

Optimasi Kompresi Berkas Dengan Memanfaatkan Teknik Steganografi LSB

Marhadiasha Kusumawardhana / 13508091

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganessa 10 Bandung 40132, Indonesia

marhadiasak@gmail.com

Abstract—Steganografi adalah teknik menyembunyikan suatu informasi tanpa orang lain kecuali pengirim dan penerima mengetahui file itu ada. Steganografi dengan teknik LSB adalah penyimpanan informasi dengan “menyisipkan” file yang ingin disembunyikan pada suatu gambar. Kompresi file adalah teknik-teknik untuk mengecilkan ukuran file dengan tidak menghilangkan informasi yang ada di dalamnya. Makalah ini akan membahas dan membuktikan bagaimana teknik steganografi LSB dapat dimanfaatkan dalam teknik kompresi file sehingga pengompresian file dapat dioptimasi, yaitu dengan lebih mengecilkan ukuran file yang telah dikompres.

Kata Kunci—Optimasi, Kompresi, Steganografi, LSB, Gambar, File

I. PENDAHULUAN DAN PAPARAN IDE

Pada revolusi industri ke-3 yang kita jalan saat ini, industri informasi, muncul satuan baru yang kini sudah populer dan mendarahdaging di masyarakat. Apabila pada revolusi uap muncul satuan populer “watt”, pada revolusi listrik muncul satuan populer “volt”. Pada revolusi informasi kini satuan baru yang populer adalah “byte”.

Satuan byte adalah ukuran standar jumlah informasi di sistem digital komputer. Tergantung konteksnya, satuan byte ini bisa berarti makin besar makin baik, atau makin kecil makin baik. Biasanya turunannya, byte per second, bersifat makin baik bila makin besar. Namun, pada kebanyakan kasus sifat satuan byte adalah makin besar makin buruk, atau dengan kata lain, makin kecil makin baik.

Sifat satuan byte yang akan dibahas pada makalah ini adalah sifat yang muncul pada mayoritas kasus yang kita temui bila bertemu dengan satuan ini. Yaitu, makin kecil ukuran dalam byte, maka makin baik/makin bagus. Lebih spesifik lagi, makalah ini membahas tentang teknik-teknik untuk mengoptimasi pengecilan ukuran byte pada suatu file pada sistem komputer. File adalah bagian/potongan dari sistem penyimpanan permanen (hard disk, misalnya) pada sistem komputer yang mempunyai nama dan ukuran tersendiri. Ukuran dari tiap file bergantung pada jumlah informasi yang tersimpan pada file tersebut (dalam byte).

Salah satu cara untuk memperkecil ukuran byte dari suatu file adalah dengan mengkompresi file. Tujuan dari

kompresi file ini adalah agar suatu file tidak memakan banyak ruang di perangkat penyimpanan atau bahkan untuk memperkecil jumlah informasi yang dikirimkan pada jaringan komputer sehingga pengiriman lebih cepat.

Apabila file dikompresi secara *lossless*, file yang menjadi kecil ukurannya tersebut tidak kehilangan sepeserpun informasi yang file tersebut kandung. Namun, apabila dalam keadaan terkompresi, file tersebut sulit dibaca karena byte yang tersimpan disusun dengan cara yang berbeda dari file aslinya. Oleh karena itu, kompresi file biasanya digunakan pada penyimpanan file yang sudah tidak digunakan lagi (arsip) atau pada pengiriman file lewat jaringan komputer.

Beberapa teknik kompresi sudah menjadi populer seperti DEFLATE dan LZ. Penulis akan membahas lebih lanjut tentang pengompresian file pada subab 2.

Steganografi merupakan salah satu teknik menyembunyikan pesan. Perbedaan steganografi dari teknik-teknik lainnya adalah, tidak hanya pesan tersebut tersembunyi, tapi sang “pengendus”, yaitu pihak lain selain pengirim dan penerima, tidak tahu bahwa pesan itu ada. Bisa dibilang steganografi menggunakan konsep “security through obscurity”, atau aman karena tidak terlihat/tidak jelas.

