

Steganografi dan *Watermarking* pada Citra Digital

Steganografi (*steganography*) adalah teknik menyembunyikan data rahasia di dalam wadah (media) digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang. Steganografi membutuhkan dua properti: wadah penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara (audio), teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video.

Penggunaan steganografi antara lain bertujuan untuk menyamarkan eksistensi (keberadaan) data rahasia sehingga sulit dideteksi, dan melindungi hak cipta suatu produk. Steganografi dapat dipandang sebagai kelanjutan kriptografi. Jika pada kriptografi, data yang telah disandikan (*ciphertext*) tetap tersedia, maka dengan steganografi cipherteks dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaannya. Data rahasia yang disembunyikan dapat diekstraksi kembali persis sama seperti keadaan aslinya.


Bab ini akan memaparkan steganografi dan *watermarking* pada citra digital. *Watermarking* adalah aplikasi dari steganografi, di mana citra digital diberi suatu penanda yang menunjukkan label kepemilikan citra tersebut. Sebagian besar dari materi bab ini dikutip dari [POL98].

13.1 Sejarah Steganografi

Steganografi sudah dikenal oleh bangsa Yunani. Penguasa Yunani dalam mengirimkan pesan rahasia menggunakan kepala budak atau prajurit sebagai media. Dalam hal ini, rambut budak dibotaki, lalu pesan rahasia ditulis pada kulit kepala budak. Ketika rambut budak tumbuh, budak tersebut diutus untuk membawa pesan rahasia di kepalanya.

Bangsa Romawi mengenal steganografi dengan menggunakan tinta tak-tampak (*invisible ink*) untuk menuliskan pesan. Tinta tersebut dibuat dari campuran sari buah, susu, dan cuka. Jika tinta digunakan untuk menulis maka tulisannya tidak tampak. Tulisan di atas kertas dapat dibaca dengan cara memanaskan kertas tersebut.

Sebagai contoh ilustrasi, Gambar 13.1.a adalah citra lada (*peppers.bmp*) yang akan digunakan untuk menyembunyikan sebuah dokumen teks (Gambar 13.1.b) yang berukuran 20 KB). Perhatikanlah citra lada sebelum penyembuan data (13.1.a) dan citra setelah disisipi data teks (13.1.c). Citra lada tetap kelihatan mulus, seolah-olah tidak pernah disisipi data sebelumnya. Sebenarnya tidaklah demikian, gambar lada tersebut mengalami *sedikit* perubahan akibat steganografi, namun mata manusia mempunyai sifat kurang peka terhadap perubahan kecil ini, sehingga manusia sukar membedakan mana gambar yang asli dan mana gambar yang sudah disisipi data.

 <p>(a) Citra <i>peppers</i> asli</p>	<p style="text-align: center;">LETTER OF RECOMMENDATION</p> <p>To Whom It May Concern,</p> <p>Herewith I highly recommend Mr. R. Hendro Wicaksono continue his postgraduate study at your university. My recommendation is based on my experience as his lecturer in several courses for the past four years.</p> <p>He has shown me his excellent attitude and personality. He is a hard working person and he has a lot of creative ideas. He is also a very intelligent student and he cooperates very well with his peers whenever they had to work together.</p> <p>During his study, he showed diligence and eagerness to achieve his goal. He sets very high standard for himself and organizes himself very well to achieve the standard. I am confident that if he can maintain his goal work, he should be able to complete the postgraduate program well within the stipulated time.</p> <p>I am sure that his abilities and his personal qualities along with his academic capabilities will help his to obtain his Master's degree at your university, which will be very useful for our country.</p> <p>Bandung, November 15, 2002 Yours Sincerely,</p> <p>Ir. Rinaldi Munir, M.Sc. Senior Lecturer Informatics Engineering Department, Institute Technology of Bandung (ITB) Jl. Ganesha No. 10, Bandung 40132 Email : rinaldi@informatika.org Phone +62-22-2508135 Indonesia</p> <p>(b) Dokumen <i>hendro.doc</i> yang akan disembunyikan ke daalm citra lada</p>
 <p>(c) Citra <i>peppers</i> setelah “diisi” dengan data teks <i>hendro.doc</i></p>	

Gambar 13.1 Contoh penyembunyian data di dalam citra digital

13.2 Kriteria Steganografi yang Bagus

Seperti sudah disebutkan pada bagian awal bab, data yang disembunyikan tidak hanya berupa teks, tetapi juga berupa citra, audio, atau video. Selain citra digital, media penampung data rahasia juga bisa berupa teks, audio, atau video. Namun di sini kita membatasi media penampung hanya citra digital saja.

