Blind steganalysis pada Citra Digital dengan Metode *Support Vector Machine*

Marcelinus Henry M
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung, Indonesia
henrymenori@yahoo.com

Rinaldi Munir
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung, Indonesia
rinaldi@informatika.org

Abstrak—*Blind steganalysis* adalah salah satu metode untuk mendeteksi adanya pesan rahasia pada media tanpa perlu mengetahui algoritme steganografi yang digunakan. Citra digital diubah menjadi fitur dengan algoritme ekstraksi fitur *subtractive pixel adjacency matrix*. Fitur yang sudah dihasilkan dibuat modelnya dengan metode pembelajaran mesin *support vector machine*. Metode *support vector machine* yang digunakan mempunyai perbedaan pada *kernel* yang digunakan, yaitu linear, *polynomial*, dan RBF. Model yang sudah dibangun diuji untuk mengukur kinerja akurasi pendeteksian pesan rahasia dan estimasi panjang pesan rahasia. Akurasi yang didapat dari pengujian model cukup baik untuk mendeteksi adanya pesan rahasia dalam citra digital, namun belum cukup baik untuk melakukan estimasi panjang pesan. Akurasi tertinggi didapat dengan penggunaan *kernel polynomial*.

Kata kunci—*blind steganalysis*, *support vector machine*, citra digital, ekstraksi fitur.

I. PENDAHULUAN

Steganografi adalah sebuah teknik untuk menyelipkan pesan rahasia pada sebuah media tertentu. Media yang umum digunakan yaitu gambar, video, suara, dan teks. Pesan rahasia yang disisipkan biasanya berupa teks, namun tidak menutup kemungkinan berupa media lain seperti gambar dan suara. Media yang sering digunakan dalam penyisipan pesan rahasia adalah gambar. Gambar terdiri dari *pixel* yang masing-masing memiliki *bit* warna. Teknik steganografi yang sering digunakan yaitu metode domain spasial, yaitu pada suatu gambar disembunyikan sebuah pesan dengan mengganti beberapa *bit* pada gambar tersebut. Metode ini cukup populer karena sedikit mengubah gambar aslinya dan pesan yang dapat disisipkan memiliki ukuran yang cukup besar [1].

Pada zaman sekarang steganografi semakin berkembang sehingga semakin banyak orang yang mulai berpikir bagaimana cara untuk membalikkan prosesnya. Teknik ini disebut steganalisis. Steganalisis dapat dibedakan menjadi 2, yaitu *targeted steganalysis* dan *blind steganalysis* [2]. *Targeted steganalysis* dilakukan dengan membalikkan algoritme steganografi sehingga dapat diketahui apakah ada pesan tersembunyi di dalam sebuah citra. Namun, *targeted steganalysis* membutuhkan informasi mengenai algoritme apa yang digunakan dalam penyisipan pesan rahasia ke dalam citra tersebut. *Blind steganalysis* dikembangkan karena tidak setiap citra diketahui metode penyisipan pesan rahasianya. Metode ini tidak selalu akurat, namun sangat berguna apabila kita tidak memiliki informasi apapun mengenai algoritme steganografi yang digunakan. Selain mendeteksi adanya pesan rahasia, *blind steganalysis* juga dikembangkan untuk mendeteksi atribut-atribut penting dari pesan rahasia seperti panjang pesan rahasia dan algoritme yang digunakan untuk sampai kepada tujuan akhir dari steganalisis yaitu mengetahui isi pesan rahasia tersebut [3].

Teknik *blind steganalysis* bisa digabungkan dengan pembelajaran mesin untuk mendapatkan hasil yang lebih baik. Steganalisis memerlukan metode pembelajaran mesin yang berupa klasifikasi biner untuk menentukan apakah sebuah citra digital memiliki pesan rahasia atau tidak. Metode pembelajaran mesin yang berupa klasifikasi biner adalah *Support Vector Machine* [4]. *Support vector machine* juga sangat baik untuk klasifikasi dengan fitur yang banyak, yang pada sebuah citra digital, dapat diekstrak banyak fitur yang bisa digunakan untuk mengetahui pesan rahasia di dalam citra digital tersebut.