Salah satu teknik steganografi digital, atau steganografi pada sistem komputer adalah teknik steganografi LSB. Teknik steganografi LSB (Least significant byte) memanfaatkan sifat mata manusia yang tidak pandai dalam menyadari ada perbedaan kecil pada sebuah gambar. Oleh karena itu, pada perbedaan kecil yang tidak terlihat inilah kita bisa menyimpan informasi.

Dengan teknik LSB, informasi disisipkan (file, misalnya) pada sebuah gambar dengan mengganti bit (1 byte = 8 bit) yang paling tidak signifikan, yaitu bit yang tidak “menentukan”/”berkuasa” pada penentuan warna, pada gambar tersebut dengan bit-bit file yang ingin disisipkan pada gambar tersebut.

Dengan teknik ini, walaupun secara teknis dan eksak informasi atau nilai bit-bit dari gambar berubah, namun manusia hampir tidak mungkin bisa mendeteksi perbedaannya bila gambar tersebut dilihat pada layar karena bit yang diubah adalah bit yang paling tidak “berkuasa” pada penentuan warna gambar. Sehingga,

gambar terlihat persis dengan gambar sebelum disisipkan file. Padahal, gambar tersebut berbeda dengan gambar sebelumnya dan telah mengandung informasi tersembunyi didalamnya, yaitu file yang disisipkan tadi. Teknik LSB ini sangat penting dan mempunyai peran utama pada makalah ini.

Fitur dari teknik LSB, yaitu file dapat “disisipkan” tanpa mengubah gambar secara visual, bisa dimanfaatkan untuk kompresi file. Apabila sebuah file disisipkan pada suatu gambar, ukuran gambar tersebut tidak berubah karena bit-bitnya diganti (tidak ditambah) dengan bit file. Artinya, sebagian bit digambar (sekali lagi, bit yang tidak signifikan) hilang dan digantikan dengan bit-bit baru yang menyimpan informasi file.

Karena kita bisa mengekstrak file yang disisipkan tersebut dari gambar, maka bila file asli yang disisipkan tadi kita hapus dari sistem penyimpanan, namun gambar yang telah disisipkan file tetap ada, kita tidak kehilangan apapun. Hal ini disebabkan karena informasi file asli tadi sudah disisipkan di gambar, dan kita bisa mengekstraknya dari gambar untuk mendapatkannya kembali.

Apabila diperhatikan pada aspek ukuran file, sekarang kita punya dua informasi (isi file yang disisipkan dan gambar) dengan ukuran 1 file (ukuran gambar saja). Artinya jumlah byte untuk menyimpan informasi tersebut berkurang. Dengan teknik LSB ini, bisa dibayangkan kita telah melakukan kompresi dengan cara menggabungkan dua file dan menjadikan ukurannya menjadi satu file.

Bagaimana bila pada proses kompresi, sebelum file-file yang ingin dikompresi tersebut dikompres, file-file non-gambar (atau mungkin juga file gambar), disisipkan pada sebuah file non-gambar yang memang ingin dikompres juga? Jelas hasil ukuran file dari kompresi akan lebih kecil karena beberapa file telah disisipkan pada file gambar tersebut. Oleh karena itu, proses pengompresian file bisa kita optimasi bila file non-gambar disisipkan dengan file-gambar terlebih dahulu.

Namun, teknik ini hanya berguna bila format gambar tersebut adalah raster dan *lossless*, seperti PNG atau BMP[2] dan memang gambar tersebut memang ingin dikompresi bersama file lain oleh sang pengompresi file.

Ada satu batasan lagi, yaitu kompresi dengan penyisipan file pada gambar terlebih dahulu ini bersifat *lossy*, yaitu, secara teknis, ada informasi yang hilang pada proses pengompresian. Informasi yang hilang ini adalah bit-bit pada gambar yang diganti oleh bit-bit file yang disisipkan. Namun, seperti format audio MP3, hilangnya informasi pada file ini tidak begitu berpengaruh bila manusia yang mengonsumsi file tersebut, karena yang hilang tidaklah “signifikan” dalam menentukan bagaimana file tersebut ditangkap indra manusia.