Penyembunyian data rahasia ke dalam citra digital akan mengubah kualitas citra tersebut. Kriteria yang harus diperhatikan dalam penyembunyian data adalah:

1. *Fidelity*. Mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
2. *Robustness*. Data yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi yang dilakukan pada citra penampung, seperti pengubahan kontras, penajaman, pemampatan, rotasi, perbesaran gambar, pemotongan (*cropping*), enkripsi, dan sebagainya. Bila pada citra penampung dilakukan operasi-operasi pengolahan citra tersebut, maka data yang disembunyikan seharusnya tidak rusak (tetap valid jika diekstraksi kembali)
3. *Recovery*. Data yang disembunyikan harus dapat diungkapkan kembali (*reveal*). Karena tujuan steganografi adalah data hiding, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

13.3 Teknik Penyembunyian Data

Penyembunyian data dilakukan dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit data rahasia. Hingga saat ini sudah banyak dikemukakan oleh para ilmuwan metode-metode penyembunyian data. Metode yang paling sederhana adalah metode modifikasi *LSB* (*Least Significant Bit Modification*). Pada susunan bit di dalam sebuah *byte* (1 *byte* = 8 bit), ada bit yang paling berarti (*most significant bit* atau *MSB*) dan bit yang paling kurang berarti (*least significant bit* atau *LSB*). Sebagai ilustrasi, di bawah ini dijelaskan metode modifikasi LSB untuk menyisipkan *watermark* pada citra (gambar) digital.

Misalnya pada *byte* 11010010, bit 1 yang pertama (digarisbawahi) adalah bit *MSB* dan bit 0 yang terakhir (digarisbawahi) adalah bit *LSB*. Bit yang cocok untuk diganti adalah bit *LSB*, sebab penggantian hanya mengubah nilai *byte* tersebut satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan *byte* tersebut di dalam gambar menyatakan warna tertentu, maka perubahan satu bit *LSB* tidak mengubah warna tersebut secara berarti. Lagi pula, dan ini keuntungan yang dimanfaatkan, mata manusia tidak dapat membedakan perubahan yang kecil.

Misalkan segmen *pixel-pixel* citra sebelum penambahan bit-bit *watermark* adalah

00110011 10100010 11100010 01101111

Misalkan data rahasia (yang telah dikonversi ke sistem biner) adalah 0111. Setiap bit dari *watermark* menggantikan posisi *LSB* dari segmen data citra menjadi:

00110010 10100011 11100011 01101111

Untuk memperkuat penyembunyian data, bit-bit data tidak digunakan untuk mengganti *byte-byte* yang berurutan, namun dipilih susunan *byte* secara acak. Misalnya jika terdapat 50 *byte* dan 6 bit data yang akan disembunyikan, maka *byte* yang diganti bit *LSB*-nya dipilih secara acak, misalkan *byte* nomor 36, 5, 21, 10, 18, 49.

Bilangan acak dibangkitkan dengan *pseudo-random-number-generator (PRNG)*. *PRNG* menggunakan kunci rahasia untuk membangkitkan posisi *pixel* yang akan digunakan untuk menyembunyikan bit-bit. *PRNG* dibangun dalam beberapa cara, salah satunya dengan menggunakan algoritma kriptografi *DES (Data Encryption Standard)*, algoritma *hash MD5*, dan mode kriptografi *CFB (Cipher-Feedback Mode)*. Tujuan dari enkripsi adalah menghasilkan sekumpulan bilangan acak yang sama untuk setiap kunci enkripsi yang sama. Bilangan acak dihasilkan dengan cara memilih bit-bit dari sebuah blok data hasil enkripsi.

Teknik penyembunyian data untuk citra 8-bit berbeda dengan citra 24-bit. Seperti sudah dijelaskan di dalam Bab 3, berkas citra *bitmap* terdiri atas bagian *header*, palet *RGB*, dan data *bitmap*.

Pada citra 8-bit, setiap elemen data *bitmap* menyatakan indeks dari peta warnanya di palet *RGB*. Pada citra 24-bit, tidak terdapat palet *RGB*, karena nilai *RGB* langsung diuraikan dalam data *bitmap*. Setiap elemen data *bitmap* panjangnya 3 *byte*, masing-masing *byte* menyatakan komponen *R*, *G*, dan *B*.

Teknik Penggantian Bit pada Citra bukan 24-bit.

