

PENERAPAN METODE HUFFMAN DALAM PEMAMPATAN CITRA DIGITAL

Edy Victor Haryanto

Universitas Potensi Utama, Jl. K.L. Yos Sudarso Km. 6,5 No. 3 A Tj Mulia – Medan
edy@potensi-utama.ac.id, edyvictor@gmail.com

abstrak

Citra adalah representasi dari sebuah objek. Ukuran file sebuah gambar sangat berpengaruh terhadap kapasitas tempat penyimpanan, semakin besar kapasitas file sebuah gambar maka semakin besar pula dibutuhkan ruang untuk penyimpanan gambar tersebut, pada penelitian ini dibuat bagaimana sebuah gambar yang kapasitasnya dapat diperkecil sehingga ruang atau media untuk menyimpannya juga kecil dengan cara di mampatkan dan gambar yang dihasilkan pun juga akan berpengaruh. Bahasa pemrograman yang digunakan untuk membuat aplikasi ini adalah java. Citra ini juga melakukan pengelompokan nilai derajat keabuan pixel yang sama dari citra dan menyusun secara berurut agar dapat dikompresi dengan metode Huffman, serta melakukan proses dekompresi dengan mengembalikan citra yang sudah dikompresi.

Kata kunci : Kompresi, Citra digital, Metode Huffman, Java

1. Pendahuluan

Citra merupakan representasi digital dari objek gambar, yang tidak lepas dari kebutuhan manusia. Pada umumnya representasi citra membutuhkan memori yang cukup besar, khususnya citra warna. Hal ini tentu sangat mempengaruhi ketersediaan tempat (*space*) maupun pengolahan data khususnya data citra dalam perkembangan ilmu pengetahuan khususnya dibidang ilmu komputer. Permasalahan lain yang sering timbul dalam pengolahan citra dengan ukuran besar adalah ketika dilakukan proses *transmisi* (pengiriman gambar) melalui media komunikasi. Ini tentu akan memperlambat waktu pengiriman. Untuk itu perlu dikembangkan aplikasi untuk kompresi citra yang bertujuan untuk minimalisasi memori.

Citra dalam format BMP ada tiga macam : citra biner, citra berwarna, dan citra hitam-putih (grayscale). Citra biner hanya mempunyai dua nilai keabuan, yaitu 0 dan 1. Oleh karena itu, 1 bit sudah cukup untuk merepresentasikan nilai pixel. Citra berwarna adalah citra yang lebih umum. Warna yang terlihat pada citra bitmap merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru. Setiap pixel disusun oleh tiga komponen warna : R (red), G (green), dan B (blue). Kombinasi dari tiga warna RGB tersebut menghasilkan warna yang khas untuk pixel yang bersangkutan.

Nilai setiap pixel tidak menyatakan derajat keabuan secara langsung, tetapi nilai pixel menyatakan indeks tabel RGB yang membuat nilai keabuan merah (R), nilai keabuan hijau (G), dan nilai keabuan biru (B) untuk pixel yang bersangkutan. Pada citra hitam-putih, nilai $R = G = B$ untuk menyatakan bahwa citra hitam-putih

hanya mempunyai satu kanal warna. Citra hitam-putih umumnya adalah citra 8-bit.

Citra yang lebih kaya warna adalah citra 24-bit. Setiap pixel panjangnya 24 bit, karena setiap pixel langsung menyatakan komponen warna merah, komponen warna hijau, dan komponen warna biru. Masing-masing komponen panjangnya 8 bit. Citra 24-bit disebut juga citra 16 juta warna.

Adapun batasan-batasan dan manfaat pada penelitian ini adalah image yang digunakan adalah berekstension jpg dan bmp, ukuran citra nya adalah 80 x 80 dan dalam format RGB, metode kompresinya huffman, dapat memperkecil ukuran file gambar.

Penelitian Terkait

Maria juga dalam penelitiannya membandingkan beberapa metode kompresi terhadap citra antara lain, Huffman, LZW, RLE dan Shannon Fano dan kesimpulan dari pengujian tersebut bahwa metode LZW menghasilkan ukuran kapasitas yang lebih kecil dibandingkan dengan metode yang lain[1].

