

Teknik Pembangkitan Kode Huffman

Muhammad Riza Putra

Program Studi Teknik Informatika ITB, Bandung 40132, email: zha@students.itb.ac.id

Abstrak – Makalah ini membahas suatu teknik dalam pembangkitan kode Huffman untuk karakter-karakter pada suatu string. Teknik pembangkitan kode Huffman ini berdasarkan pada algoritma yang dikembangkan oleh David A. Huffman, namun lebih mudah untuk dipahami oleh pelajar yang baru mengenal dan mempelajari algoritma pemampatan data ini. Makalah ini akan menjelaskan langkah per langkah dalam membangkitkan kode Huffman dengan menggunakan pohon biner dan tabel kekerapan kemunculan untuk setiap simbol yang akan dikodekan.

Kata Kunci: Kode, Huffman, kompresi, pemampatan, kekerapan, frekuensi, ASCII, pohon, biner, teknik, larik

1. PENDAHULUAN

Dalam bidang ilmu komputer dan teori informatika, kode Huffman merupakan algoritma pengkodean yang paling efisien dan *lossless* dalam melakukan pemampatan data. Algoritma ini dikembangkan dan diperkenalkan oleh salah satu doktor MIT, David A. Huffman melalui sebuah makalah yang diterbitkan tahun 1952 berjudul “A Method for the Construction of Minimum-Redundancy Codes” [1].

Peng-kodean Huffman ini didasarkan pada kekerapan kemunculan suatu simbol pada data sumber yang akan dimampatkan. Hasil dari peng-kodean ini akan menghasilkan kode awalan (*prefix code*) untuk setiap simbol dimana simbol yang kekerapan kemunculannya lebih besar memiliki kode awalan yang lebih pendek dari simbol yang memiliki kekerapan kemunculannya lebih kecil pada data sumber.

Sekarang sudah banyak bermunculan variasi peng-kodean Huffman, baik itu penggunaannya masih seperti algoritma asli Huffman, sampai pada algoritma lain yang melakukan optimasi terhadap kode awalan. Berikut beberapa varian dari Kode Huffman:

- Kode Huffman *n-ary*
Algoritma ini menggunakan alfabet $\{0, 1, \dots, n-1\}$ untuk meng-kodekan pesan dan membentuk pohon *n-er*. Pendekatan ini sudah terdapat pada paper asli Huffman
- Kode Huffman *adaptive*
Varian ini meng-kalkulasikan kekerapan kemunculan secara dinamis berdasarkan kekerapan kemunculan pada saat itu dalam string sumber

- Algoritma *Template Huffman*
Kebanyakan, atribut digunakan dalam pengimplementasian algoritma Huffman untuk menyatakan peluang numerik, namun algoritma ini tidak membutuhkannya; varian ini hanya membutuhkan suatu cara untuk mengurutkan atributnya dan menambahkannya pada pembangkitan kode Huffman
- Kode Huffman *length-limited*
Adalah suatu varian lainnya dimana tujuannya masih untuk mencapai panjang lintasan minimum, namun terdapat tambahan persyaratan dimana panjang untuk setiap kode harus lebih kecil dari konstanta yang diberikan
- Kode Huffman dengan *unequal letter costs*
Dalam permasalahan peng-kodean Huffman yang standar, diasumsikan bahwa setiap simbol dimana kode dibangkitkan, memiliki biaya yang sama untuk mentransmisikannya, dengan kata lain meminimalisasi total biaya dari pesan dan meminimalisasi jumlah total digit adalah sesuatu yang sama. Dengan metode ini, asumsi diatas tidak lagi benar, kode yang dihasilkan tidak harus memiliki panjang yang sama, tergantung pada karakteristik media transmisinya
- Optimasi Alfabet Pohon Biner
Metode peng-kodean Huffman lainnya untuk menangani permasalahan Hu-Tucker [1].

2. ALGORITMA KODE HUFFMAN

Berikut penjelasan singkat mengenai algoritma peng-kodean Huffman.

2.1. Definisi Masalah

Deskripsi Informal

Diberikan:

Diberikan himpunan simbol/karakter dan jumlah kemunculannya pada data sumber.

Temukan:

Himpunan kode awalan setiap simbol dengan meminimumkan panjang kode.