Sebelum melakukan penggantian bit *LSB*, semua data citra yang bukan tipe 24-bit diubah menjadi format 24-bit. Jadi, setiap data *pixel* sudah mengandung komponen *RGB*. Setiap *byte* di dalam data *bitmap* diganti satu bit *LSB*-nya dengan bit data yang akan disembunyikan. Jika *byte* tersebut merupakan komponen hijau (*G*), maka penggantian 1 bit *LSB*-nya hanya mengubah sedikit kadar warna hijau, dan perubahan ini tidak terdeteksi oleh mata manusia.

Teknik Penggantian Bit pada Citra 24-bit.

Karena data *bitmap* pada citra 24-bit sudah tersusun atas komponen *RGB*, maka tidak perlu dilakukan perubahan format. Setiap *byte* di dalam data *bitmap* diganti satu bit *LSB*-nya dengan bit data yang akan disembunyikan.

Perubahan Jumlah Warna

Pada citra 8-bit, jumlah warna terbatas, hanya 256 warna. Perubahan format citra 8-bit menjadi 24-bit akan menghasilkan warna baru (yang semula tidak terdapat di dalam palet *RGB*). Setiap elemen *RGB* pada tabel palet berpotensi menjadi 8 warna berbeda setekah proses penggantian bit *LSB*. Hal ini karena setiap data *bitmap* terdiri atas 3 *byte*, maka tersedia 3 bit *LSB* untuk penggantian. Penggantian 3 bit *LSB* menghasilkan $2^3 = 8$ kombinasi warna. Dengan demikian, steganografi pada citra 256 warna berpotensi menghasilkan $256 \times 8 = 2048$ warna.

Untuk menghindari kelebihan warna dari 256, maka sebelum proses penyembunyian data, warna citra 8-bit diturunkan terlebih dahulu menjadi 32 warna (jika jumlah warnanya kurang dari 32, tidak perlu dilakukan penurunan warna). Dengan demikian, jika setiap warna menghasilkan 8 warna baru, jumlah warna seluruhnya maksimum $32 \times 8 = 256$ warna.

Penurunan jumlah warna dilakukan dengan cara kuantisasi warna (*color quantization*). Penurunan jumlah warna harus tetap menghasilkan citra yang tampak persis seperti citra semula. Algoritma kuantisasi warna ada beberapa buah, antara lain algoritma *diversity*. Prinsip algoritma *diversity* adalah memaksimumkan perbedaan warna.

Algoritma *Diversity*:

1. Buat histogram citra. Warna yang frekuensi kemunculannya 0 dibuang karena tidak akan digunakan.
2. Pilih warna dengan frekuensi kemunculan tertinggi sebagai warna patokan. Masukkan warna ini ke dalam senarai warna terpilih.
3. Cari warna yang mempunyai perbedaan terjauh dengan warna patokan. Masukkan warna tersebut ke dalam senarai warna terpilih. Perbedaan dua buah warna dihitung dengan rumus jarak Euclidean:

$$d = \{ (r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2 \}^{1/2}$$

yang dalam hal ini, r_1 , g_1 , dan b_1 adalah komponen *RGB* dari warna pertama, dan r_2 , g_2 , dan b_2 adalah komponen *RGB* dari warna kedua.

4. Untuk setiap warna yang tersisa di dalam *list*, hitung jaraknya dari masing-masing warna di dalam senarai warna terpilih. Ambil warna yang paling jauh berbeda dengan warna yang sudah dipilih. Lakukan langkah 4 ini berulang kali sampai k warna sudah terpilih.