II. DASAR TEORI

A. Citra Digital

Citra digital adalah salah satu media yang bisa disisipi pesan rahasia. Citra digital merupakan kumpulan *pixel* yang memiliki nilai intensitas yang menggambarkan keabuan atau warna dari *pixel* tersebut. Pada penyisipan pesan, citra digital tak terkompresi sangat mendukung penyisipan pesan karena nilai-nilai *pixel* bisa langsung diubah dan dimanipulasi sesuai pesan yang akan disisipkan. Citra tak terkompresi yang biasa digunakan untuk penyisipan pesan yaitu citra 24-*bit* seperti BMP.

Pada citra digital, setiap *pixel* memiliki beberapa nilai yang menggambarkan intensitas. *Pixel* yang memiliki satu nilai disebut *grayscale*, atau citra hitam putih. *Pixel* yang memiliki tiga nilai menggambarkan *red*, *green*, dan *blue*, atau citra berwarna. *Pixel* yang memiliki empat nilai adalah citra berwarna dengan tambahan satu nilai yaitu nilai transparan. Setiap nilai yang ada pada *pixel* memiliki rentang dari 0 sampai 255.

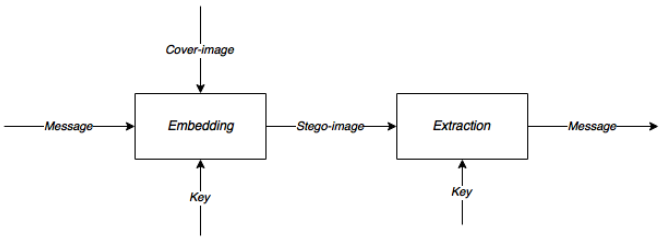
B. *Steganografi*

Steganografi adalah teknik untuk menyisipkan pesan rahasia ke dalam sebuah media tertentu sehingga tidak dicurigai keberadaan pesan rahasia tersebut. Steganografi dan kriptografi memiliki perbedaan. Kriptografi bertujuan agar pesan yang diterima tidak dapat dimengerti apabila orang tersebut tidak memiliki kunci yang dibutuhkan, sedangkan steganografi bertujuan untuk menyembunyikan pesan rahasia tersebut sehingga keberadaannya tidak diketahui dan tidak dicurigai oleh pihak yang tidak berkepentingan [1].

Proses penyisipan pesan dimulai dengan mempersiapkan *cover image* atau citra yang akan disisipi pesan. Citra ini sebisa mungkin tidak menimbulkan kecurigaan. Kemudian lakukan enkripsi terhadap pesan rahasia dengan menggunakan sebuah *key* tertentu. Hal ini dilakukan untuk mencegah diketahuinya pesan rahasia oleh pihak lain apabila steganografi gagal atau pihak lain mendeteksi adanya pesan rahasia. Kemudian, tahap terakhir yaitu menyisipkan pesan yang sudah dienkripsi dengan sebuah algoritme steganografi sehingga dihasilkan *stego image* atau citra yang sudah disisipi pesan rahasia. Proses penyisipan pesan rahasia dalam sebuah media disebut *embedding,* sedangkan proses untuk mendapatkan pesan rahasia dari sebuah *stego image* disebut *extraction.* Gambar 1 memperlihatkan proses penyisipan dan ekstraksi pesan rahasia dengan media citra digital.

Tabel 1. Contoh pesan, cover-object, dan stego-object

Embedded message	***Cover-object***	***Stego -object***
"Siapkan pesawat tempur dan perlengkapan penyerangan sekarang"		



Gambar 1. Diagram alur penyisipan dan ekstraksi pesan

Teknik penyisipan pesan rahasia ke dalam sebuah media dapat dilakukan dalam dua ranah [1].

1. Ranah spasial

Teknik ini dilakukan dengan mengubah *bit* dari *pixel* citra yang merepresentasikan warna. Contoh metode ranah spasial adalah metode LSB.

2. Ranah transform

Teknik ini dilakukan dengan mengubah frekuensi sinyal dengan suatu fungsi matematika. Sinyal dalam ranah spasial diubah menjadi ranah frekuensi dengan menggunakan transformasi *Discrete Cosine Transform* (DCT), *Discrete Fourier Transform* (DFT), dan *Discrete Wavelet Transform* (DWT).

