

## Kompresi Citra dari Format BMP ke Format PNG

Hendri  
STMIK TIME, Jl. Merbabu No 32 AA-BB Medan  
Email : h4ndr7@hotmail.com

### Abstrak

Pengolahan Citra merupakan suatu bidang ilmu yang semakin berkembang dan penting pada era informasi ini. Citra yang disimpan secara digital membutuhkan kapasitas media penyimpanan dan bandwidth yang relatif besar. Untuk efisiensi ruang penyimpanan dan mengurangi beban transmisi, maka dibutuhkan format kompresi citra digital. Salah satu kompresi citra digital yang bersifat lossless ini adalah Portable Network Graphic (PNG). Hasil uji coba kompresi ke format PNG menunjukkan bahwa file citra dapat dikompresi dengan rasio antara 30% - 70%. Hal ini menunjukkan bahwa format PNG sangat tepat digunakan untuk menyimpan data citra yang dikompres secara lossless dan mendukung kompresi pada tingkat kedalaman warna yang tinggi mencapai 48bit)

**Kata Kunci** : Citra, PNG, Lossless

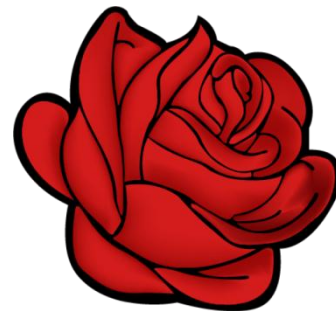
### 1. PENDAHULUAN

Keterbatasan kapasitas media penyimpanan dan bandwidth pada media transmisi merupakan tantangan yang dihadapi pada hal penyimpanan maupun transmisi data, untuk mengatasi hal tersebut, data yang akan disimpan ataupun dikirim harus dapat dimampatkan (dikompres). Ada beberapa format data citra digital yang umum digunakan. Masing-masing format memberikan kelebihan dan keuntungan masing-masing. Salah satu format citra digital yang umum digunakan di internet adalah format GIF, namun mempunyai keterbatasan dalam kedalaman warna yaitu maksimum 256 warna. Untuk menggantikan penggunaan format GIF diinternet, dikembangkan suatu format citra digital yang baru. Format baru ini diberi nama dengan PNG (Portable Network Graphic). Format ini memiliki atribut-atribut yang terdapat pada GIF dan juga beberapa keunggulan lainnya. Keunggulan dari format PNG antara lain mampu menangani kedalaman warna sampai 48bit, mendukung transparansi dan portable.

### 2. LANDASAN TEORI

Ada dua format utama untuk citra digital pada komputer, yaitu pertama format raster, yang terdiri dari titik-titik berwarna dan disimpan pada array dua dimensi, dan kedua, format vector adalah terdiri

dari garis dan lingkaran serta elemen dasar lainnya yang dapat dibedakan satu dengan yang lainnya. Adapun tujuan dari pengelompokan format citra digital pada komputer ini adalah disesuaikan terhadap penggunaan citra tersebut. Format vector pada prinsipnya cocok untuk digunakan untuk citra yang terdiri dari grafik dan text yang sederhana seperti logo. Sedangkan format raster lebih cocok untuk digunakan untuk gambar natural seperti gambar pemandangan. Citra yang tersimpan pada format vector ataupun raster selalu ditampilkan dalam format raster, seperti pada monitor dan rinter, dengan pengecualian terhadap plotter dan monitor vector. Pada penulisan ini, akan membahas mengenai format citra PNG, yaitu format citra yang terkelompok dalam kelas citra raster.



Contoh Citra Format Vector



Contoh Citra Format Raster

Citra raster terdiri dari kumpulan titik-titik yang disebut sebagai picture elements atau sering juga disingkat dengan pixel. Secara umum pixel digunakan untuk menyatakan dimensi dari citra pada komputer seperti 640x480, 800x600. Nilai nilai tersebut menyatakan jumlah pixel pada dimensi panjang dan lebar pada citra tersebut. Selain dimensi panjang dan lebar, citra digital juga memiliki atribut kedalaman warna. Semakin dalam warna sebuah citra digital semakin banyak jumlah warna yang membentuk citra digital tersebut.

