

STUDI BANDING PENGGUNAAN FORMAT DjVu DAN FORMAT JPG UNTUK PENYIMPANAN DOKUMEN DALAM BENTUK ELEKTRONIK

Petrus Santoso

Fakultas Teknik, Jurusan Teknik Elektro – Universitas Kristen Petra
e-mail: petrus@petra.ac.id

ABSTRAK : Di masa ini sebagian besar informasi masih berupa kertas, belum dalam bentuk elektronik. Untuk memindahkannya ke dalam bentuk elektronik, diperlukan cara penyimpanan yang efisien. Untuk saat ini ada teknologi penyimpanan dokumen yang baru dikembangkan oleh AT & T Labs., yaitu DjVu.

Dalam makalah ini, dicoba melakukan perbandingan antara format DjVu dan JPG. Dengan melakukan perbandingan, diharapkan bisa menentukan, apakah format DjVu ini cukup bagus dan nantinya bisa menjadi standar penyimpanan yang baru.

Untuk itu dilakukan percobaan untuk menyimpan dokumen dalam format DjVu dan JPG. Hasilnya kemudian dibandingkan, baik dari segi ukuran maupun kualitasnya. Perbandingan dilakukan untuk 3 jenis dokumen: photo hitam putih, dokumen berisikan teks, dan dokumen berwarna.

Sebagai hasil akhir, disimpulkan bahwa format DjVu sangat efisien untuk menyimpan dokumen (file yang dihasilkan kecil). Kualitas gambar yang dihasilkan juga bagus. Dengan demikian mempunyai kesempatan besar untuk menjadi standar.

Kata kunci: DjVu, kompresi gambar, format file

ABSTRACT : *Nowadays, most of the information is still in printed format. To reproduce it in the electronic form, we need an efficient way to save the scanned document. For this purpose, AT & T Labs has been developing a new technology called DjVu.*

In this paper, we try to compare DjVu and JPG format. The goal is to decide whether DjVu is better or not.

To achieve the goal, we try to scan the document and save it in DjVu and JPG format, make comparison between these two based on file size and image quality. Comparison is done for three types of document, such as b/w photo, document with text, and document in colour.

The conclusion is, DjVu format is better than JPG in file size and quite good in image quality. So, DjVu have a chance to become a new standart.

Keywords: DjVu, image compression, file format

1. PENDAHULUAN

Pada masa sekarang ini Internet sudah merambah berbagai pelosok dunia. Tapi sebagian besar dokumen yang ada di dunia masih dalam bentuk kertas (*hard copy*). Banyak sekali dokumen-dokumen tersebut dalam bentuk gambar, foto atau bentuk lain yang mempunyai nilai penting untuk disimpan.

Untuk menyimpan dokumen-dokumen yang ada dalam bentuk elektronik tentunya membutuhkan usaha yang besar sekali. Salah satu usaha yang sudah dijalankan dan hasilnya bisa diakses di Internet adalah

Project Gutenberg. Proyek ini mempunyai tujuan sebagai berikut:

The Goal of Project Gutenberg is to Give Away One Trillion Etext Files by the December 31, 2001. [10,000 x 100,000,000 = Trillion]¹

Tapi proyek ini hanya menghasilkan file teks biasa. Untuk menghasilkan dokumen lain seperti gambar ataupun foto, tentunya kita harus men-*scan* gambar tersebut dan menyimpannya ke dalam file.

¹Berbagai judul *Etext* yang dihasilkan *Project Gutenberg*

Di sini muncul masalah yang kompleks. Di satu sisi, agar mudah diakses di Internet, file harus berukuran kecil. Di sisi lain agar informasi cukup akurat (kualitas gambar tetap bagus), dibutuhkan resolusi tinggi, yang tentunya menghasilkan file dengan ukuran besar.

Format file gambar yang menjadi standar saat ini adalah format GIF dan JPG. Tapi sekarang muncul teknologi baru yang dikembangkan oleh para ilmuwan di AT & T Labs. Yaitu penggunaan format DjVu [2,3]. Diharapkan teknologi ini akan bisa menggantikan standar yang sekarang sudah ada. Untuk mengembangkan teknologi ini AT & T Labs. mengambil kebijaksanaan *open source* (teknologi yang dipakai tidak dirahasiakan, tapi diberikan untuk dikembangkan).

