

# Steganografi pada Berkas yang dimampatkan menggunakan Huffman Coding

Ibnul Qoyyim<sup>1)</sup>

1) Jurusan Teknik Informatika ITB, Bandung, email: if14066@students.if.itb.ac.id

**Abstract** – Makalah ini membahas tentang steganografi pada berkas yang dimampatkan menggunakan Huffman coding.

**Kata Kunci:** steganografi, Huffman.

## 1. PENDAHULUAN

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui. Steganografi membutuhkan dua property yaitu media penampung dan pesan rahasia[3]. Media penampung dalam makalah ini adalah berkas yang dimampatkan menggunakan Huffman.

Huffman adalah cara memampatkan data dengan mengkodekan setiap karakter dalam data dengan kode yang lebih pendek. Untuk meminimumkan jumlah bit yang dibutuhkan, panjang kode untuk setiap karakter sedapat mungkin diperpendek, terutama untuk karakter yang kekerapannya (*frequency*) kemunculannya besar[1].

## 2. DASAR TEORI

Steganografi pada berkas yang dikompresi menggunakan Huffman dimungkinkan karena berkas Huffman memiliki suatu hal yang pasti ada pada header filenya yaitu Huffman tree atau Huffman table yang digunakan untuk menirmampatkan berkas tersebut menjadi berkas semula.

Informasi rahasia dapat dimasukkan kedalam berkas dengan cara transformasi header file Huffman ke header yang equal atau sedikit berbeda dengan header awal.

Transformasi pada tabel Huffman dapat dilakukan dengan mengubah urutan karakter pada tabel Huffman semula atau mengubah distribusi panjang kode pada tabel Huffman.

Sementara transformasi pada pohon Huffman dapat dilakukan dengan mengatur urutan karakter yang muncul pada daun di pohon Huffman jika ditelusuri secara preorder atau mengubah distribusi bobot keseimbangan pohon (misal jika terlalu banyak daun diletakkan pada satu sisi pohon akan mengakibatkan skew)

## 3. HASIL DAN PEMBAHASAN

Dari kedua macam header yang dapat digunakan sebagai media penampung (*cover-object*) pohon Huffman lebih baik daripada tabel Huffman.

Hal itu karena transformasi pada tabel Huffman lemah terhadap perubahan bentuk misal perubahan dari tabel Huffman ke pohon Huffman kemudian dikembalikan lagi menjadi tabel Huffman dapat menjadi tabel Huffman yang berbeda urutan karakter atau distribusi panjang kode-nya, hal itu karena tidak ada aturan tentang pengurutan pada tabel Huffman sementara pada pohon Huffman sudah memiliki aturan tentang urutan penelusuran pada binary tree (pre order, in order, atau post order).

Kemudian memilih metode yang cocok untuk menyisipkan informasi rahasia pada pohon Huffman dipilih dengan pengaturan urutan karakter yang muncul pada daun di pohon Huffman diusahakan tanpa merubah panjang kodenya. Hal itu disebabkan keputusan apakah suatu simpul diletakkan di kiri atau kanan pohon Huffman tidak mempengaruhi panjang kode namun mempengaruhi kode yang dihasilkan[1].

Cara memasukkan informasi yang akan disembunyikan pada pohon Huffman adalah sebagai berikut :

1. Mula-mula tetapkan suatu byte pembanding byte tersebut akan digunakan untuk membandingkan byte-byte pada pohon Huffman.
2. Rumuskan informasi perbandingan dengan byte pembanding untuk menghasilkan bit yang disembunyikan. (misal jika lebih dari byte pembanding maka menghasilkan bit 1 dan sebaliknya jika kurang dari byte pembanding maka menghasilkan bit 0)
3. Buat pohon Huffman seperti pada [1] namun dengan tambahan byte-byte dipilih agar bertetangga sedemikian rupa sehingga sesuai dengan informasi yang disembunyikan. (didalam program untuk mempermudah diimplementasikan dengan membuat pohon skew, seharusnya pembuatan pohon Huffman sealami mungkin agar efektifitas kompresi tidak berkurang)

Sementara cara mengembalikan informasi tersebunyi dari pohon Huffman adalah dengan cara :

1. Dapatkan byte pembanding (dalam program byte pembanding merupakan konstanta namun seharusnya ditentukan berdasarkan statistic bit 0 dan 1 pada informasi yang disembunyikan (*hiddentext / embedded message*))
2. Telusuri pohon Huffman sesuai cara yang ditentukan (pre order, in order, atau post order)
3. Dari hasil penelusuran dapatkan bit-bit *hiddentext* dengan cara membandingkan dengan bit pembanding.

Dari hasil perhitungan maka pohon atau tabel Huffman yang lengkap akan memiliki seluruh karakter akan memiliki 256 daun pada pohon Huffman atau 256 baris pada tabel Huffman.

Hal itu berarti informasi yang dapat disembunyikan pada pohon Huffman maksimal adalah 256 bit  $\approx$  32 byte dengan asumsi menggunakan metode diatas sehingga setiap daun dapat menyimpan 1 bit informasi.

#### 4. KESIMPULAN

Dari pembahasan diatas dapat disimpulkan bahwa steganografi pada berkas yang dimampatkan menggunakan Huffman coding dimungkinkan namun tidak semua kemungkinan atau macam bit pesan dapat

disimpan hal itu karena media penampung merupakan hasil perhitungan dari berkas asal dan memiliki constrain tertentu yang jika dilanggar akan mengurangi keefektifan pemampatan Huffman tersebut.

Beberapa saran untuk pengembangan program dimasa depan adalah :

1. Penyisipan tanpa mengurangi efektifitas kompresi dengan cara membuat pohon Huffman secara alami namun untuk mengimplementasikan hal tersebut dibutuhkan algoritma yang rumit.
2. Pemilihan byte pembanding yang dinamis.
3. Mode penyisipan yang berbeda.

#### DAFTAR REFERENSI

- [1] Munir, Rinaldi., Diktat Kuliah IF2151 Matematika Diskrit, 2004.
- [2] Munir, Rinaldi., Diktat Kuliah IF2251 Strategi Algoritmik, 2004.
- [3] Munir, Rinaldi., Diktat Kuliah IF5054 Kriptografi, 2004.
- [4] Mostava, Hatem. *Simple and Fast Huffman Coding*, [http://codeproject.com/cpp/huffman\\_coding.asp](http://codeproject.com/cpp/huffman_coding.asp), 2005