II. TEORI DASAR STEGANOGRAFI DENGAN METODE LSB^[1]

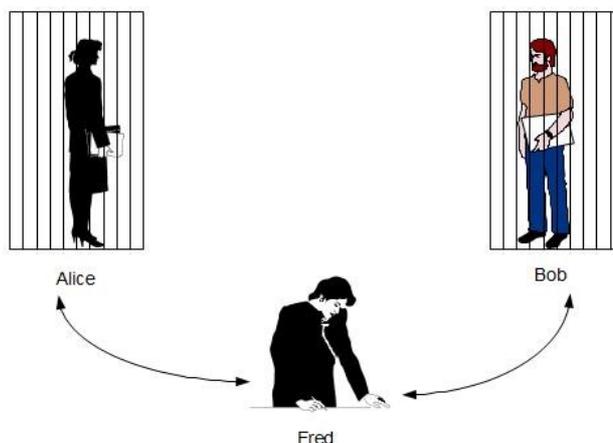
Pada subab teori dasar ini, penulis akan menjelaskan tentang dua topik yang akan disinergikan pada teknik yang dibahas makalah ini, yaitu konsep dasar

steganografi, terutama steganografi dengan teknik LSB dan konsep pengompresian file.

Konsep dan pengertian steganografi sebenarnya tidak terlalu dibutuhkan dan tidak memiliki peran besar dalam teknik ini. Teknik ini hanya memanfaatkan fitur dari penyisipan file pada LSB sebuah gambar, yang awalnya digunakan untuk tujuan steganografi. Sebenarnya, teknik ini sama sekali tidak dalam bentuk apapun bertujuan “menyembunyikan” informasi.

Sekali lagi, hanya teknik LSB, yang awalnya untuk steganografi, yang dimanfaatkan teknik ini. Oleh karena itu, pada subbab ini, tidak akan begitu banyak membahas lebih mendalam tentang steganografi. Tetapi yang penulis bahas lebih dalam adalah tentang metode penyisipan file pada gambar dengan teknik LSB.

Steganografi, seperti yang telah dijelaskan pada subbab sebelumnya, adalah salah satu cara untuk mengamankan suatu informasi. Kita ingat juga bahwa ada salah satu cara untuk mengamankan informasi lainnya, dengan mengenkripsi informasi. Enkripsi adalah metode menyembunyikan isi informasi dengan cara mengacak atau mengubah-ubah bit atau byte dari informasi tersebut sehingga tidak memiliki arti bagi yang membaca. Perbedaan enkripsi dengan steganografi adalah enkripsi membuat informasi tersebut tidak memiliki arti, namun tidak menyembunyikan keberadaannya. Sedangkan steganografi, informasi disembunyikan keberadaannya namun apabila dibaca, tetap memiliki arti.



Gambar 1: Pengirim, Penerima, dan Pengendus

Oleh karena itu, apabila pengendus melihat ada informasi yang telah dienkripsi, pengendus tetap akan curiga pada pengirim dan penerima walaupun tidak dapat mengetahui isi informasi tersebut, sedangkan bila pesan disembunyikan dengan steganografi, sang pengendus tidak akan curiga pada pengirim dan penerima, karena pengendus memang tidak tahu bahwa informasi yang disembunyikan itu ada.

Setelah mengerti konsep singkat dasar steganografi, mari kita tinjau teknik steganografi digital dengan metode LSB. LSB atau Least-significant bit adalah bit yang, apabila diterjemahkan secara harfiah, adalah bit yang

paling tidak signifikan. Arti signifikan di sini adalah bagaimana suatu bit bila pada suatu byte, bit itu berubah, misal dari 0 ke 1 atau 1 ke 0, mempengaruhi nilai keseluruhan byte tersebut.

LSB terletak pada bit terakhir, atau bit paling kanan pada sebuah byte. Sedangkan MSB, bit yang paling signifikan terletak di ujung kiri sebuah byte. Ya, seperti yang Anda duga, makin ke kiri, makin signifikan byte



Gambar 2: Letak LSB dibandingkan dengan Letak MSB

tersebut. Mari kita ubah bit pada LSB dan MSB, dan bandingkan perbedaannya. Pada byte 11001010, yang bernilai 202, bila kita ubah LSBnya, maka nilainya menjadi 11001011, atau 203 (perubahannya kecil). Sedangkan bila kita ubah MSBnya, menjadi 01001010, maka nilai berubah menjadi 74. Selisih nilai byte bila kita ubah LSBnya adalah 1, sedangkan MSB adalah 128.