Dalam makalah ini akan dicoba untuk membandingkan format DjVu dengan format JPG yang saat ini sudah menjadi standar. Studi banding ini bertujuan untuk melihat seberapa jauh kemampuan format DjVu melakukan kompresi dibandingkan dengan JPG. Juga akan dibandingkan mutu tampilan dari format DjVu dengan format JPG. Untuk keperluan ini akan dibandingkan tiga jenis dokumen:

- Photo dengan warna yang mempunyai dua level (hitam putih).
- Halaman dokumen yang berisikan teks.
- Gambar/Halaman dokumen yang berwarna.

Perbandingan tersebut juga akan dilakukan untuk berbagai macam resolusi.

Untuk melakukan studi banding ini akan dipelajari format file DjVu. Berikutnya akan dikembangkan program yang dipakai untuk melakukan konversi ke format DjVu. Untuk keperluan ini digunakan *developer library* yang diberikan oleh AT & T Labs. Berikutnya akan dilakukan pengujian mengenai ukuran file dan kualitas gambar yang dihasilkan.

Dari hasil studi banding ini diharapkan bisa ditarik kesimpulan mengenai kemampuan dari format DjVu dalam memenuhi tuntutan kebutuhan yang ada sekarang ini.

2. FORMAT FILE

Dalam bagian ini akan diuraikan format DjVu yang saat ini sedang dikembangkan di AT & T Labs. Ada 3 jenis format yang saat ini ada:

- Photo DjVu
- Bilevel DjVu
- Compound DjVu

Format Umum

File dengan format DjVu mempunyai struktur yang sesuai dengan tata letak standar "EA IFF 85"[1]. Tata letak ini dikembangkan oleh Electronic Arts. Informasi lebih lengkap tentang format standar ini bisa dilihat di http://www.cica.indiana.edu/graphics/image_specs/ilbm.format.txt.

Dalam hal ini, file tersusun atas sederetan *data chunk*. Tiap *chunk* diidentifikasi dengan identifier *chunk* sebanyak 4 karakter. Identifier dipakai untuk mengenali jenis data apa yang disimpan dalam *chunk* tersebut.

Dalam struktur IFF ada identifier khusus yaitu "FORM" yang dipakai untuk menghasilkan *chunk* komposit yang berisikan sederetan *data chunk* lagi. Dengan cara ini dihasilkan struktur yang hirarkis.

Chunk komposit tersebut dikenali dengan identifier sekunder. Untuk format DjVu diberikan identifier seperti "FORM:DJVU".

Sebelumnya, di awal dari file DjVu diberikan 4 buah oktet "AT&T". Oktet ini bukan bagian dari format IFF dan juga bukan komponen yang diperlukan dari sebuah file DjVu. Oktet ini dipakai karena versi tertentu MSIE mengenali file DjVu sebagai file suara Microsoft AIFF.

Photo DjVu

Jenis format ini paling tepat dipakai untuk gambar/foto berwarna ataupun *grayscale*.

File dengan format ini disusun oleh sebuah *chunk* komposit "FORM:DJVU". *Chunk* ini dimulai dengan satu *chunk* "INFO" berisikan ukuran gambar dan resolusi. Berikutnya diikuti oleh satu atau lebih *chunk* "BG44" yang berisikan data yang sudah di-*encode* dengan representasi IW44.

Bilevel DjVu

Jenis format ini biasa dipakai untuk melakukan kompresi gambar hitam putih yang menyatakan teks ataupun bentuk-bentuk geometris sederhana.

Untuk melakukan kompresi digunakan model JB2 [4]. *Encoding* dilakukan dengan menyatakan bagaimana karakter yang di-*encode* berbeda dengan karakter terpilih yang sudah di-*encode*.

File dengan format ini disusun oleh sebuah *chunk* komposit "FORM:DJVU". *Chunk* ini dimulai dengan satu *chunk* "INFO" berisikan ukuran gambar dan resolusi. Berikutnya adalah *chunk* "Sjbz" yang berisikan data bilevel yang dinyatakan dengan representasi JB2.

Compound DjVu

Compound DjVu merupakan cara yang sangat efisien untuk melakukan kompresi data resolusi tinggi untuk dokumen yang berisikan gambar maupun teks, misalnya koran.

Dokumen yang disimpan dengan format ini menggunakan dua layer; layer background dan layer foreground. Layer background dipakai untuk menyatakan gambar dan tekstur kertas. Layer foreround dipakai untuk menyatakan teks dan bentuk geometri sederhana.