13.4 Ukuran Data Yang Disembunyikan

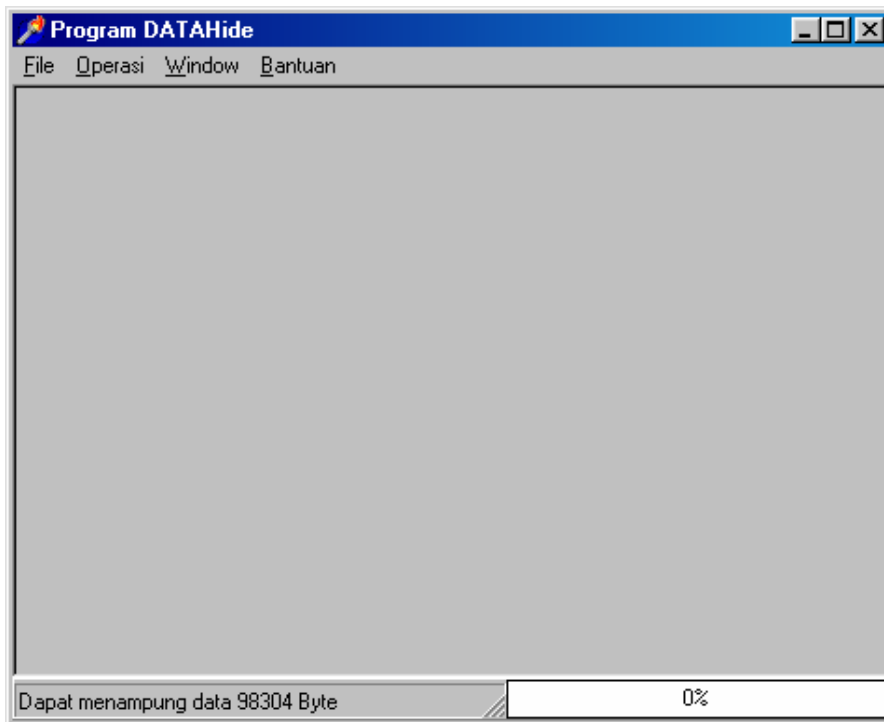
Ukuran data yang akan disembunyikan bergantung pada ukuran citra penampung. Pada citra 8-bit yang berukuran 256×256 *pixel* terdapat 65536 *pixel*, setiap *pixel* berukuran 1 *byte*. Setelah diubah menjadi citra 24-bit, ukuran data bitmap menjadi $65536 \times 3 = 196608$ *byte*. Karena setiap *byte* hanya bisa menyembunyikan satu bit di *LSB*-nya, maka ukuran data yang akan disembunyikan di dalam citra maksimum $196608/8 = 24576$ *byte*. Ukuran data ini harus dikurangi dengan panjang nama berkas, karena penyembunyian data rahasia tidak hanya menyembunyikan isi data tersebut, tetapi juga nama berkasnya.

Semakin besar data disembunyikan di dalam citra, semakin besar pula kemungkinan data tersebut rusak akibat manipulasi pada citra penampung.

13.5 Teknik Pengungkapan Data

Data yang disembunyikan di dalam citra dapat dibaca kembali dengan cara pengungkapan (*reveal* atau *extraction*). Posisi *byte* yang menyimpan bit data dapat diketahui dari bilangan acak yang dibangkitkan oleh *PRNG*. Karena algoritma kriptografi yang digunakan menggunakan kunci pada proses enkripsi, maka kunci yang sama digunakan untuk membangkitkan bilangan acak. Bilangan acak yang dihasilkan sama dengan bilangan acak yang dipakai pada waktu penyembunyian data. Dengan demikian, bit-bit data rahasia yang bertaburan di dalam citra dapat dikumpulkan kembali.

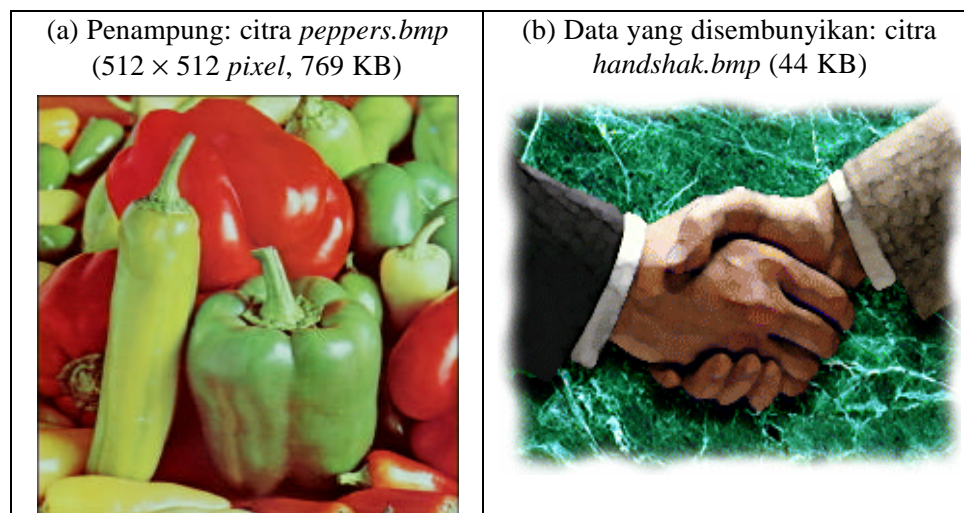
Di bawah ini ditampilkan contoh steganografi yang diambil dari program Tugas Akhir Lazarus Poli [POL98] yang diberi nama **DATAhide**. Untuk setiap contoh, digunakan kunci yang sama: *informatika*. Tampilan awal program diperlihatkan pada Gambar 13.2. Upa-menu yang ada pada menu Operasi adalah Penyembunyian data dan Pengungkapan data.