C. *Steganalisis*

Steganalisis adalah teknik untuk mendeteksi adanya pesan rahasia dalam suatu media. Penyisipan suatu informasi pada suatu media tertentu akan mengubah karakteristik media tersebut sehingga bisa dideteksi melalui beberapa teknik.

Metode steganalisis dibagi menjadi dua [2], yaitu sebagai berikut.

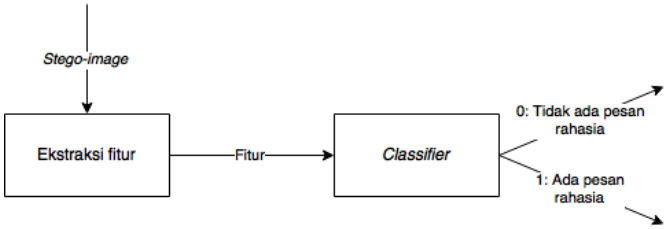
1. *Specific Technique Steganalysis* (*Targeted Steganalysis*)

Steganalisis untuk metode steganografi tertentu seperti LSB (*Least Significant Byte*). Metode ini memiliki akurasi yang besar apabila metode yang digunakan dalam steganografi sama dengan metode steganalisis tersebut.

2. *Blind steganalysis*

Metode steganalisis ini tidak menekankan pada suatu metode steganografi tertentu, tetapi untuk semua metode steganografi. Metode ini digunakan dengan menganalisis perubahan pada *bit pixel* ataupun dengan statistik.

Proses steganalisis berawal dari sebuah *stego-image* yang dicurigai telah disisipi pesan rahasia. Dari *stego-image*, fitur diekstrak dengan metode ektstraksi fitur yang sudah didefinisikan. Kemudian, fitur-fitur yang sudah didapat dimasukkan ke dalam sebuah *classifier* untuk ditentukan apakah citra tersebut memiliki pesan rahasia atau tidak. Hasil lain dari *classifier* mungkin berupa panjang pesan, ataupun algoritme steganografi yang digunakan untuk menyisipkan pesan tersebut. Proses steganalisis dapat dilihat pada Gambar 2.



Gambar 2. Diagram alur proses steganalisis

D. *Support Vector Machine*

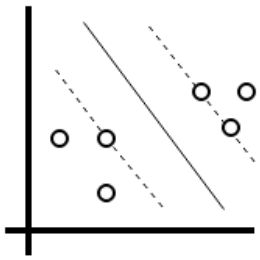
Support Vector Machine (SVM) adalah sebuah metode pembelajaran mesin yang didasarkan pada klasifikasi biner. Klasifikasi biner adalah metode untuk membagi data menjadi dua kelas (biner) dimana setiap data akan mempunyai nilai kelas +1 atau -1. Untuk setiap data $(\vec{x}_i, y_i)$ dengan $i = 1 \ldots N$, $\vec{x}_i \in R^d$, dan $y_i \in \{-1, +1\}$, klasifikasi biner $f(\vec{x}_i)$ yang dihasilkan adalah sebagai berikut

$$y_i = \begin{cases} +1, & f(\vec{x}_i) \geq 0 \\ -1, & f(\vec{x}_i) < 0 \end{cases} \qquad (1)$$

$\vec{x}_i$ adalah data latih ke-*i* yang adalah kumpulan bilangan real *a* sebagai atributnya dan *k* adalah jumlah atribut dalam data tersebut.

$$\vec{x}_i = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix} \quad (2)$$

Metode SVM akan membentuk *support vector* berdasarkan data yang paling dekat dengan bidang pembagi sehingga akan dibentuk satu *support vector* untuk masing-masing kelas. *Support vector* ini akan membantu dalam melakukan klasifikasi dalam menentukan keyakinan. Apabila data terletak di antara *support vector* tersebut, maka data diklasifikasikan dengan keyakinan yang lebih rendah dibandingkan dengan data di bawah atau di atas *support vector*. Gambar 3 memperlihatkan representasi *support vector* yang digambarkan oleh garis putus-putus.



Gambar 3. Representasi support vector

Berikut adalah tahap-tahap untuk mendapat fungsi pembagi dan *support vector* sehingga terbentuk model [5].