Ketidaksinifikanan LSB inilah yang bisa dimanfaatkan untuk steganografi. Misal, kita memiliki sebuah gambar raster yang lossless, misal gambar dengan format BMP atau PNG^[2], dan gambar ini sama sekali tidak memunculkan kecurigaan pada siapapun. Lalu, kita memiliki pesan yang ingin dirahasiakan, misal tulisan “Saya belum mandi”.

Contoh: Pesan (teks) disembunyikan ke dalam gambar (citra)



Gambar 3: Contoh Gambar dan Pesan Rahasia

Idenya adalah dengan mengganti LSB byte-byte pada gambar dengan bit-bit pada tulisan “Saya belum mandi” tersebut. Karena yang bit yang diubah pada gambar adalah LSB, maka nilai byte-byte yang menentukan warna-warna pada gambar tersebut tidaklah banyak berubah. Apalagi, satu pixel biasanya direpresentasikan dengan 3-4 byte, mengubah 1 bit dari 4 byte, apalagi LSB, sangatlah tidak berpengaruh dalam tampilan visual gambar tersebut. Setelah dicoba dan dianalisis, ternyata dengan teknik LSB, apapun yang disisipkan pada LSB byte-byte di gambar, baik gambar putih polos atau gambar yang ramai, manusia sama sekali tidak dapat melihat perbedaan dari gambar sebelum disisipkan informasi dan gambar sesudah disisipkan informasi. Oleh karena itu teknik LSB ini

sangat ampuh untuk menyembunyikan informasi.

Yang kita manfaatkan dari teknik inilah adalah fakta bahwa bit dari gambar **diganti**, bukan **ditambah** dengan gambar tidak berubah secara signifikan. Artinya, besar ukuran file gambar tetap, namun kita memiliki dua file didalamnya, yaitu file gambar dan file yang disisipkan. File yang disisipkan itu bisa diekstrak dan dibangun kembali menjadi file utuh. Oleh karena itu, teknik ini bisa dimanfaatkan dalam teknik pengompresian.

III. TEORI DASAR PENGOMPRESIAN FILE^[3]

Setelah mengerti konsep singkat tentang steganografi, steganografi dengan teknik LSB, mari kita bahas konsep pengompresian file. Walaupun teknik yang akan dibahas makalah ini tidak mengubah atau menghilangkan sepeserpun langkah dari proses pengompresian file dan sebenarnya tanpa mengerti proses pengompresian file, kita sudah bisa menerapkan teknik ini, tidak ada salahnya kita sedikit menyentuh bidang ini.

Pengompresian file adalah proses mengubah data file tersebut sehingga untuk menampung informasi yang ada di dalam file tersebut, dibutuhkan jumlah bit yang lebih sedikit. Artinya dengan pengompresi file, ukuran file tersebut menjadi lebih kecil dari sebelumnya. Data yang dikompresi tidak musti satu file, namun bisa berbagai file dan dijadikan satu file arsip. File arsip ini akan berukuran lebih kecil dibanding kumulah ukuran seluruh file yang dikompres tadi.

Bagaimana file bisa dikompresi? Bukankah informasi akan hilang jika jumlah bit yang merepresentasikannya lebih kecil? Apakah ini sihir? Jelas bukan. Mari kita bahas sedikit tentang konsep dasar bagaimana file bisa dikompresi. Pada contoh ini, file dikompresi dengan metode LZ, atau Lempel-Ziv. LZ bersifat *lossless*, artinya tidak ada sedikitpun bit yang hilang bila file arsip diekstrak, dan bersifat dictionary coder.