Erik dalam penelitiannya membandingkan beberapa metode kompresi antara lain lossless dan lossy, yang digunakan untuk memperkecil ruang penyimpanan yang dibutuhkan sebuah file, parameter yang digunakan untuk membandingkan metode tersebut adalah : kualitas kompresi, rasio kompresi, ruang penyimpanan[2].

Menurut Rismon dalam penelitiannya mengungkapkan bahwa dengan menggunakan parameter laju bit dan PSNR lebih baik dalam mengkonversi citra digital dengan metode Wavelet dapat mengkompresi sebuah citra yang kapasitas citra tersebut menghasilkan sampai 2/5

dari kapasitas ukuran citra aslinya dan memiliki derajat halus[3].

Kompresi Data

Proses kompresi merupakan proses mereduksi ukuran suatu data untuk menghasilkan representasi digital yang padat atau mampat (compact) namun tetap mewakili kuantitas informasi yang terkandung pada data. Pada citra, video, dan audio, kompresi mengarah pada minimisasi jumlah bit rate untuk representasi digital.

Data dan informasi adalah dua hal yang berbeda. Pada data terkandung suatu informasi. Namun tidak semua data terdapat bagian-bagian data yang berulang untuk mewakili informasi yang sama. Bagian data yang tidak terkait atau bagian data yang berulang disebut dengan data berlebihan (redundancy data). Tujuan dari pada kompresi data adalah untuk mengurangi data berlebihan sehingga ukuran data menjadi lebih kecil dan lebih ringan dalam proses transmisi[4].

Algoritma Huffman

Metode huffman adalah metode pengkodean yang telah banyak diterapkan untuk aplikasi kompresi citra. Seperti metode shannon fano, metode huffman juga membentuk pohon atas dasar probabilitas setiap simbol, namun teknik pembentukan pohonnya berbeda. Berikut ini langkah-langkah algoritma huffman. (Darma Putra : 2010 : 279)

1. Data dianalisis dahulu dengan cara membuat tabel frekuensi kemunculan setiap simbol ASCII, tabel frekuensi tersebut memiliki atribut berupa simbol ASCII dan frekuensi.
2. Dua data yang memiliki frekuensi kemunculan paling kecil dipilih sebagai simpul pertama pohon huffman.
3. Data dua simpul dibuat simpul induk yang mencatat jumlah frekuensi dua simpul pertama.
4. Dua simpul tersebut dihapus dari tabel digantikan oleh simpul induk. Simpul ini kemudian dijadikan acuan untuk membentuk pohon.
5. Langkah 3-5 dilakukan berulang-ulang hingga isi tabel tinggal satu saja. Data inilah yang akan menjadi simpul bebas atau simpul akar.
6. Setiap simpul yang terletak pada cabang kiri (simpul dengan frekuensi lebih besar) diberikan nilai 0 dan simpul yang terletak pada cabang kanan (Simpul dengan frekuensi lebih kecil) diberikan nilai 1.
7. Pembacaan dilakukan dari simpul akar ke simpul daun dengan memperhatikan nilai setiap cabang[4].

Algoritma Huffman diperkenalkan oleh David A. Huffman seorang mahasiswa MIT dalam papernya yang berjudul "A Method for the Construction of Minimum-Redundancy Codes" dan diterbitkan pada tahun 1952. Prinsip kode Huffman adalah karakter yang paling sering muncul di dalam data dikodekan dengan kode yang jumlah bitnya lebih sedikit, sedangkan karakter yang jarang muncul dikodekan dengan kode yang jumlah bitnya lebih panjang. Algoritma Huffman menggunakan tabel frekuensi kemunculan karakter untuk menggambarkan setiap karakter menjadi kode atau string biner. Kode atau string biner yang digunakan untuk mengkodekan setiap karakter dinamakan kode Huffman.

Algoritma Huffman yang akan dibahas dalam tulisan ini adalah algoritma Huffman yang menggunakan metode statik, yaitu metode yang menggunakan peta kode yang sama, metode ini membutuhkan dua fase, yaitu fase pertama untuk menghitung frekuensi kemunculan tiap karakter dan menentukan peta kodenya dan fase ke dua untuk mengubah pesan atau data menjadi kumpulan kode yang akan ditransmisikan.