Deskripsi Formal

Masukan:

$S = \{s_1, s_2, s_3, \dots, s_n\}$ adalah himpunan simbol dari data sumber n

$F = \{f_1, f_2, f_3, \dots, f_n\}$ adalah jumlah kemunculan dimana $f_i = \text{frek}(s_i)$, $1 = i = n$

Keluaran:

Kode $(S, F) = \{c_1, c_2, c_3, \dots, c_n\}$ dimana c_i adalah kode awalan untuk s_i

Tujuan:

Misalkan $L(C)$ adalah panjang untuk kode c . Maka $L(C) = L(T)$ untuk setiap kode $T(S, F)$

2.2. Algoritma Dasar

Untuk mendapatkan kode Huffman, teknik dasar adalah dengan membangkitkan pohon biner dari simbol-simbol/karakter yang akan dicari kodenya. Hal ini dapat disimpan ke suatu larik, dengan ukurannya tergantung pada jumlah simbol, n . Sebelumnya, dihitung terlebih dahulu kekerapan kemunculan tiap simbol di dalam data sumber. Berikut langkah-langkah dalam pembangkitan pohon biner Huffman:

1. Pilih dua simbol x dan y dengan peluang terkecil. Kemudian kombinasikan kedua simbol tersebut (xy) sebagai simpul orangtua dari simbol x dan y , dimana peluangnya adalah jumlah dari peluang anaknya ($P(x) + P(y)$)
2. Kemudian pilih kembali dua simbol yang memiliki peluang terkecil berikutnya, termasuk simpul yang baru. Lakukan hal yang sama seperti pada langkah sebelumnya.
3. Ulangi langkah-langkah sebelumnya sampai menghasilkan simpul akar dimana merupakan kombinasi dari seluruh simbol yang ada dan peluangnya adalah 1.
4. Daun pada pohon biner Huffman menyatakan simbol-simbol yang digunakan pada data sumber. Peng-kodean setiap simpul daun dilakukan dengan memberi label label 0 untuk cabang (kiri) dan label 1 untuk setiap cabang kanan [2].

Dengan algoritma peng-kodean Huffman ini, akan dihasilkan kode yang jumlah bitnya lebih kecil untuk simbol yang memiliki kekerapan kemunculan lebih besar. Selain itu, tidak ada kode simbol yang merupakan kode awalan untuk kode simbol yang lain. Kode Huffman tidak bersifat unik untuk setiap simbol/karakter, artinya kode untuk setiap karakter berbeda-beda pada setiap data sumber tergantung pada kekerapan kemunculannya.

3. TEKNIK PEMBANGKITAN KODE HUFFMAN

Untuk menjelaskan bagaimana teknik membangkitkan kode Huffman untuk berbagai simbol dengan algoritma yang sudah dijelaskan pada bagian sebelumnya, maka akan digunakan suatu contoh sebagai berikut.

Kasus

Bangkitkan kode Huffman untuk setiap kumpulan angka beserta kekerapan kemunculannya seperti pada tabel berikut:

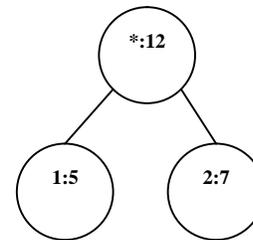
Angka	Frekuensi
1	5
2	7
3	10
4	15
5	20
6	45

Tabel 1: Isi Larik Awal

Penyelesaian

1. Pilih dua buah angka yang memiliki frekuensi terkecil. Dua angka yang dipilih tersebut adalah 1 dan 2 dengan kekerapan kemunculannya 5 dan 7

Pohon Huffman:



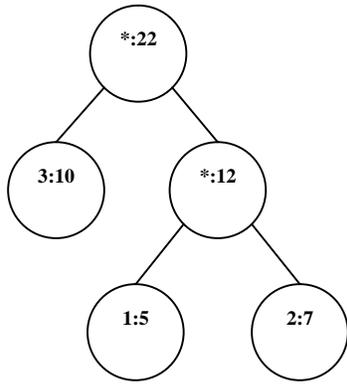
Gambar 1: Pohon Huffman Tahap 1

2. Kedua angka 1 dan 2 dihapus dari larik, sedangkan simpul orangtua baru, dengan frekuensi 12 dimasukkan ke dalam larik.

