

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan aplikasi komputer saat sekarang ini dalam pengolahan datanya cenderung membutuhkan data dengan ukuran yang besar. Dan hal ini tentu juga berpengaruh pada tuntutan terhadap media penyimpanan yang besar sehingga dapat mendukung jalannya aplikasi secara keseluruhan. Beberapa tahun yang lalu kapasitas media penyimpanan (berupa disk) sebesar 100 MB dapat untuk mendukung aplikasi yang pengolahan datanya tidak membutuhkan data dengan jumlah dan ukuran yang besar. Namun untuk saat ini, kapasitas disk sebesar 100 MB tentu saja kurang mendukung untuk aplikasi yang pengolahan datanya membutuhkan data dengan jumlah dan ukuran yang besar. Karena setiap penambahan media penyimpanan membutuhkan dana, maka diinginkan untuk mengurangi ukuran data sehingga memberikan keuntungan pada penghematan penggunaan media penyimpanan.

Salah satu cara untuk mengurangi ukuran dari data adalah dengan menggunakan kompresi data pada data yang akan disimpan. Hal ini berarti data yang akan disimpan dikompresi terlebih dahulu, sedemikian sehingga ukuran data secara keseluruhan menjadi lebih kecil.

Untuk data yang berupa teks, maka perlu diketahui bahwa suatu teks tersusun dari beberapa huruf yang masing-masing huruf jumlahnya tidak merata. Ada huruf yang muncul dalam teks lebih sering dan ada pula huruf yang jarang muncul dalam teks. Jika digunakan skema pengkodean yang menggunakan string

bit terpendek untuk huruf yang sering muncul dan string bit yang lebih panjang untuk huruf yang jarang muncul, maka rata-rata dari bit yang diinginkan untuk disimpan dapat dikurangi jumlahnya

1.2 Permasalahan

Pengolahan data dalam suatu aplikasi memerlukan data yang dalam hal ini berupa pesan yang ukurannya seefisien mungkin. Oleh karena itu diperlukan suatu metode untuk pengkodean pesan yang menghasilkan kode biner yang minimum jumlahnya. Selain menggunakan standar pengkodean ASCII, ada salah satu cara pengkodean yang dapat digunakan untuk pengkodean pesan yaitu dengan algoritma Huffman.

Permasalahan yang akan dibahas dalam tugas akhir ini adalah sebagai berikut :

1. Bagaimana aplikasi dari algoritma Huffman dalam menemukan kode bilangan biner dari teks atau pesan yang diinputkan.
2. Bagaimana mengkodekan pesan tersebut menjadi suatu barisan bilangan biner.
3. Bagaimana mengkodekan kembali barisan bilangan biner yang dihasilkan menjadi suatu pesan semula.
4. Bagaimana menentukan rasio kompresi.

1.3 Pembatasan Masalah

Pada penulisan tugas akhir ini, penulis menggunakan bahasa pemrograman C++ untuk aplikasi algoritma Huffman dalam kompresi data. Data yang diinputkan merupakan suatu pesan yang berupa kata.

1.4 Metodologi Penelitian

Pada penulisan tugas akhir ini menggunakan metode studi pustaka dalam penyusunan dan pembuatannya.

1.5 Garis Besar Pemecahan Masalah

Algoritma Huffman merupakan salah satu bentuk penyajian dari pohon binary yang tepat dan efisien untuk pengkodean pesan. Untuk suatu pesan yang terdiri dari n simbol huruf, maka akan dikodekan menjadi string bit (bit didefinisikan sebagai 0 atau 1) dengan mencocokkan kode string bit dari setiap simbol huruf yang ada pada pesan.

- Contoh : suatu pesan terdiri dari 4 huruf A, B, C dan D. Jika kode dengan dua bit dipasangkan untuk setiap simbol huruf sebagai berikut :

Simbol Huruf	Kode
A	00
B	01
C	10
D	11

maka kode untuk pesan ABACCDAA akan menjadi 00010010101100, yang memuat 14 bit.