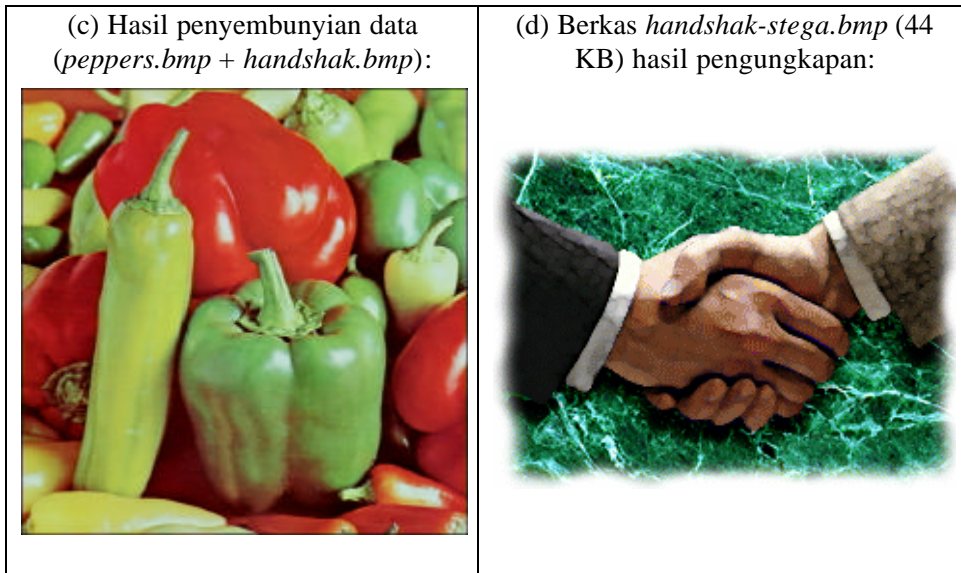


Gambar 13.2 Tampilan awal program **DATAhide** [POL98].

Contoh-contoh Hasil Steganografi

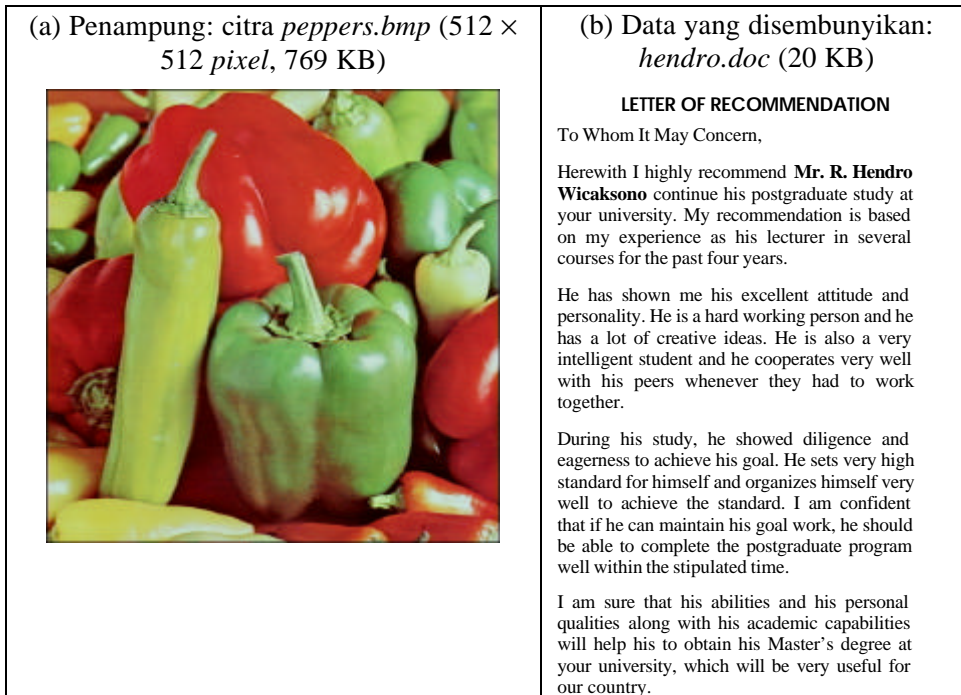
1. Citra penampung: citra 24 bit (berwarna)
Data yang disembunyikan: citra berwarna






Gambar 13.4 Penyembunyian citra handshaking ke dalam citra berwarna 24 bit (*peppers.bmp*)