Misalnya, kita memiliki kata “Satu untuk semua dan semua untuk satu”. Apabila disimpan dalam suatu file teks, jumlah byte yang digunakan adalah 37 byte (satu karakter = satu byte). Namun apabila kita perhatikan susunan kata dari kalimat tersebut, sebenarnya kalimat tersebut redundan. Artinya, ada lebih dari satu kata yang berulang. Bagaimana algoritma pengompresian file memanfaatkan hal ini dalam mengompresi file?

Dari sinilah nama “dictionary” muncul. Sang pengompresi membuat suatu “kamus” berisi daftar kata yang redundan. Misal, dari kalimat di atas terbentuk kamus pada tabel 1.

1	satu
2	untuk
3	semua

Tabel 1: Contoh Kamus

Pada tabel tersebut, dapat dilihat bahwa tiap kata yang redundan memiliki nomor pada kolom di sebelah kirinya. Sekarang mari kita ubah kalimat tersebut menjadi “1 2 3 dan 3 2 1”. Dengan nomor yang ada dikalimat itu adalah nomor kata yang direpresentasikan nomor tersebut pada tabel kamus. Kini kalimat memiliki ukuran 15 byte. Jelas lebih sedikit dengan jumlah kalimat awal, yakni 37 byte.

Kalimat yang berisi angka-angka referensi ke kamus ini akan dimasukkan dalam file arsip. Selain itu, tentunya file arsip harus mengandung kamus, tentunya sebagai tempat mencari kata apa yang direpresentasikan oleh nomor-nomor pada kalimat terkompres tersebut. Besar kamus pada contoh di atas adalah 16 byte (1 byte untuk tiap kata = 13 byte, ditambah angka 1, 2, dan 3). Isi dari file arsip adalah kamus ditambah dengan kalimat yang telah diubah menjadi kalimat dengan katanya direferensikan oleh nomor-nomor pada kamus. Maka besar file arsip yang kita dapatkan adalah $16 + 15 = 31$ byte. Jumlah byte ini lebih kecil dari jumlah awal, yakni, 37 byte.

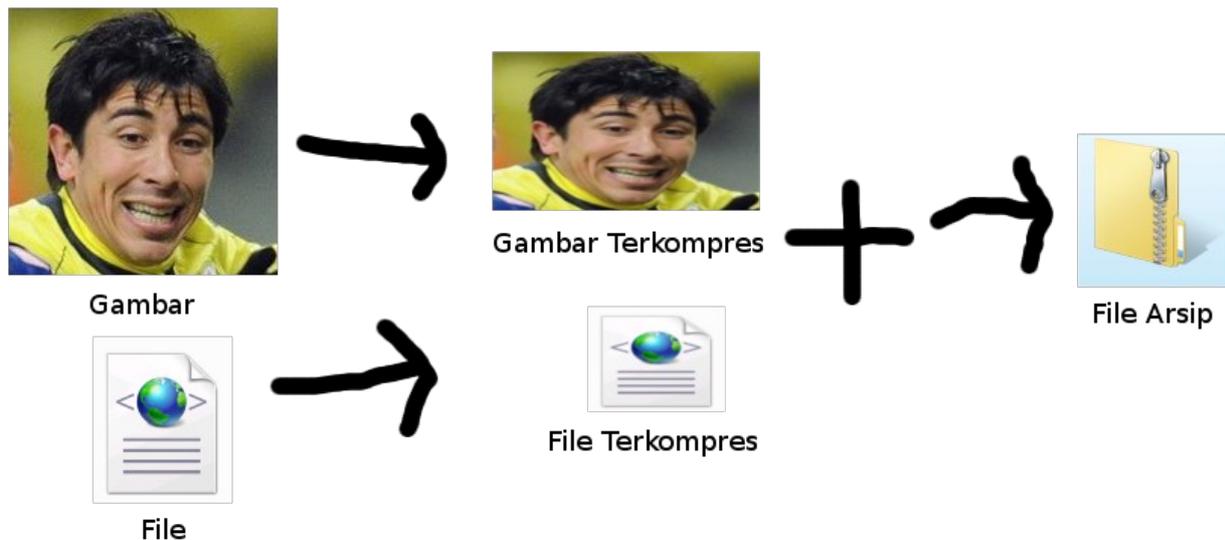
Rasio kompresi $31/37$ cukup besar, namun ini merupakan contoh kecil saja. Bayangkan yang dikompresi adalah file buku yang mengandung ribuan kata yang berulang. Jelas, ukuran file akan lebih kecil dan rasio kompresi akan kecil pula. Besar rasio kompresi juga