Kode optimal untuk sebuah file digambarkan dengan pohon biner penuh, di mana setiap node bukan daun mempunyai dua anak. Pada awalnya Huffman hanya mengkodekan karakter menggunakan pohon biner biasa, namun setelah itu Huffman menemukan bahwa penggunaan algoritma Greedy dapat membentuk kode prefiks yang optimal. Penggunaan algoritma Greedy pada algoritma Huffman adalah pada saat pemilihan dua pohon dengan frekuensi terkecil dalam membuat pohon Huffman. Cost yang digunakan untuk menggabungkan dua buah pohon pada akar setara dengan jumlah frekuensi dua buah pohon yang digabungkan. Oleh karena itu, total cost pembentukan pohon Huffman adalah jumlah seluruh penggabungan daun-daun.

Algoritma Huffman menggunakan struktur data string sebagai masukan, struktur data binary tree pada pembentukan pohon biner dan array untuk mendeklarasikan kumpulan variabel yang bertipe sama[4].

2. Perancangan

Dalam penelitian ini, yang dibahas adalah pengkompresian gambar (citra) dengan menggunakan metode Huffman. Sebagaimana telah diuraikan dalam bab sebelumnya, pada dasarnya citra (*.jpg atau *.bmp) dibentuk dengan mendefinisikan nomor warna untuk setiap pixel yang terdapat dalam daerah gambar atau citra. Dengan demikian banyak pendefinisian warna adalah hasil perkalian lebar dengan tinggi citra. Warna yang digunakan tentu kemungkinan besar ada yang berulang untuk lokasi yang berbeda.

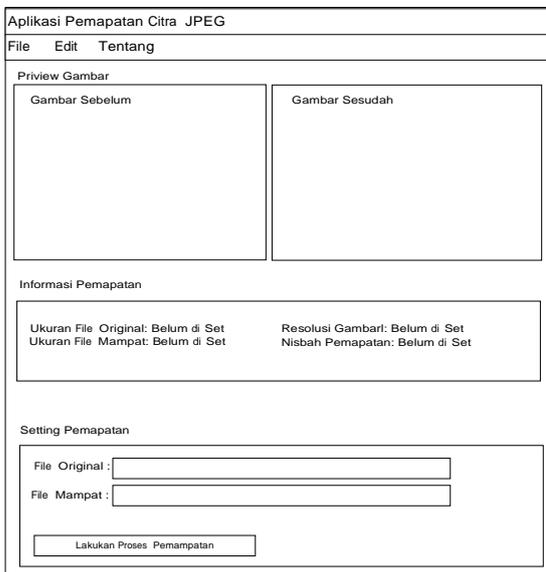
Dalam bagian ini akan dilakukan pembahasan dalam menerapkan metode Huffman untuk melakukan kompresi citra berdasarkan urutan kemunculan warna dalam citra. Hasil kompresi citra akan disimpan dalam suatu file (*.jpg_mapat).

2.1. Analisis Algoritma Huffman

Algoritma Huffman membangun pohon biner dari bawah ke atas yaitu dimulai dari daun hingga ke akar. Pembentukan ini dimulai dengan urutan karakter-karakter yang dinyatakan sebagai daun-daun pohon atau pohon bersimpul tunggal. Karakter-karakter digabungkan berkali-kali dengan karakter atau subpohon lainnya yang mempunyai frekuensi kemunculan karakter paling kecil untuk membentuk subpohon baru. Proses ini terus berlanjut hingga semua karakter atau subpohon telah digabungkan pada sebuah akar. Untuk membentuk kode Huffman, telusuri pohon biner dari akar ke daun. Barisan angka 0 dan 1 pada sisi pohon dari akar ke daun menyatakan kode Huffman untuk karakter yang bersesuaian.