1. Setiap data memiliki nilai α. Nilai tersebut merepresentasikan besar pengaruh data tersebut terhadap fungsi pembagi dan *support vector*.

2. Hitung nilai α untuk setiap data agar nilai L_D maksimal dalam persamaan berikut.

$$L_D(\alpha) \equiv \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j k\left(\vec{x}_i \vec{x}_j\right) \quad (3)$$

Dengan syarat $\sum_{i=1}^{n} \alpha_i y_i = 0$ dan $c \geq \alpha_i \geq 0$, dimana c adalah suatu konstanta yang sudah ditentukan. *Kernel* yang digunakan yaitu linear $k(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j$, *polynomial* $k(\vec{x}_i, \vec{x}_j) = \left(\vec{x}_i \cdot \vec{x}_j\right)^2$, dan RBF $k(\vec{x}_i, \vec{x}_j) = -\gamma \left\| \vec{x}_i - \vec{x}_j \right\|^2$

3. Setelah L_D memiliki nilai yang maksimal, simpan setiap data yang memiliki nilai $\alpha > 0$. Data tersebut akan menjadi *support vector*.

4. Fungsi pembagi adalah:

$$f(\vec{x}_d) = \sum_{i=1}^{ns} \alpha_i y_i \vec{x}_i \vec{x}_d + b \quad (4)$$

5. Untuk melakukan klasifikasi terhadap sebuah data *x*, persamaan (4) digunakan untuk kalkulasi terlebih dahulu. Setelah itu, persamaan (1) digunakan untuk melakukan klasifikasi.

Untuk menghitung nilai maksimal dari L_D, bisa digunakan beberapa algoritme, salah satunya adalah algoritme *modified sequential minimal optimization* [6].

Support vector machine merupakan *binary classifier*, namun tidak menutup kemungkinan untuk membuat model *multi-class* dengan metode *support vector machine*. Model *multi-class* untuk *support vector* machine terdiri dari beberapa sub-model. Beberapa cara untuk membentuk model *multi-class support vector machine* yaitu *one-against-all*, *one-against-one*, dan *directed acyclic graph* [4].

III. PENELITIAN TERKAIT

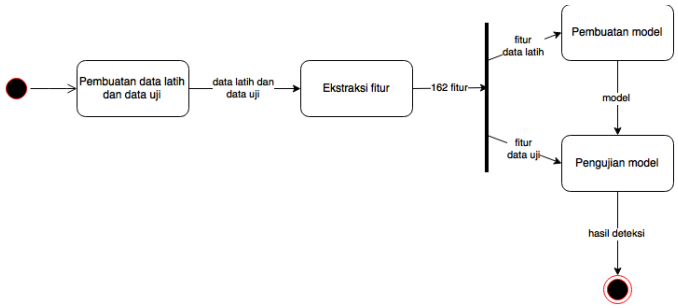
Metode steganalisis yang diajukan oleh Siwei dan Hany [7] menggunakan *color wavelet statistics* sebagai ekstraksi fitur dan *support vector machine* sebagai *classifier*. Metode ini bisa dilakukan di ranah spasial maupun ranah transform. Namun, akurasi pada ranah spasial lebih rendah dibandingkan ranah transform karena metode ekstraksi fitur yang sangat dekat dengan ranah transform.

Metode steganalisis yang diajukan oleh Jiang [8] menggunakan campuran dari beberapa metode. Untuk ekstraksi fitur digunakan *filter* berupa *shifting* dan juga *first order statistics*. Untuk *classifier*, digunakan *ensemble classifier* dengan tambahan *AdaBoost* dan *bagging*. Metode steganalisis ini digunakan di dalam ranah spasial dan dibuat untuk mengatasi kecilnya akurasi di bagian penyisipan teks dalam ukuran kecil. Namun metode ekstraksi fitur yang dipakai memiliki dimensi yang cukup besar.

Metode steganalisis yang diajukan oleh Tomas [9] ini menggunakan *substractive pixel adjacency matrix* sebagai metode ekstraksi fitur dan *support vector machine* sebagai *classifier*. Metode steganalisis ini dilakukan di ranah spasial. Metode ini lebih menekankan pada ekstraksi fitur dan deteksi terhadap *pixel* yang mencurigakan. Metode *support vector machine* yang digunakan hanya didasarkan pada *kernel* RBF.