Kedalaman dari setiap pixel diukur dalam bit, disebut dengan istilah "bit per pixel". Untuk menghitung besar kapasitas penyimpanan citra digital dapat menggunakan rumus berikut :

$$\text{Panjang} \times \text{lebar} \times \text{kedalaman warna}$$

Dengan rumus diatas, ukuran citra digital dengan dimensi 320x240, dengan kedalaman warna 24 bit, adalah:

$$320 \times 240 \times 24 = 1.843.200 \text{ bits}$$

Apabila diubah kedalam satuan byte menjadi  $1.843.200/8 = 230.400$  bytes, dan dalam satuan kilobytes adalah  $230.400/1024 = 255$  kilobytes.

Format Digital disimpan dalam format tertentu. Format citra digital yang paling sederhana adalah format bitmap, yaitu format data citra yang sama dengan format data yang disimpan pada memori komputer waktu menampilkan citra

Citra digital mengandung data yang bersifat redundan, yaitu data yang bernilai sama muncul berulang-ulang. Sering kali suatu pixel memiliki nilai yang sama dengan pixel disekelilingnya. Oleh karena itu, citra digital dapat diperkecil ukurannya dengan melakukan kompresi terhadap citra digital tersebut.

Ada beberapa format citra digital yang sering digunakan, antara lain :

#### 1. BMP

Merupakan format citra digital yang menyimpan data citra digital tanpa melakukan pemrosesan terhadap citra tersebut. Representasi data citra dalam format ini sama dengan data citra yang digunakan untuk representasi data citra pada memori komputer.

#### 2. GIF (Graphich Interchange Format)

Adalah format citra digital yang mendukung kompresi lossless, yaitu kompresi yang tidak menghilangkan informasi citra digital. Format ini tidak mengurangi kualitas citra digital dibandingkan dengan aslinya, namun kekurangan terbesarnya adalah terbatasnya kedalaman warna yang didukung yaitu hanya sampai 256 warna.

#### 3. JPEG ( Joint Picture Expert Group)

Adalah format citra digital yang menggunakan kompresi lossy. Citra digital yang disimpan dengan format ini akan mengalami penurunan kualitas bila dibandingkan dengan citra aslinya. Format ini sangat cocok digunakan

untuk menyimpan citra fotografi, karena ukuran data yang dihasilkan kecil dan kualitas yang baik untuk penglihatan manusia.

#### 4. PNG (Portable Network Graphic)

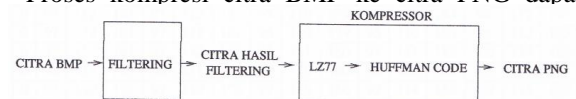
PNG hampir sama dengan format GIF, bahkan format PNG ini didesain untuk menggantikan format GIF, dengan beberapa peningkatan. Format PNG menggunakan kompresi lossless untuk mengkompresi citra digital.

### 3. ANALISA DAN PERANCANGAN

Proses kompresi ke format PNG terbagi menjadi dua tahap,yaitu tahap filtering dan kemudian diikuti kompresi terhadap hasil dari tahap filtering. Sedangkan untuk proses dekompresi, tahap yang dilalui adalah dekompresi citra dan kemudian filtering terhadap hasil dekompresi

#### 3.1. KOMPRESI

Proses kompresi citra BMP ke citra PNG dapat



#### Filtering

Sebelum memasuki proses kompresi data citra, citra akan melewati tahap filtering, dimana data citra akan ditransformasi sedemikian rupa untuk meningkatkan rasio kompresi. Ada 5 tahapan filtering :

##### 1. Filter type 0 : None

Pada metode filter 0 (none), data citra tidak dilakukan proses filtering, hanya menambahkan tipe filter pada awal data citra ini.

##### 2. Filter type 1 : Sub

Filter ini menghasilkan selisih antara masing-masing byte dan nilai byte dari pixel sebelumnya.

Untuk menghitung Sub, gunakan rumus berikut untuk tiap byte pada scanline :

$$\text{Sub}(x) = \text{Raw}(x) - \text{Raw}(x-\text{bpp})$$

X bernilai 0 sampai jumlah byte pada scanline kurang 1. Raw() merupakan byte mentah pada posisi byte ke x dalam scanline, dan bpp adalah jumlah bytes perpixel, dibulatkan ke atas.

Untuk  $x < 0$  maka  $\text{Raw}(x) = 0$

##### 3. Filter type 2 : Up

Untuk menghitung filter Up, digunakan rumus berikut pada setiap byte pada scanline :

$$Up(x)=Raw(x) - Prior(x)$$

Dimana x berkisar dari 0 sampai jumlah bytes dalam scanline kurang 1, Raw() merupakan byte mentah pada posisi byte ke x dalam scanline, dan Prior() mengacu pada byte yang belum difilter pada scanline sebelumnya. Untuk scanline pertama ataupun citra interlaced, Prior(x) bernilai 0.