IV. TEKNIK MENGOPTIMASI PROSES PENGOMPRESIAN FILE DENGAN MEMANFAATKAN TEKNIK STEGANOGRAFI LSB.

Setelah mengerti konsep-konsep dasar dari steganografi LSB dan sedikit mengerti konsep dasar mengenai kompresi file, sekarang mari kita terjun ke bagian utama dari makalah ini.

Pada makalah ini, kita akan membuat proses pengompresian baru. Proses baru ini tidak mengubah atau mengedit sedikitpun algoritma kompresi yang telah ada. Yang berubah adalah ada langkah baru dalam keseluruhan proses pengompresian.

Seperti yang telah kita ketahui, kita bisa menyisipkan file pada file gambar dengan teknik LSB, tanpa menambah sepeserpun ukuran gambar dan dengan perubahan visual gambar yang dapat diabaikan. Kita ketahui juga bahwa tidak hanya satu file yang bisa dikompresi, tapi juga satu direktori, dan juga sejumlah file. Direktori dan sejumlah file bila dikompresi akan tetap menjadi 1 file arsip walaupun jumlah file yang dikompresi lebih dari satu.



Gambar 4: Langkah-langkan Pengompresian Dua File

berbeda-beda tergantung algoritma yang dipakai. Biasanya algoritma pengompresian dengan metode dictionary akan bervariasi dalam organisasi, pemilihan kata yang masuk kamus, dan lain sebagainya. Sebagai contoh, sebuah file XML sudoku.xml, yang memiliki ukuran 144 KB, apabila dikompresi dengan algoritma 7-zip, menjadi 5 KB. Sedangkan pada dikompresi dengan algoritma ZIP, menjadi 7 KB.

Selain teknik dictionary encoding, ada beberapa teknik lain. Beberapa teknik kompresi lain adalah entropy encoding, CTW, BWT, PPM, DMC, dan Delta [4].

Seperti yang Anda lihat di gambar 4, kita melakukan kompresi dua file. Tergantung dari algoritma pengompresiannya, bit-bit dan informasi dari file gambar dan file teks akan dikompresi dan digabungkan menjadi satu file arsip. Bisa dengan masing-masing dikompresi, lalu dikonkatenasi, atau seluruh file dianggap satu deret byte dan langsung dikompresi. Tentu dengan dua cara berbeda ini akan membuat cara dekompresinya pun berbeda. Namun, intinya, yang dikompresi adalah dua file, yaitu file gambar dan file teks.

Apabila dihitung ukuran hasil kompresinya maka yang

kita dapatkan adalah

$$[ukuran\ file\ arsip] = [kompresi\ file\ teks] + [kompresi\ gambar]$$

Sekali lagi, dari persamaan di atas, kita mendapatkan ukuran file arsip yang merupakan konkatenasi hasil kompresi file dengan kompresi gambar atau deret byte konkatenasi keduanya, lalu dikompresi sekaligus.

Secara logis, ukuran file arsip mana yang lebih kecil, apakah ukuran hasil kompresi dua file, yaitu file teks dan file gambar, atau hasil kompresi satu file, yaitu file gambar saja (dengan ukuran gambar sama dengan kasus sebelumnya)? Jelas semua orang akan menjawab bahwa apabila kita mengompres hanya gambar saja, tidak termasuk file teksnya, maka ukuran file arsip akan lebih kecil. Hal ini disebabkan panjang konkatenasi byte lebih kecil yang menyebabkan kamus pun lebih kecil, dan body file kompresi pun lebih pendek.

Ide yang penulis ingin paparkan di makalah ini adalah apabila seseorang ingin mengompresi lebih dari satu file, dan salah satunya ada file gambar (lossless dan raster), maka kita dapat mengecilkan ukuran file arsip dengan terlebih dahulu menyisipkan file-file yang ingin dikompresi ke salah satu file gambar tersebut. Lalu, kakas pengompres hanya mengompresi file gambar yang sudah disisipkan file-file lain tersebut.