Dari contoh di atas, huruf B dan D kelihatan hanya 1 kali muncul dalam pesan, dan huruf A kelihatan 3 kali muncul dalam pesan, serta huruf C kelihatan 2 kali muncul dalam pesan. Dengan algoritma Huffman akan ditentukan bit terakhir untuk 2 huruf yang mempunyai frekuensi terkecil, menjadi suatu simbol huruf gabungan dengan frekuensi baru yang merupakan gabungan dari frekuensi

masing-masing simbol huruf tersebut. Langkah di atas diulang lagi sedemikian sehingga didapat suatu simbol huruf gabungan yang mencakup semua simbol huruf yang menyusun pesan.

Metode pengembangan suatu skema pengkodean yang optimal berdasarkan frekuensi dari setiap simbol huruf dari suatu pesan adalah sebagai berikut :

- Menemukan 2 simbol huruf yang frekuensinya paling kecil, dimana pada contoh yaitu B dan D dan bit terakhir dibedakan diantara keduanya yaitu 0 untuk B dan 1 untuk D.
- Mengkombinasikan 2 simbol tersebut menjadi 1 simbol yaitu BD yang memiliki kode yang mewakili bahwa simbol tersebut B atau D, maka frekuensi BD menjadi 2.
- Sekarang ada 3 simbol huruf yaitu A (frekuensi 3), C (frekuensi 2) dan BD (frekuensi 2). Kemudian dipilih lagi 2 simbol dengan frekuensi terkecil yaitu C dan BD. Bit terakhir dari kodenya akan dibedakan antara keduanya yaitu 0 untuk C dan 1 untuk BD. Dua simbol tersebut dikombinasikan menjadi simbol tunggal yaitu CBD dengan frekuensi 4.
- Sekarang ada 2 simbol lagi yaitu A dan CBD. Keduanya dikombinasikan menjadi simbol tunggal ACBD. Bit terakhir dari kode untuk A dan CBD akan dibedakan antara keduanya yaitu 0 untuk A dan 1 untuk CBD.

Jadi kode biner dari masing-masing simbol huruf yang didapat dari skema pengkodean diatas adalah sebagai berikut :

Simbol Huruf	Kode
A	0
B	110
C	10
D	111

Dengan menggunakan kode biner pada tabel di atas, maka pesan ABACCCA dikodekan menjadi 0110010101110 yang memuat 13 bit.

1.6 Garis Besar Sistematika Penulisan

BAB I. Pendahuluan

- 1.1 Latar Belakang
- 1.2 Permasalahan
- 1.3 Pembatasan Masalah
- 1.4 Metodologi Penelitian
- 1.5 Garis Besar Pemecahan Masalah
- 1.6 Garis Besar Sistematika Penulisan

BAB II Materi Penunjang

- 2.1 Tree
- 2.2 Binary Tree
- 2.3 Bahasa Pemrograman C++
 - 2.3.1 Pernyataan if dan if...else
 - 2.3.2 Arrays, Character and String
 - 2.3.3 Loop
 - 2.3.4 Procedure dan Functions

BAB III Pembahasan

- 3.1 Kompresi Data
- 3.2 Algoritma Huffman
- 3.3 Menentukan Kode Biner dari Suatu Simbol Huruf dari Suatu Pesan
- 3.4 Menentukan Kode Biner Menggunakan Binary Tree
- 3.5 Aplikasi Algoritma Huffman dalam Kompresi Data dengan Menggunakan Bahasa Pemrograman C++
 - 3.5.1 Menentukan Frekuensi dari Masing-Masing Simbol Huruf dari Suatu Pesan
 - 3.5.2 Menentukan Kode Biner dari Suatu Simbol Huruf dari Suatu Pesan dengan Algoritma Huffman
 - 3.5.3 Menentukan Barisan Kode Biner dari Suatu Pesan yang Diinputkan
 - 3.5.4 Menentukan Pesan dari Barisan Kode Biner yang Diinputkan
 - 3.5.5 Menentukan Rasio Kompresi

BAB IV Kesimpulan