2. Citra penampung: citra 24 bit (berwarna)
Data yang disembunyikan: teks



	<p>Bandung, November 15, 2002 Yours Sincerely,</p> <p>Ir. Rinaldi Munir, M.Sc. Senior Lecturer Informatics Engineering Department, Institute Technology of Bandung (ITB) Jl. Ganesha No. 10, Bandung 40132 Email : rinaldi@informatika.org Phone +62-22-2508135 Indonesia</p>
<p>(c) Hasil penyembunyian data (<i>peppers.bmp + hendro.doc</i>):</p> 	<p>(d) Hasil pengungkapan data (<i>hendro-stega.doc</i>, 20 KB):</p> <p>LETTER OF RECOMMENDATION</p> <p>To Whom It May Concern,</p> <p>Herewith I highly recommend Mr. R. Hendro Wicaksono continue his postgraduate study at your university. My recommendation is based on my experience as his lecturer in several courses for the past four years.</p> <p>He has shown me his excellent attitude and personality. He is a hard working person and he has a lot of creative ideas. He is also a very intelligent student and he cooperates very well with his peers whenever they had to work together.</p> <p>During his study, he showed diligence and eagerness to achieve his goal. He sets very high standard for himself and organizes himself very well to achieve the standard. I am confident that if he can maintain his goal work, he should be able to complete the postgraduate program well within the stipulated time.</p> <p>I am sure that his abilities and his personal qualities along with his academic capabilities will help his to obtain his Master's degree at your university, which will be very useful for our country.</p> <p>Bandung, November 15, 2002 Yours Sincerely,</p> <p>Ir. Rinaldi Munir, M.Sc. Senior Lecturer Informatics Engineering Department, Institute Technology of Bandung (ITB) Jl. Ganesha No. 10, Bandung 40132 Email : rinaldi@informatika.org Phone +62-22-2508135 Indonesia</p>

Gambar 13.5 Penyembunyian dokumen teks ke dalam citra berwarna 24 bit (*peppers.bmp*)

2. Citra penampung: citra 8 bit (*greyscale*)
 Data yang disembunyikan: audio

<p>(a) Penampung: citra <i>barbara.bmp</i> (512 × 512 pixel, 258 KB)</p> 	<p>(b) Data yang disembunyikan: <i>chord.wav</i> (95 KB), yaitu berkas musik dari Windows.</p> <p>(dimainkan dengan media player)</p>
<p>(c) Hasil penyembunyian data (<i>barbara.bmp</i> + <i>chord.wav</i>)</p>  <p>Pada kasus ini, terjadi penurunan kualitas gambar karena pengaruh penurunan jumlah warna (<i>color quantization</i>)!</p>	<p>(d) Berkas <i>chord-stega.wav</i> (95 KB) hasil pengungkapan:</p> <p>(dimainkan dengan media player)</p>

Gambar 13.6 Penyembunyian data audio ke dalam citra greyscale 8-bit

13.6 Watermarking

Salah satu karya intelektual yang dilindungi adalah barang dalam bentuk digital, seperti *software* dan produk multimedia seperti teks, musik (dalam format MP3 atau WAV), gambar/citra (*image*), dan video digital (VCD). Selama ini penggandaan atas produk digital tersebut dilakukan secara bebas dan leluasa. Hasil penggandaan persis sama dengan aslinya. Pemegang hak cipta atas produk digital tersebut tentu dirugikan karena ia tidak mendapat royalti dari usaha penggandaan tersebut.

Sebenarnya masalah penyalahgunaan hak cipta pada bidang multimedia tidak hanya mengenai penggandaan dan pendistribusiannya saja, tetapi juga mengenai label kepemilikan. Kebanyakan produk digital tersebut tidak mencantumkan siapa pemegang hak ciptanya. Kalaupun bukti kepemilikan itu ada, biasanya informasi kepemilikan disertakan pada sampul pembungkus yang menerangkan bahwa produk multimedia tersebut adalah milik pembuatnya. Masalahnya, distribusi produk multimedia saat ini tidak hanya secara *offline*, tetapi juga dapat dilakukan lewat internet. Jika anda masuk ke situs-situs *web* di internet, anda dapat menemukan informasi berupa teks, gambar, suara, dan video. Semua produk digital tersebut dapat anda *download* dengan mudah. Anda pun juga dapat mempertukarkan data digital dengan layanan internet seperti *e-mail*.

Masalahnya, hampir semua data digital yang bertebaran di dunia internet tidak mencantumkan informasi pemiliknya. Seseorang yang telah mendapatkan produk digital dapat mengklaim bahwa produk tersebut adalah hasil karyanya. Berhubung tidak ada bukti kepemilikan sebelumnya, maka klaim tersebut mungkin saja dipercaya.