File dengan format ini disusun oleh sebuah *chunk* komposit "FORM:DJVU". *Chunk* ini dimulai dengan satu *chunk* "INFO" berisikan ukuran gambar dan resolusi.

Bagian berikut adalah menyimpan layer foreground (dinamakan *foreground mask*, bisa dilihat pada gambar 1) disimpan dalam *chunk* "Sjbz" menggunakan representasi JB2.

Layer ini juga berisikan gambar berwarna yang kecil disimpan dalam *chunk* "FG44" menggunakan representas IW44.

Layer background berisikan gambar berwarna berupa satu atau lebih *chunk* "BG44" yang dinyatakan dalam representasi IW44. Contoh layer background bisa dilihat pada gambar 2.



Gambar 1. Foreground Mask⁵



Gambar 2. Layer Background⁵

Gambar dihasilkan dengan cara melakukan proses render. Layer background dicat dengan warna dari *chunk* "FG44". Foreground mask dalam hal ini dipergunakan sebagai stensil.

Chunk "FG44" dan *Chunk* "BG44" ukurannya diperkecil dengan melakukan sub-sampling. Faktor sub-sampling yang dipakai antara 1 sampai 12.

Contoh struktur file dari hasil kompresi bisa dilihat di bawah ini:

```
DJVU Image (2324x3090) version 17:
0.0 Kb 'INFO' Page information.
66.9 Kb 'Sjbz' JB2 foreground mask (2324x3090)
10.6 Kb 'BG44' IW44 background (775x1030)
12.3 Kb 'FG44' IW44 foreground colors (194x258)
```

16.2 Kb 'BG44' IW44 background (part 2).
 11.2 Kb 'BG44' IW44 background (part 3).
 35.8 Kb 'BG44' IW44 background (part 4).

3. PROSES ENCODING

Untuk melakukan proses *encoding*, pertama ditentukan dulu jenis file DjVu yang akan digunakan. Dalam hal ini, sebagian besar dokumen memberikan hasil yang bagus sebagai file Compound DjVu.

Langkah kedua adalah melakukan kompresi berdasarkan format yang sesuai. Khusus untuk Compound DjVu dilakukan pemisahan layer terlebih dahulu. Ini dilakukan baik dengan bantuan PhotoShop ataupun program tersendiri.

Algoritma untuk menuliskan file DjVu bisa diringkaskan sebagai berikut:

1. Mempersiapkan stream
2. Menulis header AT&T diikuti *chunk* "FORM:DJVU"
3. Menulis *chunk* "INFO" berdasarkan format file asal atau dispesifikasikan melalui parameter.
4. Untuk format Photo DjVu dilakukan penulisan *chunk* "BG44"
5. Untuk format Bilevel DjVu dilakukan penulisan *chunk* "Sjbz"
6. Untuk format Compound DjVu dilakukan penulisan *chunk* "Sjbz", *chunk* "FG44" dan *chunk* "BG44"
7. Langkah terakhir adalah menutup stream

Contoh kode program sebagai berikut:

Untuk membuka stream

```
StdioByteStream obs (argv[1],
"wb");
IFFByteStream iff(obs);
```

Menghasilkan header

```
iff.put_chunk("FORM:DJVU", 1);
```

Menghasilkan *chunk* "INFO"

```
create_info_chunk(iff, argv,
argv);
```

Menghasilkan *chunk* "Sjbz"

```
create_jb2_chunk(iff, "Sjbz",
argv[i]+5);
```

Menghasilkan *chunk* "FG44"

```
create_fg44_chunk(iff, "FG44",
argv[i]+5);
```

Menghasilkan *chunk* "BG44"

```
create_bg44_chunk(iff, "BG44",
argv[i]+5);
```

Menutup stream

```
iff.close_chunk();
```

4. PROSES DECODING

Dalam contoh di atas didefinisikan instance *img* yang merupakan class *DjVuImage*. Untuk melakukan proses *decoding* file DjVu, dilakukan dua tahap/ stage.

Stage pertama, disebut "*decoding*". Dalam stage ini, file dianalisa dan sebagian gambar dibentuk dalam memori. Stage ini memerlukan memori sekitar 1-2 Mbyte. Sedangkan file hasil scan/raw memerlukan sekitar 20-30 Mbyte (Asumsi resolusi 300 dpi).