IV. ALUR KERJA

Citra digital yang digunakan berasal dari internet dengan berbagai ukuran dan tipe. Kemudian akan digunakan alur kerja seperti pada Gambar 4.



Gambar 4. Diagram alur kerja

Pada tahap pembuatan data latih dan data uji akan dibuat data latih dan data uji untuk pembelajaran dan pengujian model. Data latih dan data uji merupakan *cover-image* dan *stego-image* dari citra digital yang dikumpulkan sebelumnya. Citra digital pada

data latih dan data uji dikonversi menjadi ukuran dan tipe yang sama yaitu 800x600 *pixel* dengan tipe *bitmap*. Untuk data latih, *stego-image* didapat dengan menyisipkan pesan dengan ukuran 0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, dan 0.95 bpp. Untuk data uji *stego-image* didapat dengan menyisipkan pesan dengan ukuran yang acak antara 0.01 – 0.99 bpp. Penyisipan pesan ke dalam citra digital menggunakan kakas steganografi Steghide [10], Four Pixel [11], dan Nine *Pixel* [12].

Tahap ekstraksi fitur adalah proses mengambil fitur-fitur penting dari citra digital pada data latih dan data uji. Algoritme ekstraksi fitur yang digunakan yaitu *substractive pixel adjacency matrix* yang diajukan oleh Tomas [9]. Algoritme dari *subtractive pixel adjacency matrix* adalah sebagai berikut.

1. Hitung nilai matriks selisih *D* pada citra digital berukuran *mxn* pada arah kanan dengan rumus

$$D^{\rightarrow}_{i,j} = I_{i,j} - I_{i,j+1} \qquad (5)$$

dengan batasan $i \in \{1,\ldots,m\}$ dan $j \in \{1,\ldots,n-1\}$. I adalah nilai *pixel* yaitu nilai *grayscale* untuk citra digital *grayscale* dan nilai rata-rata *red*, *green*, dan *blue* untuk citra digital berwarna Contoh pembuatan matriks selisih pada citra berukuran 4x4 *pixel* dapat dilihat pada Gambar 5.

5	3	3	3
2	1	2	5
5	4	3	2
1	3	7	4

⇨

2	0	0
1	-1	-3
1	1	1
-2	-4	3

Gambar 5. Contoh pembuatan matriks selisih pada citra 4x4 pixel

2. Kemudian hitung nilai Markov chain dengan rumus

$$M^{\rightarrow}_{u,v} = \Pr\left(D^{\rightarrow}_{i,j+1} = u \mid D^{\rightarrow}_{i,j} = v\right) \qquad (6)$$

dengan batasan $u,v \in \{-4,\ldots,4\}$. Jika $\Pr\left(D^{\rightarrow}_{i,j} = v\right) = 0$ maka $M^{\rightarrow}_{u,v} = \Pr\left(D^{\rightarrow}_{i,j+1} = u \mid D^{\rightarrow}_{i,j} = v\right) = 0$. Contoh perhitungan Markov chain pada Gambar Gambar 5 adalah $M^{\rightarrow}_{0,0} = \Pr\left(D^{\rightarrow}_{i,j+1} = 0 \mid D^{\rightarrow}_{i,j} = 0\right) = 1$ dan $M^{\rightarrow}_{1,1} = \Pr\left(D^{\rightarrow}_{i,j+1} = 1 \mid D^{\rightarrow}_{i,j} = 1\right) = \frac{2}{3}$.

3. Ulangi untuk arah lainnya

4. Satukan fitur *F* dari delapan arah dengan rumus

$$F_{1,\ldots,k} = \frac{1}{4}\left[M^{\rightarrow} + M^{\leftarrow} + M^{\downarrow} + M^{\uparrow}\right] \qquad (7)$$

$$F_{k+1,\ldots,2k} = \frac{1}{4}\left[M^{\nearrow} + M^{\swarrow} + M^{\searrow} + M^{\nwarrow}\right] \qquad (8)$$

sehingga dihasilkan 162 fitur untuk setiap citra digital.