#### 4. Filter type 3: Average

Filter ini menggunakan dua pixel disekelilingnya (kiri dan atas) untuk memprediksi nilai pixel.

Gunakan rumus berikut pada setiap byte pada scanline :

$$Average(x)=Raw(x)-Floor((Raw(x-bpp)+Prior(x))/2)$$

Untuk x antara 0 sampai jumlah bytes pada scanline kurang satu. Raw() merujuk pada byte mentah ke x pada scanline. Prior() merujuk kepada bytes yang belum difilter pada scanline sebelumnya, dan bpp adalah jumlah bytes perpixel, dibulatkan ke atas.

Untuk  $x < 0$  maka  $Raw(x)=0$  untuk scanline pertama ataupun pada citra interlaced, Prior(x) bernilai 0.

#### 5. Filter type 4 : Paeth

Filter Paeth() menghitung fungsi linier sederhana dari tiga pixel yang berdekatan(kiri, atas) kemudian memilih nilai pixel yang paling dekat dengan hasil fungsi sebagai predictor. Teknik ini diciptakan oleh Alan W. Paeth.

Rumus untuk menghitung filter Paeth() ini adalah sebagai berikut:

$$Paeth(x)=Raw(x)-PaethPredi(Raw(x-bpp), Prior(x), Prior(x-bpp))$$

#### Kompresi LZ77

Setelah melewati proses filtering, output dari proses filtering tersebut akan dikompres dengan menggunakan algoritma LZ77.

Berikut ini adalah algoritma kompresi LZ77 yang digunakan pada proses kompresi dalam format citra PNG.

```
while (lookaheadbuffer tidak kosong) {
    ambil pointer ke (position, match) dari
    match terpanjang dalam windows yang
    dibandingkan dengan lookaheadbuffer;
    if (length >
    MINIMUM_MATCH_LENGTH) {
        output (length, distance);
        majukan posisi window sebanyak
        length;
    } else {
        output karakter pertama pada
        lookaheadbuffer; majukan posisi
        window 1 karakter;
    }
}
```

#### Huffman Code

Huffman Code adalah sebuah algoritma kompresi data yang bersifat lossless. Ide di belakang algoritma ini adalah menggunakan pola bit yang lebih pendek untuk karakter yang sering muncul pada suatu data. Pola bit ini disebut dengan prefix code.

Prefix code didefinisikan dalam bentuk pohon biner, dimana setiap node yang bukan leaf merupakan kode (dilambangkan oleh angka 0 dan 1), dan leaf merupakan simbol (dilambangkan oleh huruf A,B,C dan D). Representasi Huffman Tree dalam bentuk pohon biner dapat dilihat pada gambar dibawah.

#### 6. HASIL

Perangkat lunak yang dirancang, memiliki dua bagian utama yaitu front-end dan back-end. Bagian front-end berhubungan dengan antar muka

perangkat lunak dengan pemakai baik itu dalam bentuk baris perintah (command line) ataupun dalam bentuk grafik (graphical user interface). Bagian back-end berhubungan dengan pemanggilan fungsi-fungsi untuk membaca dan menulis file citra dalam format PNG maupun dalam format BMP.

Berikut ini adalah algoritma untuk mengkompresi citra dari format BMP ke dalam format PNG :

1. Buka file input (BMP) dalam modus baca, STOP apabila file input tidak bisa dibuka/dibaca.
2. Baca header file BMP dari file input
3. Periksa apakah signature file input (BMP) sesuai dengan signature file BMP. Jika tidak sesuai, STOP dan cetak pesan kesalahan.
4. Buka file output (PNG) dalam modus penulisan, STOP apabila file output tidak bisa dibuka untuk ditulis
5. Jika file input adalah file citra dengan tipe warna indexed color, baca palet warna dari file input (BMP)
6. Baca data citra file input (BMP)
7. Tutup file input (BMP)
8. Isikan nilai chunk file output (PNG) sesuai dengan nilai yang dibaca dari header file input (BMP)
9. Tuliskan chunk IHDR, PLTE (untuk tipe warna indexed color), pHYs ke file out-put (PNG)
10. Lakukan kompresi terhadap data citra file input (BMP) yang telah dibaca, kemudian tuliskan hasilnya ke file output (PNG)
11. Tuliskan trailer pada file output (PNG)
12. Tutup file output (PNG)

Algoritma diatas merupakan algoritma untuk front-end. Algoritma tersebut akan memanggil fungsi-fungsi yang berada pada lapisan perekat (glue layer) yang akan menghubungkan permintaan atau perintah dari front-end dengan fungsi-fungsi yang berada pada back-end.

