

Adaptive Huffman Coding Sebagai Variasi Huffman Coding

Muhammad Ardhin (13509033)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509033@std.stei.itb.ac.id

Abstrak - Makalah ini berisikan tentang pembahasan salah satu variasi dari Kode Huffman, yaitu Adaptive Huffman Coding dan aplikasinya dalam kompresi data. Adaptive Coding sendiri merupakan suatu metode dalam kompresi data yang merupakan variasi dari Entropy Encoding. Adaptive Huffman Coding adalah Adaptive Coding yang berlandaskan Huffman Coding. Adapun Huffman Coding, adalah salah satu aplikasi terkemuka dari Binary Tree

Kata Kunci – Kompresi, Adaptive, Huffman, Entropy Encoding

I. PENDAHULUAN

Sebagai pendahuluan, akan terlebih dahulu dipaparkan berbagai definisi dari istilah-istilah yang banyak digunakan di makalah ini

A) Data Compression

Data Compression adalah proses meng-enkodekan informasi sehingga menggunakan sejumlah lebih sedikit bit dari pada data yang belum ter-enkode. Proses meng-enkodekan data sendiri adalah sesuatu yang banyak dipelajari dan merupakan cabang tersendiri dalam ilmu computer yang telah lama diteliti.

Data Compression sendiri terbagi menjadi 2 kategori utama, yaitu **Lossy Compression** dan **Lossless Compression**. Lossy Compression adalah pengompresan data yang menyebabkan banyak data utama yang hilang (lossy). Lossy Compression kurang secara kualitas, namun efektif dalam mendapatkan ruang lebih. Lossless Compression adalah kompresi data yang tidak membuang data utama (lossless), tapi tetap menghasilkan data hasil kompresi. Meski tidak seefektif Lossy dalam hal mendapatkan ruang, tapi unggul dalam kualitas data.

B) Lossless Compression

Seperti dijelaskan di atas, Lossless Compression memungkinkan data dikompres tanpa menghilangkan data-data utama. Secara formal, Lossless Compression adalah sebuah kelas dari kumpulan algoritma Data Compression yang memungkinkan data original untuk direkonstruksi dari data terkompres. Kebanyakan program Lossless Compression melakukan 2 hal secara

sekuensial : Langkah pertama membuat model statistical untuk data input, dan kedua menggunakan model ini untuk memetakan input data ke rentetan bit sehingga data termanipulasi dan dihasilkan data yang lebih pendek

C) Entropy Encoding

Entropy Encoding adalah sebuah skema dari Lossless Compression yang independen terhadap karakteristik medium. Selain digunakan untuk meng-kompres data digital, Entropy Encoding juga dapat digunakan untuk menentukan kesamaan antara data-data dalam stream. 2 teknik Entropy Encoding yang terkenal adalah Huffman Coding dan Arithmetic Coding.

D) Adaptive Coding

Adaptive Coding adalah variasi dari Entropy Encoding. Adaptive Coding adalah cocok untuk stream data, karena dinamik dan beradaptasi terhadap perubahan di karakteristik data. Adaptive Coding membutuhkan encoder dan decoder yang lebih rumit untuk mempertahankan keadaan tetap tersinkronisasi, dan juga daya komputasi yang lebih.

Adaptive Coding memanfaatkan model yang dimiliki oleh sebuah metode Data Compression, yang merupakan prediksi dari komposisi data. Enkoder mentransmisikan isi data dengan memanfaatkan referensi terhadap model.

Pada Adaptive Coding, Enkoder dan Dekoder memiliki meta model terdefinisi, yang berguna untuk merenspons kepada isi actual dari data. Saat data ditransmisikan, encoder dan decoder mengadaptasikan modelnya masing-masing.

Enkoder terlebih dahulu menginisialisasi data model, dan bila masih ada data yang harus dikirim, meng-enkodekan symbol selanjutnya menggunakan data model dan mengirimkannya. Seterusnya memodifikasi data model berdasarkan symbol terakhir

Dekoder terlebih dahulu menginisialisasi data model seperti encoder, kemudian bila masih ada data yang harus diterima, meng-dekodekan symbol selanjutnya menggunakan data model dan mengeluarkannya (sebagai output), kemudian memodifikasi data model berdasarkan symbol yang telah terdecode.