Salah satu cara untuk melindungi hak cipta multimedia adalah dengan menyisipkan informasi ke dalam data multimedia tersebut dengan teknik *watermarking*. Informasi yang disisipkan ke dalam data multimedia disebut *watermark*, dan *watermark* dapat dianggap sebagai sidik digital (*digital signature*) dari pemilik yang sah atas produk multimedia tersebut. Dengan kata lain, *watermark* yang disisipkan menjadi label hak cipta dari pemiliknya. Pemberian *signature* dengan teknik *watermarking* ini dilakukan sedemikian sehingga informasi yang disisipkan tidak merusak data digital yang dilindungi. Sehingga, seseorang yang membuka produk multimedia yang sudah disisipi *watermark* tidak menyadari kalau di dalam data multimedia tersebut terkandung label kepemilikan pembuatnya.

Jika ada orang lain yang mengklaim bahwa produk multimedia yang didapatkannya adalah miliknya, maka pemegang hak cipta atas karya multimedia tersebut dapat membantahnya dengan mengekstraksi *watermark* dari dalam data multimedia yang disengketakan. *Watermark* yang diekstraksi dibandingkan dengan *watermark* pemegang hak cipta. Jika sama, berarti memang dialah pemegang hak cipta produk multimedia tersebut.

Pada dasarnya, teknik *watermarking* adalah proses menambahkan kode identifikasi secara permanen ke dalam data digital. Kode identifikasi tersebut dapat berupa teks, gambar, suara, atau video. Selain tidak merusak data digital produk yang akan dilindungi, kode yang disisipkan seharusnya memiliki ketahanan (*robustness*) dari berbagai pemrosesan lanjutan seperti perubahan, transformasi geometri, kompresi, enkripsi, dan sebagainya. Sifat *robustness* berarti data *watermark* tidak terhapus akibat pemrosesan lanjutan tersebut.

Sejarah Watermarking

Watermarking sudah ada sejak 700 tahun yang lalu. Pada akhir abad 13, pabrik kertas di Fabriano, Italia, membuat kertas yang diberi *watermark* atau tanda-air dengan cara menekan bentuk cetakan gambar atau tulisan pada kertas yang baru setengah jadi. Ketika kertas dikeringkan terbentuklah suatu kertas yang ber-*watermark*. Kertas ini biasanya digunakan oleh seniman atau sastrawan untuk menulis karya mereka. Kertas yang sudah dibubuhi tanda-air tersebut sekaligus dijadikan identifikasi bahwa karya seni di atasnya adalah milik mereka [HEN03].

Ide *watermarking* pada data digital (sehingga disebut *digital watermarking*) dikembangkan di Jepang tahun 1990 dan di Swiss tahun 1993. *Digital watermarking* semakin berkembang seiring dengan semakin meluasnya penggunaan internet, objek digital seperti video, citra, dan suara yang dapat dengan mudah digandakan dan disebarluaskan.

Perbedaan Steganografi dengan Watermarking

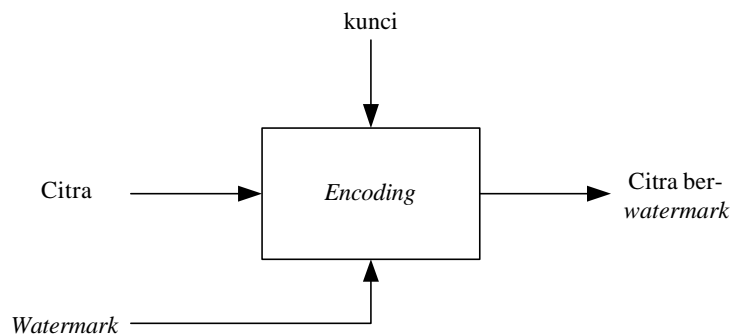
Watermarking merupakan aplikasi dari steganografi, namun ada perbedaan antara keduanya. Jika pada steganografi informasi rahasia disembunyikan di dalam media digital dimana media penampung tidak berarti apa-apa, maka pada *watermarking* justru media digital tersebut yang akan dilindungi kepemilikannya dengan pemberian label hak cipta (*watermark*).

Meskipun steganografi dan *watermarking* tidak sama, namun secara prinsip proses penyisipan informasi ke dalam data digital tidak jauh berbeda. Beberapa metode yang sudah ditemukan untuk penyisipan *watermark* adalah metode *LSB* (seperti pada penjelasan steganografi di atas), metode adaptif, metode *spread spectrum*, dan sebagainya.

