Layer Red ($R(x,y)$)

127415426522625693626

Layer Green ($G(x,y)$)

5335249126585774139100

Layer Blue ($B(x,y)$)

Selanjutnya dibangkitkan bilangan random pada layer Blue dengan syarat tidak boleh ada nilai yang sama dalam satu kolom, untuk menggantikan nilai pixel yang telah ada.

23018649265821717439157

Layer Red ($R(x,y)$)

127415426522625693626

Layer Green ($G(x,y)$)

312233121

Layer Blue ($B_1(x,y)$)

Tukarkan $R(x,y)$ dengan $G(B(x,y),y)$ sehingga menghasilkan

697425265362612226154

Layer Red ($R_1(x,y)$)

17418615726394923058217

Layer Green ($G_1(x,y)$)

312233121

Layer Blue ($B_1(x,y)$) 29

Selanjutnya bagilah tiap pixel dengan kunci, dalam hal ini adalah 25. Hasil pembagian merupakan pembulatan kebawah

2211011096	676111928	Layer Blue
Layer Red	Layer Green	(B ₂ (x,y))
(R ₂ (x,y))	(G ₂ (x,y))	
	000000000	

Setiap layer matriks ditambah satu baris yang berisi rata-rata sisa hasil bagi dari kunci dalam satu kolom sehingga menghasilkan

15122221101109

6

Layer Red

(R₁(x,y))

101116676111928

Layer Green

(G(x,y))

222000000000

Layer Blue

(B₁(x,y))

Dimensi dari encrypt image menjadi lebih besar daripada image asli karena adanya penambahan satu baris pixel yang berisi rata-rata sisa hasil bagi.

Untuk proses dekripsi kalikan setiap pixel pada baris kedua hingga terakhir dengan kunci (dalam hal ini 25) dan jumlahkan hasilnya dengan pixel pada baris pertama.

Berikut ini adalah contoh gambar yang dikodekan dengan cara memanipulasi pixel dan menurunkan intensitas cahaya, dengan kunci "25B".

Ukuran file : 4,74 Kb

Gambar 6. Hasil Pengkodean

Ukuran file : 14,8 Kb

Gambar 5. Gambar asli 3