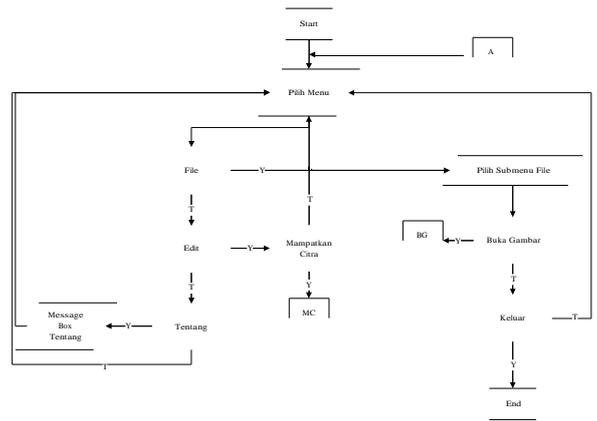
2.2. Perancangan Pemampatan Citra

Menu utama adalah tampilan awal dari program dimana berisikan menu dan tombol yang mempermudah untuk memilih jenis citra yang akan di kompresi dengan metode *Huffman*

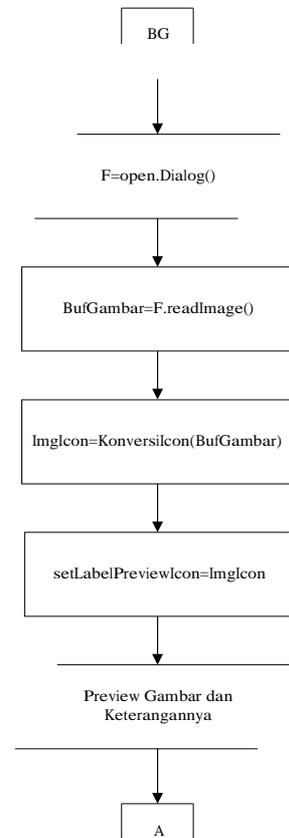


Gambar 1 : Perancangan Menu Utama

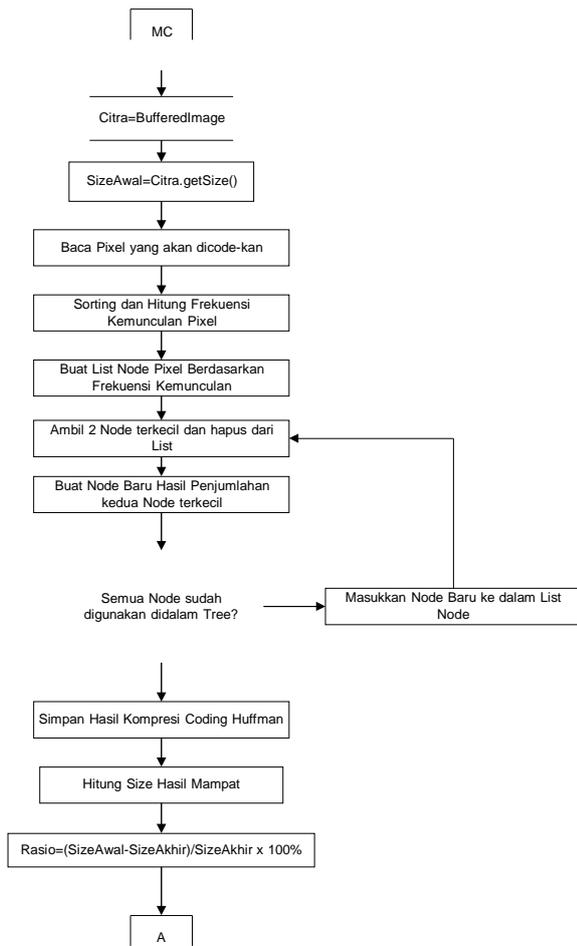
2.3. Flowchart



Gambar 2 : Flowchart Menu Utama



Gambar 3 : Flowchart Buka Gambar



Gambar 4 : Flowchart Pemampatan Citra

3. Hasil dan Pembahasan

Tampilan Menu Utama

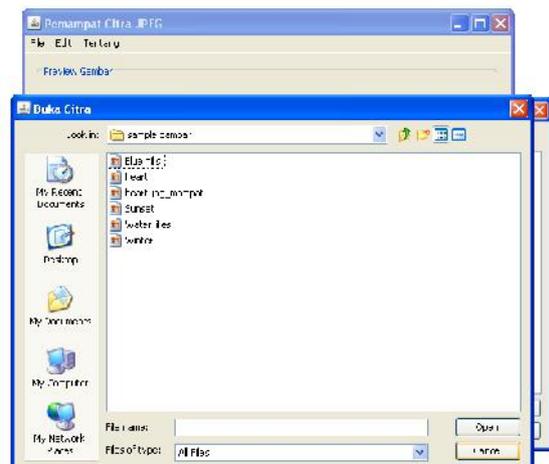
Pada menu utama ini berfungsi sebagai tampilan utama dari program aplikasi pemampatan citra digital. Di dalam menu utama ini terdiri dari proses pemampatan citra.



Gambar 5 : Menu Utama

Tampilan Buka File Citra

Pada tampilan buka file citra ini berfungsi untuk mengambil gambar citra untuk pengujian pemampatan citra.



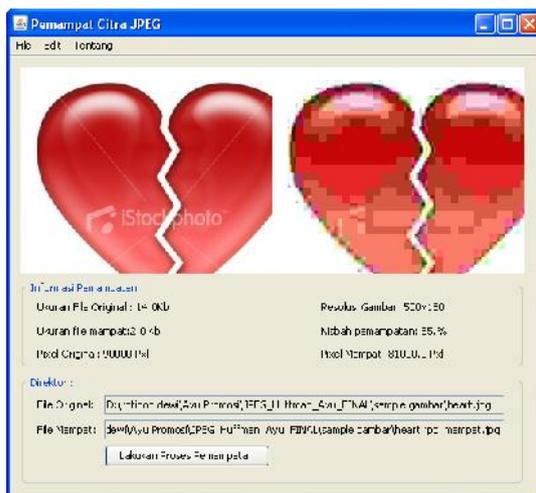
Gambar 6 : Buka File Citra



Gambar 7 : Tampilan Hasil Buka File Citra

Tampilan Hasil Proses Pemampatan

Pada tampilan proses ini berfungsi untuk membuat hasil file citra menjadi kecil.



Gambar 8 : Tampilan Hasil Citra Sebelum dan Sesudah dimampatkan

Hasil dari pemampatan akan memperlihatkan bahwa baik itu pixel dan ukuran kapasitas gambar asli dan hasilnya akan menjadi lebih kecil atau berbeda.