Data *watermark* yang lazim disisipkan ke dalam data digital adalah teks, citra, atau suara. *Watermark* berupa teks mengandung kelemahan karena kesalahan satu bit akan menghasilkan hasil teks yang berbeda pada waktu verifikasi (ekstraksi). *Watermark* berupa suara atau citra lebih disukai karena kesalahan pada beberapa bit *watermark* tidak menghasilkan perubahan yang berarti pada waktu verifikasi. Hasil ekstraksi *watermark* yang mengandung kesalahan tersebut masih dapat dipersepsi secara visual (atau secara pendengaran jika *watermark*-nya berupa suara). Citra yang sering digunakan sebagai *watermark* biasanya logo atau lambang.

Penyisipan *Watermark*

Di sini kita hanya meninjau *watermarking* pada citra digital. Proses penyisipan *watermark* ke dalam citra disebut *encoding* dan ditunjukkan Gambar 13.7. *Encoding* dapat disertai dengan pemasukan kunci atau tidak memerlukan kunci. Kunci diperlukan agar *watermark* hanya dapat diekstraksi oleh pihak yang sah. Kunci juga dimaksudkan untuk mencegah *watermark* dihapus oleh pihak yang tidak berhak.



Gambar 13.7 Proses penyisipan *watermark* pada citra digital

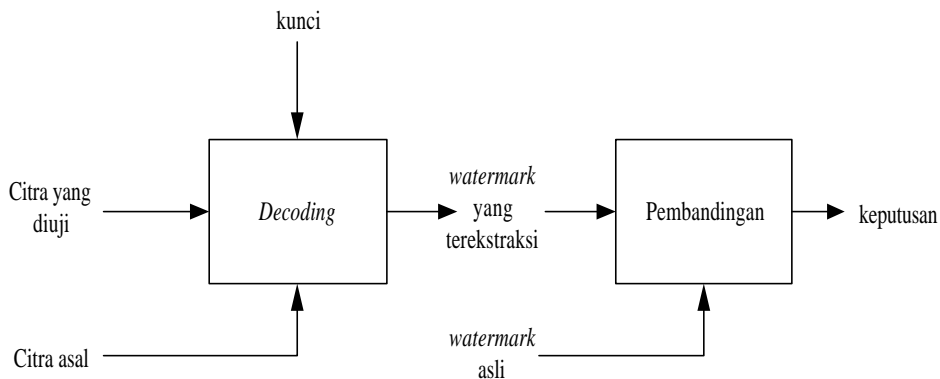
Gambar 13.8 memperlihatkan sebuah gambar (*image*) paprika yang disisipi dengan *watermark* berupa gambar hitam putih yang menyatakan identifikasi pemiliknya (Shanty) [HEN03]. Perhatikanlah bahwa setelah disisipi *watermark*, gambar paprika tetap kelihatan mulus, seolah-olah tidak pernah disisipi *watermark* sebelumnya. Sebenarnya tidaklah demikian, gambar paprika tersebut mengalami *sedikit* perubahan akibat *watermarking*, namun mata manusia mempunyai sifat kurang peka terhadap perubahan kecil ini, sehingga manusia sukar membedakan mana gambar yang asli dan mana gambar yang sudah disisipi *watermark*.



Gambar 13.8 Memberi watermark pada citra peppers

Verifikasi Watermark

Verifikasi *watermark* dilakukan untuk membuktikan status kepemilikan citra digital yang disengketakan. Verifikasi *watermark* terdiri atas dua sub-proses, yaitu ekstraksi *watermark* dan perbandingan. Sub-proses ekstraksi *watermark* disebut juga *decoding*, bertujuan mengungkap *watermark* dari dalam citra. *Decoding* dapat mengikutsertakan citra asal (yang belum diberi *watermark*) atau tidak sama sekali, karena beberapa skema *watermarking* memang menggunakan citra asal dalam proses *decoding* untuk meningkatkan unjuk kerja yang lebih baik [HEN03]. Sub-proses perbandingan bertujuan membandingkan *watermark* yang diungkap dengan *watermark* asli dan memberi keputusan tentang *watermark* tersebut. Proses verifikasi *watermark* ditunjukkan pada Gambar 13.9.



Gambar 13.9 Proses verifikasi watermark pada citra digital

Selain untuk tujuan pelabelan hak cipta (*copyright labelling*), *watermarking* juga dimanfaatkan untuk tujuan-tujuan lain sebagai berikut [SUP00]:

1. *Tamper-proofing*. *Watermarking* digunakan sebagai alat untuk mengidentifikasi atau menunjukkan bahwa data digital telah mengalami perubahan dari aslinya.
2. *Feature location*. *Watermarking* digunakan untuk mengidentifikasi isi dari data digital pada lokasi-lokasi tertentu.
3. *Annotation/caption*. *Watermarking* digunakan hanya sebagai keterangan tentang data digital itu sendiri.