Tahap pembelajaran dengan *support vector machine* membutuhkan masukan fitur data latih. Dari kumpulan fitur akan dibuat dua buah model. Model yang pertama digunakan untuk menentukan apakah citra digital memiliki pesan rahasia, sedangkan model yang kedua digunakan untuk memperkirakan panjang pesan rahasia pada citra digital. Model pertama memiliki pilihan konfigurasi *kernel* linear, *polynomial*, dan RBF, sedangkan model kedua memiliki konfigurasi tambahan *multi-class one-against-all*, *one-against-one*, dan *directed acyclic graph*. Pembelajaran model dengan metode *support vector machine* ini menggunakan algoritme *modified sequential minimal optimization* yang diajukan oleh Cao [6].

Tahap pengujian membutuhkan masukan model yang telah dibuat. Model yang telah dibuat akan diuji dengan data latih dan data uji untuk mengukur kinerjanya. Setiap pengujian akan menghasilkan sebuah nilai akurasi.

V. EKSPERIMEN

Proses pembelajaran dilakukan dengan iterasi maksimal 30000, konstanta *c* 100, dua kelas untuk pendeteksian pesan rahasia yaitu yes dan no, lima kelas untuk estimasi panjang pesan rahasia yaitu *very low* (<0.2 bpp), *low* (0.2 bpp – 0.4 bpp), *medium* (0.4 bpp – 0.6 bpp), *high* (0.6 bpp – 0.8 bpp), dan *very high* (>0.8 bpp).

Pengujian dibagi menjadi empat bagian, yaitu pengujian untuk pendeteksian pesan pada citra digital *grayscale*, pengujian untuk estimasi panjang pesan pada citra digital *grayscale*, pengujian untuk pendeteksian pesan pada citra digital berwarna, dan pengujian untuk estimasi pesan pada citra digital berwarna. Akurasi pendeteksian pesan dan estimasi panjang pesan dihitung dengan menggunakan persamaan

$$\text{akurasi} = \frac{\text{jumlah citra digital yang berhasil ditebak}}{\text{jumlah citra digital pada data uji}} \qquad (9)$$

Tabel 2. Hasil pengujian pendeteksian pesan pada citra digital grayscale

Kernel	Akurasi dengan Data Latih	Akurasi dengan Data Uji
Linear	67.25%	63%
Polynomial	75.25%	73%
RBF	69.5%	59%

Tabel 3 Hasil pengujian pendeteksian pesan pada citra digital berwarna

Kernel	Akurasi dengan Data Latih	Akurasi dengan Data Uji
linear	54.75%	57%
polynomial	64.25%	61%
RBF	51.5%	50%

Tabel 4 Hasil pengujian estimasi panjang pesan pada citra digital grayscale

Kernel	*Multi-class*	Akurasi dengan Data Latih	Akurasi dengan Data Uji
linear	*one-against-all*	5.2%	4%
linear	*one-against-one*	28.5%	24.8%
linear	*directed acyclic graph*	29.15%	24.8%
polynomial	*one-against-all*	6.45%	6.8%
polynomial	*one-against-one*	39.4%	33.6%
polynomial	*directed acyclic graph*	38.15%	30%
RBF	*one-against-all*	7.95%	6%
RBF	*one-against-one*	30.95%	25.2%
RBF	*directed acyclic graph*	30.55%	24.4%

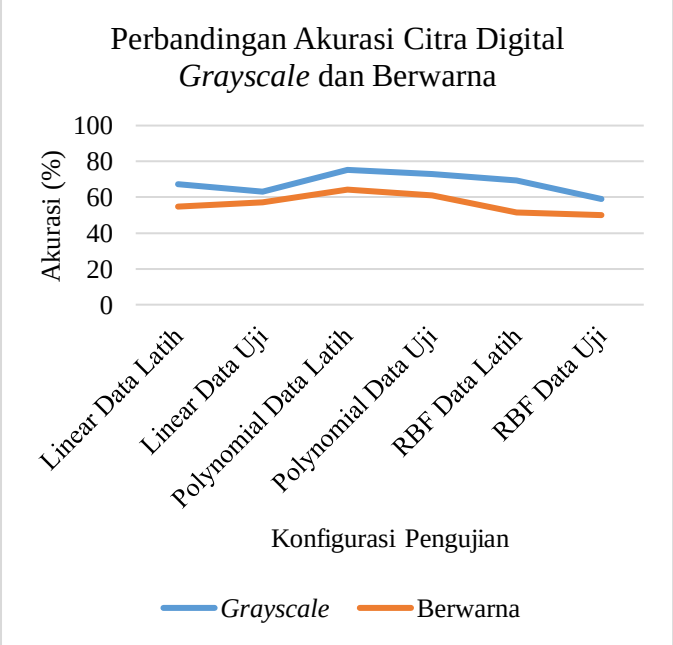
Tabel 5 Hasil pengujian estimasi panjang pesan pada citra digital berwarna