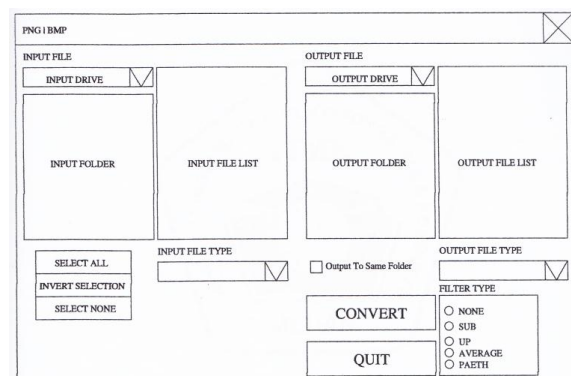
Berikut dijelaskan mengenai langkah-langkah yang dilakukan pada back-end untuk menuliskan data citra ke dalam format PNG

1. Inisialisasikan struktur data untuk menampung data citra yang akan ditulis dan struktur data (pada penjelasan selanjutnya kedua struktur data ini akan disebut dengan struktur data PNG) untuk menampung informasi mengenai karakteristik citra (kedalaman warna, resolusi, tipe warna). Struktur data diinisialisasikan dengan menggunakan fungsi `png_create_write_struct` untuk struktur data citra yang akan ditulis dan

`png_create_info_struct` untuk struktur data informasi karakteristik citra yang akan ditulis. Apabila kedua struktur data tersebut tidak bisa diinisialisasi, maka program akan berhenti dengan mengembalikan status kesalahan.

2. Buka dan inisialisasikan file output dengan memanggil fungsi `png_init_io`.
3. Set tingkat kompresi yang diinginkan dengan menggunakan `png_set_compression_level`.
4. Tuliskan header (chunk IHDR) ke struktur data PNG, fungsi yang digunakan adalah `png_set_IHDR`.
5. Panggil fungsi `png_set_filter` untuk menentukan tipe filter yang akan digunakan pada proses kompresi.
6. Jika tipe warna indexed color, maka tuliskan informasi palet warna(chunk PLTE) ke struktur data PNG dengan fungsi `png_set_PLTE`
7. Panggil fungsi `png_set_pHYs` untuk menuliskan chunk pHYs ke struktur data PNG.
8. Tuliskan struktur data PNG ke file output dengan memanggil fungsi `png_write_info`
9. Encode citra input ke dalam format PNG dengan fungsi `png_write_image` dan akhiri dengan menuliskan trailer (chunk IEND) ke file output menggunakan fungsi `png_write_end`.
10. Bebaskan memory yang telah digunakan dengan menghapus struktur data PNG dengan memanggil `png_destroy_write_struct`.

Tampilan antar muka untuk kompresi citra ke format PNG adalah seperti yang dibawah ini.



## 7. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Kompresi yang digunakan pada format PNG merupakan kompresi lossless, dimana citra

yang dikompres dengan format PNG dapat dikembalikan sama persis dengan data citra asli sebelum dikompres.

2. Penggunaan tipe filter yang berbeda akan menghasilkan rasio kompresi yang berbeda. Dari hasil percobaan, filter Paeth secara umum memberikan rasio kompresi yang lebih tinggi dibandingkan tipe filter lainnya (none, sub, up, average).
3. Dukungan kedalaman warna pada format PNG maksimum 48 bit perpixel memberikan kemampuan untuk menyimpan citra fotografi secara lossless dengan rasio kompresi yang tinggi.

#### **DAFTAR PUSTAKA**

- G. Randers – Pehrson, et. Al., “Portable Network Graphic(PNG) Specification Version 1.2”, PNG Development Group, 1999
- Hadi R, Pemrograman Windows API dengan Microsoft Visual Basic, PT Elex Media Komputindo, Jakarta
- Huffman, David A., “Method For The Construction Of Minimum Redundancy Codes”
- Roelofs, Greg, “PNG The Definitive Guide”, Greg Roelofs