Gambar 9 : Tampilan Hasil Pemampatan Citra

4. Kesimpulan dan Saran

4.1. Kesimpulan

Dari hasil pembuatan aplikasi kompresi citra dengan metode huffman ini, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Perangkat lunak kompresi citra dengan metode huffman dapat melakukan kompresi file citra dengan format bmp maupun jpg.
2. Kompresi citra dengan metode huffman dapat memperkecil ukuran file citra, sehingga dapat menghemat ruang tempat penyimpanan (Storage).
3. Gambar yang telah dimampatkan dapat dikembalikan kembali ke ukuran semula atau file aslinya.

DAFTAR PUSTAKA

- [1] Roslin, Maria, 2013, "Perbandingan Algoritma Kompresi Terhadap Objek Citra Menggunakan Java", SEMANTIK 2013.
- [2] Iman, Erik dan Sri Hartati, 2010, "Ikhtisar Kompresi Citra" Prosiding SNAST Periode II, Desember 2010.
- [3] Rismon H, 2003, "Kompresi Citra Digital Berbasis Wavelet : Tinjauan PSNR dan Laju Bit" Jurnal Informatika, Vol. 4, No. 2, 2003.
- [4] Putra, Dharma, 2010, *Dasar Pengolahan Citra*, Gunadarma, Jakarta.
- [5] Priyatno, Budi BN, 2011 : 1, *Pengolahan Citra Digital*.
- [6] Sulistiyani, Sri, 2010, *Membangun GUI dengan JAVA Netbean 6.5*, Andi, Yogyakarta.
- [7] Departemen Pendidikan Nasional, *Kamus Besar Bahasa Indonesia, Kamus Pusat Bahasa, Edisi ketiga*, Balai Pustaka, Jakarta 2001.

<http://blog.trisakti.ac.id/labkomputer/file/2010/05/MODULDKP.pdf>