Perbedaan dengan metode static adalah, pada metode static, encoder dan decoder tidak perlu mengubah data model, karena memang tidak memiliki data model

E) Kode Huffman

Huffman Coding adalah salah satu teknik kompresi data yang sangat populer. Huffman Coding adalah algoritma Entropy Encoding yang digunakan untuk Lossless Data Compression. Teknik yang ditemukan oleh David A Huffman ini menggunakan binary tree sebagai dasarnya. Huffman Coding merupakan dasar banyak metode kompresi data, diantaranya digunakan dalam mengcompress file-file MPEG dan file audio lainnya, berbagai pengolahan citra, dan juga merupakan salah satu tiang utama algoritma untuk aplikasi pembentukan file Zip yang terkenal. Huffman Coding banyak digunakan karena dengan teknik ini data yang dikompresi tidak akan kehilangan kualitasnya (Lossless Compression).

Huffman Coding banyak diaplikasikan dalam mengcompress sejumlah bilangan biner sehingga mempunyai representasi yang lebih pendek

Langkah-langkah membuat Huffman Tree untuk serangkaian karakter adalah sebagai berikut :

- 1) Urutkan karakter berdasarkan frekuensi kemunculan dari kecil ke besar, dan buat tabelnya
- 2) 2 karakter teratas (terkecil) dibuat menjadi daun dari pohon. Daun dengan frekuensi lebih kecil diletakkan di kiri
- 3) Kedua daun mempunyai simpul yang berfrekuensi jumlah kedua daun tersebut. Beri nama daun. Misal A
- 4) Hapus kedua buah data terkecil dari table
- 5) Masukkan A ke dalam table frekuensi, sesuai urutan frekuensinya
- 6) Ulangi langkah awal, yaitu kembali memilih 2 frekuensi terkecil di table, sampai table kosong
- 7) Pemberian nilai 0 untuk pohon kiri dan 1 untuk pohon kanan
- 8) Penerjemahan pohon dengan representasi baru karakter

Misalkan kata ABBCCDAA

- 1) Urutkan karakter :

D, C, C, B, B, A, A, A

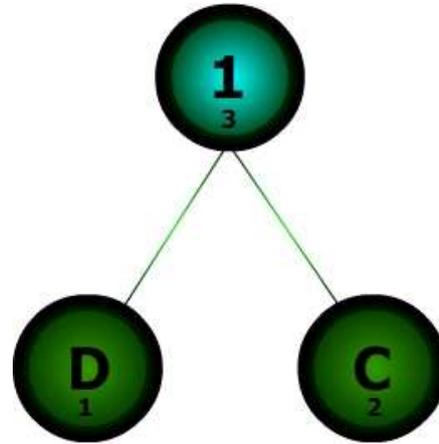
Tabel :

D	C	B	A
1	2	2	3

- 2) Pembuatan daun



- 3) Pembuatan Simpul 1



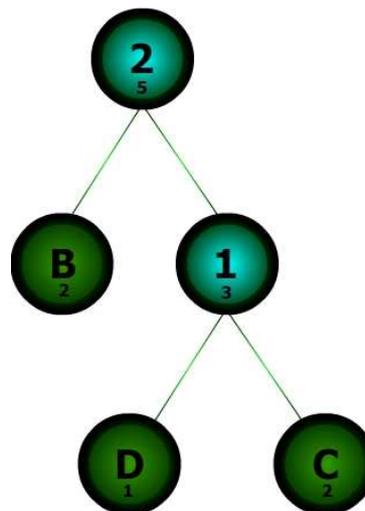
- 4) Penghapusan data table

		B	A
		2	3

- 5) Pemasukan data baru dalam table

	B	1	A
	2	3	3

- 6) Penyelesaian



		A	2
		3	5

Kompresi dengan Kode Huffman menghasilkan jumlah bit yang lebih baik dari pada kode ASCII yang berisikan 8 bit per karakter

II. ADAPTIVE HUFFMAN CODING

Adaptive Huffman Coding adalah sebuah teknik Adaptive Coding berdasarkan Huffman Coding. Teknik yang juga disebut Dynamic Huffman Coding ini ada untuk menutupi salah satu kekurangan Huffman Coding, yaitu fakta bahwa “uncompresser” perlu punya “pengetahuan” mengenai probabilitas dari symbol di file yang terkompres. Hal ini menambah bits yang diperlukan untuk meng-enkodekan file, dan apabila “pengetahuan” tidak tersedia, mengkompres file membutuhkan 2 langkah :