Kernel	*Multi-class*	Akurasi dengan Data Latih	Akurasi dengan Data Uji
linear	*one-against-all*	3.15%	2%
linear	*one-against-one*	21.35%	23.6%
linear	*directed acyclic graph*	22.3%	21.6%
polynomial	*one-against-all*	0%	0%
polynomial	*one-against-one*	26.35%	23.6%
polynomial	*directed acyclic graph*	26.05%	22%
RBF	*one-against-all*	5.25%	4%
RBF	*one-against-one*	20.9%	20.8%
RBF	*directed acyclic graph*	21.05%	20.8%

Hasil pengujian pendeteksian pesan rahasia pada citra digital *grayscale* menunjukkan konfigurasi dengan *kernel polynomial* memiliki akurasi yang paling tinggi baik dengan data latih maupun data uji. Konfigurasi *kernel polynomial* memang lebih baik karena *polynomial* lebih fleksibel dibandingkan linear dan lebih kaku dari RBF. Bidang pembatas yang dibuat *polynomial* bukanlah bidang datar, melainkan bergelombang menyesuaikan dengan data, sedangkan bidang pembatas yang dibuat linear lebih kaku sehingga tidak terlalu menyesuaikan dengan data. Bidang pembatas yang dibuat dengan *kernel* terlalu fleksibel sehingga membuat model menjadi *overfit*. *Overfit* adalah keadaan saat model yang dihasilkan terlalu bergantung pada data latihnya sehingga tidak tercipta model yang lebih umum. Hasil pengujian pendeteksian pesan rahasia pada citra digital berwarna juga menujukkan konfigurasi dengan *kernel polynomial* memiliki akurasi yang paling tinggi. Model yang umum diperlukan agar *noise* pada data tidak mempengaruhi model.

Hasil pengujian estimasi panjang pesan rahasia pada citra digital *grayscale* menunjukkan konfigurasi dengan *kernel polynomial* memiliki akurasi yang paling tinggi saat diuji dengan data latih dan konfigurasi dengan *kernel polynomial* dan *multi-class one-against-one* memiliki akurasi yang paling tinggi saat diuji dengan data uji. Hasil pengujian estimasi panjang pesan rahasia pada citra digital berwarna juga menunjukkan hasil yang sama.

Untuk perbandingan akurasi pengujian citra digital *grayscale* dan citra digital berwarna dapat dilihat pada Gambar 6.



Gambar 6 Perbandingan akurasi citra digital grayscale dan berwarna

Perbandingan akurasi menunjukkan bahwa akurasi pengujian citra digital *grayscale* secara umum lebih tinggi dari akurasi pengujian citra digital berwarna. Hal ini disebabkan perbedaan dalam ekstraksi fitur untuk citra digital *grayscale* dan

citra digital berwarna. Dalam citra digital *grayscale*, selisih antar *pixel* dapat dihitung dengan mudah karena setiap *pixel* hanya mengandung satu nilai pada kisaran 0 – 255, sedangkan untuk citra digital berwarna, setiap *pixel* mengandung tiga nilai masing-masing untuk *red*, *green*, dan *blue*. Perubahan nilai *pixel* terhadap nilai *pixel* di sekitarnya menjadi lebih sulit dideteksi karena perbedaan jumlah nilai yang dikandung tiap *pixel*, Misalkan, nilai *pixel* pada *cover-image* adalah 12, 15, dan 120 sedangkan pada *stego-image* 13, 12, dan 122. Jika dihitung perbedaan nilai *pixel* pada *cover-image* dan pada *stego-image* didapat nilai nol, hal tersebut karena nilai *grayscale* dari dua *pixel* tersebut adalah sama.