Misal, kita lanjutkan contoh pengompresian dua file yang sudah kita temui sebelumnya. Kita mempunyai dua file, yaitu file gambar dan file teks. Keduanya ingin kita kompresi. Nah, dengan kita bisa mengoptimasi hasil kompresi file arsip dengan memanfaatkan teknik steganografi dengan menyisipkan file teks pada file gambar. Setelah itu, yang dikompresi hanya file gambar saja. Kita tidak akan kehilangan file teks, karena file teks

memang sudah disisipkan pada file gambar. Lihat Gambar 5.

Seperti yang Anda lihat, yang dikompresi hanya satu file, yaitu file gambar. Namun, file teks tetap ikut di dalamnya, namun “tersembunyi” di file gambar. Ukuran file arsip akan lebih kecil karena yang dikompresi hanya satu file, yaitu file gambar. Berikut perhitungan ukuran file arsip.

$$[ukuran\ file\ arsip] = [kompresi\ gambar]$$

Tentunya, pada saat kompresi file diekstraksi, kakas tersebut juga harus mengekstraksi file yang disisipkan tersebut dari gambar.

Pada prakteknya, seharusnya kakas ini harus bisa membuat organisasi penyisipan file pada gambar dengan seefektif mungkin. Misal yang dikompresi ada 1 gambar dan 3 file teks. 3 file teks ini cukup bila disisipkan pada gambar tersebut. Maka, alangkah baiknya bila seluruh isi file teks disisipkan pada gambar terlebih dahulu sebelum dikompresi.

Lebih efektif lagi, bila yang ingin dikompresi ada dua file gambar dan bila salah satu file gambar dari keduanya cukup besar sehingga sanggup menampung file gambar yang lain bila disisipkan. Karena file gambar juga bisa disisipkan ke file gambar.

Oleh karena itu, sebelum mengompresi, kakas harus menentukan terlebih dahulu file gambar mana yang ingin dijadikan penampung sisipan file lain, dan menentukan file mana yang ingin disisipkan. Karena bisa ada satu kasus kita memiliki 100 file teks dan 3 file gambar, dan 3 file gambar ini bisa menampung masing-masing 33 file teks. Kakas harus bisa mengorganisasikan dan menentukan peran masing-masing file pada teknik



Gambar 5: Mengompresi Dua File dengan Menyisipkan File Teks Terlebih Dahulu

optimasi ini. Yakni tiap ketiga gambar masing-masing menyisipkan ke-33 file teks tersebut dan meninggalkan 1 file teks. File teks ini tidak bisa disisipkan karena memang sudah tidak ada lagi file gambar yang menampungnya. Maka, yang dikompresi ada 4 file, yaitu ke 3 file gambar (yang menampung 99 file teks) dan 1 file teks. Jelas ukuran file arsip akan *jauh* lebih kecil dibandingkan bila dikompresi langsung dengan konkatenasi 103 file.

Tentunya, bila file arsip di atas diekstrak, yang kita dapat adalah 4 file saja, maka sang kakas dekompresii harus mengekstrak ke 99 file teks yang tersisip di gambar tersebut juga.

Berikut langkah-langkah proses kompresi kakas yang memanfaatkan teknik steganografi untuk kompresinya.

1. Pisahkan, Tentukan, dan organisasikan file mana yang ingin menampung file lain, file mana yang ingin disipkan, dan menentukan file mana menampung file apa. Algoritma organisasi inilah yang penting dan yang dikompresikan bila ada kompetisi kakas kompresi yang menggunakan teknik ini.
2. Sisipkan file-file yang ingin disisipkan dengan file gambar penampung.
3. Kompresi file penampung dan sisa file yang tidak dapat ditampung file gambar.
4. Jadilah satu file arsip dengan ukuran lebih kecil.

Berikut langkah-langkah proses dekompresi kakas yang memanfaatkan teknik steganografi untuk kompresinya.