- 1) Mencari frekuensi,
- 2) Mengkompres file

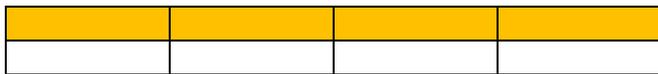
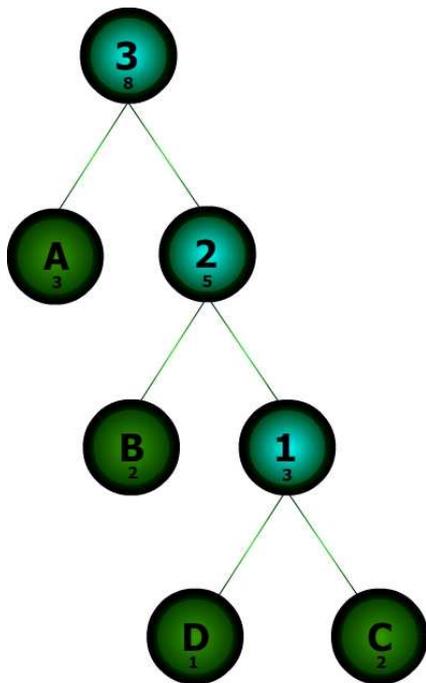
Adaptive Huffman Coding pertama kali dikembangkan oleh Faller dan Gallager, kemudian mendapatkan perbaikan oleh Donald Knuth. Sebuah metode lain dikembangkan oleh J.S. Vitter.

File yang ter-enkodekan secara dinamis punya dampak besar bagi keefektifan pohon sebagai encoder dan decoder. Adaptive Huffman Coding lebih efektif dari Huffman Coding karena pohon terus berevolusi. Pada Huffman Coding yang static, karakter yang berfrekuensi rendah akan ditempatkan jauh di posisi bawah, dan akan memakan banyak bits untuk encode. Pada Adaptive Huffman Coding, karakter itu akan dimasukkan pada daun tertinggi untuk didekodekan, sebelum akhirnya didorong ke bawah pohon oleh karakter berfrekuensi yang lebih besar.

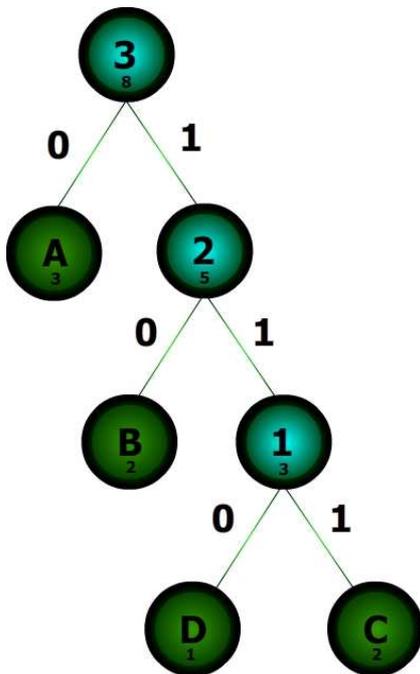
Kuncinya adalah property “sibling” yang mengatakan : Setiap simpul (kecuali akar) mempunyai sibling (saudara) dan setiap simpul dapat diurutkan dengan urutan “tidak bertambah berat”, dengan setiap simpul berdampingan dengan saudaranya”

Algoritma FKG (Faller-Gallager-Knuth) adalah algoritma standar untuk algoritma Adaptive Huffman . Pada FKG, pohon Huffman secara dinamik dikomputasi ulang saat dibutuhkan untuk mempertahankan efisiensi kode pohon. Pada setiap keadaan di proses dekompresisasi, decoder hanya dapat menerima kode yang telah terpetakan dengan symbol dari sumber, dan ada 2 set symbol dalam kasus ini :

- (1) Set yang “telah terlihat”
- (2) Set yang “belum terlihat”