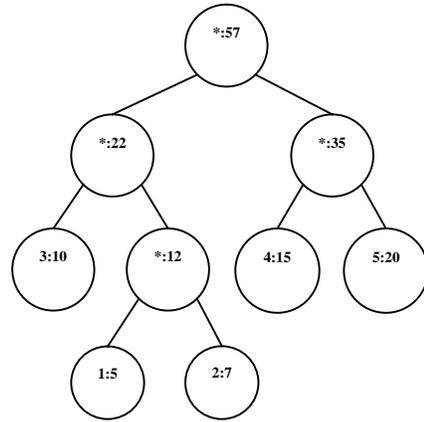
Angka	Frekuensi
3	10
*	12
4	15
5	20
6	45

Tabel 2: Isi Larik Tahap 2

Kemudian ulangi langkah pertama dengan memilih angka yang memiliki frekuensi terkecil, yaitu 3 dan * sehingga menghasilkan:



Gambar 2: Pohon Huffman Tahap 2



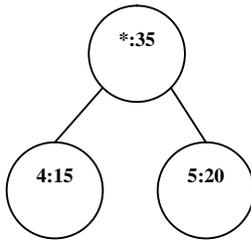
Gambar 4: Pohon Huffman Tahap 4

3. Isi dari larik sekarang adalah:

Angka	Frekuensi
4	15
5	20
*	22
6	45

Tabel 3: Isi Larik Tahap 3

Angka yang ditambahkan pada pohon Huffman adalah 4 dan 5. Karena tidak meneruskan pohon sebelumnya, maka terbentuk upa pohon baru, yaitu:



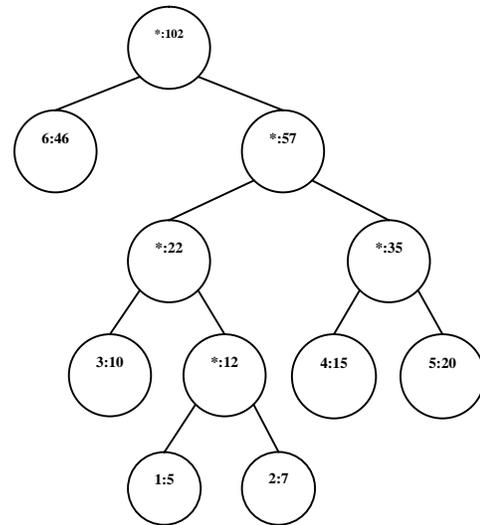
Gambar 3: Pohon Huffman Tahap 3

5. Isi larik pada tahap terakhir ini adalah:

Angka	Frekuensi
6	45
*	57

Tabel 5: Isi Larik Tahap 5

Pohon Huffman lengkap yang terbentuk dari tahap-tahap sebelumnya adalah:



Gambar 5: Pohon Huffman Lengkap

4. Isi larik sekarang adalah:

Angka	Frekuensi
*	22
*	35
6	45

Tabel 4: Isi Larik Tahap 4

Karena 2 angka yang terkecil adalah akar dari kedua upa pohon yang sebelumnya terbentuk, maka kedua upa pohon tersebut digabungkan

Dengan memberi kode 0 untuk cabang kiri dan 1 untuk cabang kanan, maka kode Huffman yang didapat dari pohon Huffman diatas adalah:

Angka	Frekuensi	Kode
1	5	1010
2	7	1011
3	10	100
4	15	110
5	20	111

6	45	0
---	----	---

Tabel 6: Kode Huffman

4. KESIMPULAN

Pemampatan data merupakan hal yang sangat esensial dalam era digital dewasa ini. Dengan melakukan pemampatan data atau pesan, maka akan meningkatkan efektifitas dan efisiensi dalam pemakaian sumber daya yang tersedia di bidang komunikasi dan informasi, serta di bidang penyimpanan data.

Dengan algoritma peng-kodean Huffman, maka proses pemampatan data benar-benar optimal, karena simbol yang kekerapan kemunculannya lebih besar, panjang kodenya lebih pendek sehingga benar-benar mengurangi besar ukuran data yang dimampatkan.

Setiap kode yang dibangkitkan oleh algoritma Huffman, tidak akan merupakan kode awalan diantara kode-kode tersebut. Hal ini dikarenakan penggunaan pohon biner dalam proses pembangkitan kode Huffman dimanacabang yang mengarah sebelah kiri dilambangkan dengan 0 dan yang mengarah ke sebelah kanan dilambangkan dengan 1.

Kode Huffman tidak bersifat unik untuk setiap simbol/karakter tertentu. Kode Huffman untuk suatu karakter yang sama, berbeda-beda tergantung pada data sumbernya.

DAFTAR REFERENSI

- [1] http://en.wikipedia.org/Huffman_coding. Tanggal Akses: 26 Desember 2007
- [2] Munir, Rinaldi, "Diktat Kuliah IF2153 Matematika Diskrit Edisi Keempat", Program Studi Teknik Informatika ITB, Bandung, Indonesia, 2006