1. Dekompresi file arsip
2. Cek apakah file-file gambar hasil ekstraksi mengandung file lain.
3. Ekstraksi file yang disisipkan di file gambar
4. Seluruh file-file sebelum dikompresi masih utuh, tanpa user mengetahui bahwa kakas sudah menyisipkan file lain pada file gambar.

Sekali lagi, pengompresian dengan metode ini bersifat *lossy*, artinya ada bit yang hilang di proses pengompresian ini. Bit yang hilang adalah bit-bit LSB di file gambar yang tergantikan oleh file yang disisipkan di file gambar tersebut. Namun, sekali lagi, *lossynya* gambar sangat tidak berpengaruh karena manusia tidak akan dapat melihat perbedaannya secara visual.

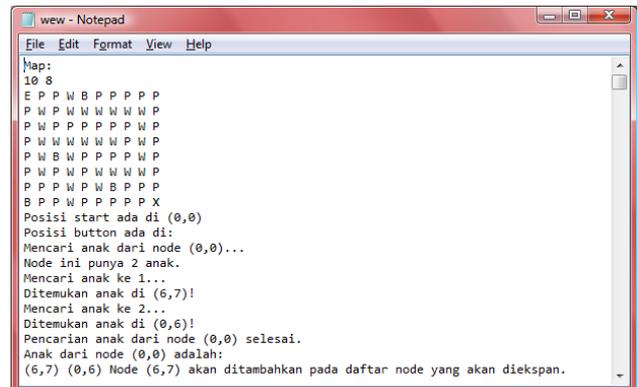
V. PENGUJIAN DAN ANALISIS HASIL UJI

Apakah dengan teknik yang sudah dijelaskan disubbab 4 tadi file arsip memang lebih kecil? Oleh karena itu, kita akan melakukan uji coba teknik ini.

Kita akan menguji 1 kasus sederhana yakni 1 file gambar dan 1 file teks.. Kita akan mengompresi dua file yaitu *beatles.bmp* yang berukuran 245 KB dan satu buah file teks *wew.txt* yang berukuran 16,8 KB.

The BEATLES

Gambar 7: File gambar



Gambar 8: File teks

Berikut perbandingan hasil ukuran pengompresian dengan file *wew.txt* disisipkan terlebih dahulu dan tanpa dengan file *wew.txt* disisipkan.

Ukuran file arsip tanpa penyisipan	3,18 KB
Ukuran file arsip dengan penyisipan	2,05 KB

Tabel 2: Perbandingan ukuran file arsip dengan kompresi teknik baru dan teknik lama.



Gambar 9: Perbedaan file arsip dengan penyisipan.

Dari hasil tersebut terbukti bahwa dengan penyisipan file non-gambar pada file gambar terlebih dahulu sebelum dikompresi akan menghasilkan ukuran lebih kecil dibandingkan pengompresian file konvensional.

Yang menjadi masalah berikutnya, adalah proses dekompresi. Kita harus mengembalikan file teks yang disisipkan secara utuh dari file arsip. Caranya, seperti yang telah dipaparkan sebelumnya, adalah dengan menambahkan satu langkah lagi di proses pendekompresian, yaitu mengekstrak file teks yang disisipkan tadi dari file gambar hasil ekstraksi file arsip.

VI. KESIMPULAN

Dari hasil uji coba dan analisis, dapat disimpulkan bahwa dengan memanfaatkan teknik steganografi LSB, kita dapat mengoptimasi proses pengompresian file, sehingga ukuran file arsip menjadi lebih kecil. Namun, perlu ditambahkan satu langkah lagi, yaitu mengekstraksi file yang disisipkan pada file gambar. Kakas yang menggunakan teknik ini juga harus bisa mengorganisasikan dan menentukan gambar sebagai penampung dan file mana yang disisipkan sedemikian sehingga pengompresian file efisien dan ukuran file arsip kecil.

VII. DAFTAR PUSTAKA

- [1]Slide kriptografi “Steganografi”
- [2]<http://club.myce.com/f34/png-image-format-lossless-107852/>
- [3]<http://www.howstuffworks.com/file-compression.htm>
- [4]<http://www.infosyssec.com/infosyssec/Steganography/techniques.htm>

VIII. PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 29 April 2010



Marhadiasha Kusumawardhana / 13508091