7) Pemberian 0 dan 1



- 8) A : 0
B : 10
C : 111
D : 110

ABBCCDAA = 0101011111111000

Pseudocode untuk Algoritma FKG (Tree Update Method)

```

Update_TreeFGK ( input Simbol )
{
    p = Pointer ke simbol;

    if ( p == NULL )
    {
        /* Sebuah simpul baru */

        Buat 2 daun dari simpul 0, sehingga
        daun anak kanan adalah simbol yang
        baru, dan yang kiri adalah simpul 0
        yang baru;

        p = orang tua dari simpul simbol yang
        baru;
    }

    /* Jika kedua simpul bersaudara */

    if ( orang tua p == 0-simpul orang tua )
    {
        Tukar p di pohon dengan daun urutan
        tertinggi dengan berat yang sama;

        Nilai p ← Nilai p + 1;

        p ← orang tua p;
    }

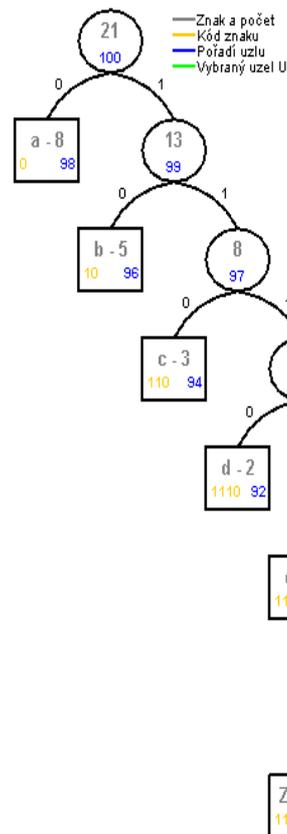
    while ( p != akar pohon )
    {
        Tukar p di pohon dengan simpul
        bernomor tertinggi dengan berat yang
        sama;

        Nilai p ← Nilai p + 1;

        p ← orang tua p;
    }
}

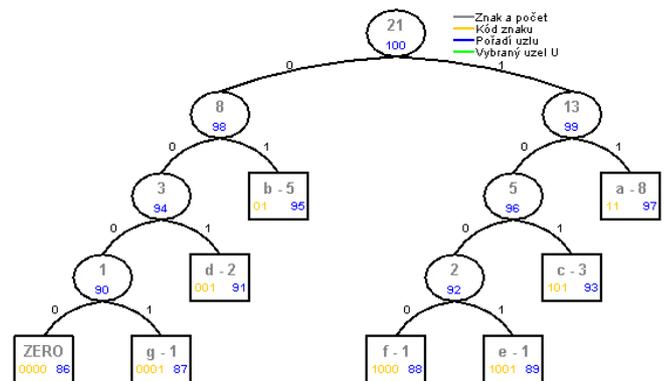
```

Diterjemahkan dari pseudocode berbahasa inggris dari http://www.stringology.org/DataCompression/ahv/index_en.html.



Contoh pohon hasil algoritma FKG : http://www.stringology.org/DataCompression/ahv/index_en.html

Prinsip dasar Algoritma Vitter sama dengan FKG. Pengirim (Enkoder) dan penerima (Dekoder) mempertahankan kesamaan secara dinamis dari pohon Huffman yang bervariasi, yang sudah berisi semua karakter terkode sebagai daun. Simpul daun dari pohon ini adalah simpul dengan jumlah simpul di masing-masing subpohon. Juga terdapat sebuah simpul special yaitu simpul ZERO, yang digunakan sebagai “escape sequence”. Perbedaannya adalah pohon yang dibuat oleh algoritma Vitter lebih seimbang.



Contoh pohon hasil algoritma FKG : http://www.stringology.org/DataCompression/ahv/index_en.html

III. KESIMPULAN

Masih ada banyak lagi Algoritma untuk mengimplementasikan Dynamic Huffman Coding, atau Adaptive Huffman Coding. Dengan menggunakan Adaptive Huffman Coding ini, telah banyak teknologi untuk memproses data yang telah dikembangkan, diantaranya teknologi memproses data-data multimedia, yang sering berubah-ubah seperti audio dan file streaming.

Makalah ini dibuat dengan cara mempelajari, menterjemahkan, dan menggabungkan ide-ide yang terdapat pada beberapa referensi

REFERENSI

- [1] Munir, Rinaldi, "Matematika Diskrit", Penerbit INFORMATIKA, Bandung, 2005
- [2] Setiadi, Robert, "Algoritma Itu Mudah", Penerbit PT Prima Infosarana Media, Jakarta, 2008
- [3] Hendrawan, HUFFMAN CODING, hend@telecom.ee.itb.ac.id
- [4] Eric Kurniawan, S.Kom, M.Kom, Kompresi Data
- [5] Adaptive Huffman Coding, Gabriele Monfardini - Corso di Basi di Dati Multimediali a.a. 2005-2006
- [6] http://en.wikipedia.org/wiki/Huffman_coding. Tanggal akses : 15-16 Desember 2010
- [7] http://en.wikipedia.org/wiki/Data_compression. Tanggal akses : 15-16 Desember 2010
- [8] http://en.wikipedia.org/wiki/Lossless_data_compression. Tanggal akses : 15-16 Desember 2010
- [9] http://en.wikipedia.org/wiki/Entropy_encoding. Tanggal akses : 15-16 Desember 2010
- [10] http://en.wikipedia.org/wiki/Adaptive_coding. Tanggal akses : 15-16 Desember 2010
- [11] http://en.wikipedia.org/wiki/Adaptive_Huffman_coding. Tanggal akses : 15-16 Desember 2010
- [12] <http://sites.google.com/site/datacompressionguide/fgk>. Tanggal akses : 16 Desember 2010
- [13] http://www.stringology.org/DataCompression/ahv/index_en.html. Tanggal akses : 16 Desember 2010
- [14] <http://www.cs.duke.edu/csed/curious/compression/adaptivehuff.html>. Tanggal akses : 16 Desember 2010

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

ttd

Muhammad Ardhin (13509033)