Akurasi estimasi panjang pesan tidak cukup baik dikarenakan metode ekstraksi fitur yang digunakan tidak cocok. *Subtractive pixel adjacency matrix* tidak hanya menunjukkan nilai probabilitas yang berubah. Sebagai akibatnya, apabila ada pesan rahasia, pesan rahasia tersebut bisa diketahui dengan metode *support vector machine*. *Subtractive pixel adjacency matrix* tidak menunjukkan seberapa besar perubahan *pixel* yang terjadi sehingga tidak cocok untuk estimasi panjang pesan rahasia.

VI. SIMPULAN

Blind steganalysis dengan metode *support vector machine* dapat diimplementasikan untuk mendeteksi pesan rahasia dan melakukan estimasi panjang pesan rahasia pada citra digital.

Model pembelajaran dengan metode *support vector machine* yang dibangun memiliki hasil yang cukup baik dalam mendeteksi pesan rahasia dengan akurasi sebesar 73% untuk citra digital *grayscale* dan 61% untuk citra digital berwarna. Estimasi panjang pesan rahasia pada citra digital memiliki hasil yang kurang baik. Akurasi paling tinggi yang dicapai yaitu sebesar 33.6% untuk citra digital *grayscale* dan 23.6% untuk citra digital berwarna. Konfigurasi dengan *kernel polynomial* lebih baik dibandingkan dua *kernel* lainnya yaitu linear dan RBF.

ACKNOWLEDGMENT

Penulis ingin mengucapkan terima kasih kepada Dr. Ir. Rinaldi Munir, M.T. selaku dosen pembimbing karena telah memberikan bimbingan dan ilmu yang sangat berharga selama penelitian.

REFERENCES

[1] Hussain, M., & Hussain, M. (2013). A survey of image steganography techniques.

[2] Chandramouli, R., Kharrazi, M., & Memon, N. (2003, October). Image steganography and *steganalysis*: Concepts and practice. In *International Workshop on Digital Watermarking* (pp. 35-49). Springer Berlin Heidelberg.

[3] Fridrich, J., Goljan, M., Hogea, D., & Soukal, D. (2003). Quantitative steganalysis of digital images: estimating the secret message length. *Multimedia systems*, *9*(3), 288-302.

[4] Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass *support vector machine*s. *IEEE transactions on Neural Networks*, *13*(2), 415-425.

[5] Schölkopf, B., & Smola, A. J. (2002). Learning with *kernel*s: *support vector* machines, regularization, optimization, and beyond. MIT press.

[6] Cao, L. J., Keerthi, S. S., Ong, C. J., Zhang, J. Q., Periyathamby, U., Fu, X. J., & Lee, H. P. (2006). Parallel sequential minimal optimization for the training of *support vector machine*s. *IEEE Transactions on Neural Networks*, *17*(4), 1039-1049.

[7] Lyu, S., & Farid, H. (2004, June). *Steganalysis* using color wavelet statistics and one-class *support vector machine*s. In *Electronic Imaging 2004* (pp. 35-45). International Society for Optics and Photonics.

[8] Yu, J., Zhang, X., & Li, F. (2015). Spatial *steganalysis* using redistributed residuals and diverse ensemble classifier. *Multimedia Tools and Applications*, 1-13.

[9] Pevny, T., Bas, P., & Fridrich, J. (2010). Steganalysis by subtractive *pixel* adjacency matrix. IEEE Transactions on Information Forensics and Security, 5(2), 215-224.

[10] *http://steghide.sourceforge.net/*

[11] Liao, X., Wen, Q. Y., & Zhang, J. (2011). A steganographic method for digital images with four-*pixel* differencing and modified LSB substitution. *Journal of Visual Communication and Image Representation*, *22*(1), 1-8.

[12] Swain, G. (2014). Digital image steganography using nine-*pixel* differencing and modified LSB substitution. *Indian Journal of Science and Technology*, *7*(9), 1444-1450.