Stage kedua, disebut "*rendering*". Stage ini menampilkan gambar pixel demi pixel dalam resolusi yang diinginkan.

Contoh kode program untuk stage pertama adalah sebagai berikut:

```
DjVuImage img;
ByteStream &bs = gambar_stream();
img.decode(bs);
```

Untuk aliran data digunakan class *ByteStream*.

Contoh kode program untuk stage kedua adalah sebagai berikut:

```
GRect part = tampil_part_redisplay();
GRect full = tampil_full_image();
GP<GPixmap>pm=
img->get_pixmap(part, full);
if (pm) { tampil_color(pm); return; }
GP<GBitmap>bm=
img->get_bitmap(part, full);
if (bm) { tampil_grays(bm); return; }
```

Ada 2 instance dalam objek *GRect* di atas, *part* dipakai untuk menyimpan data yang sedang dihitung, *full* dipakai untuk menyimpan data sebenarnya (dalam ukuran penuh). Dua baris di tengah melakukan *rendering*

untuk gambar berwarna. Dua baris terakhir melakukan *rendering* untuk gambar *grayscale*.

Tampilan secara progresif juga dimungkinkan dengan menggunakan *multi-threading*. Untuk itu digunakan class *GThread*.

5. HASIL PENGUJIAN

Dalam pengujian yang dilakukan, digunakan tiga jenis dokumen :

- Photo dengan warna yang mempunyai dua level (hitam putih).
- Halaman dokumen yang berisikan teks .
- Gambar/Halaman dokumen yang berwarna (contoh diambil dari buku TinTin: Tawanan Dewa Matahari, terbitan Indira).

Sedangkan resolusi yang dipakai juga ada beberapa macam. Untuk photo digunakan resolusi 72 dpi, 300 dpi dan 600 dpi. Untuk dokumen yang berisikan teks digunakan resolusi 72 dpi, 300 dpi dan 600 dpi. Untuk dokumen yang berwarna banyak digunakan resolusi 72 dpi, 100 dpi dan 300 dpi.

Dokumen yang ada dihasilkan dengan scanner HP 4c, disimpan dalam format TIFF.

Format JPG didapatkan dengan menggunakan Adobe Photoshop 5. Penyimpanan dilakukan dengan menggunakan kualitas medium (3).

Kompresi dilakukan pada mesin Linux (P166MMX) dengan memori 16MB. Sedangkan hasilnya ditampilkan pada mesin Windows 98 (P233MMX) dengan memory 64MB.

Pengujian dilakukan untuk 2 hal. Pertama ukuran gambar hasil kompresi. Kedua kualitas gambar dalam ukuran sebenarnya maupun diperbesar.

Hasil pengujian untuk photo hitam putih bisa dilihat pada tabel 1, 2 dan 3 berikut.

Tabel 1. Hasil Pengujian Photo Hitam Putih Resolusi 72 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 88 kByte |
| JPG | 44 kByte |
| DjVu | 20 kByte |

Tabel 2. Hasil Pengujian Photo Hitam Putih Resolusi 300 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 1478 kByte |
| JPG | 526 kByte |
| DjVu | 31 kByte |

Tabel 3. Hasil Pengujian Photo Hitam Putih Resolusi 600 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 5327 kByte |
| JPG | 1733 kByte |
| DjVu | 231 kByte |

Dari hasil di atas terlihat perbandingan antara file JPG dan DjVu berbeda cukup jauh. Terutama untuk file dengan resolusi tinggi.

Hasil pengujian untuk dokumen yang berisikan teks bisa dilihat pada tabel 4, 5 dan 6 berikut:

Tabel 4. Hasil Pengujian Dokumen Teks Resolusi 72 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 30 kByte |
| JPG | 213 kByte |
| DjVu | 16 kByte |

Tabel 5. Hasil Pengujian Dokumen Teks Resolusi 300 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 335 kByte |
| JPG | 1499 kByte |
| DjVu | 26 kByte |

Tabel 6. Hasil Pengujian Dokumen Teks Resolusi 600 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 1144 kByte |
| JPG | 3804 kByte |
| DjVu | 30 kByte |

Di sini juga didapatkan perbandingan yang mencolok antara file JPG dan DjVu. Dari hasil ini juga terlihat kelemahan file JPG jika dipakai untuk menyimpan dokumen yang berisikan teks. File yang dihasilkan menjadi lebih besar jika dibandingkan dengan file TIF.

Hasil pengujian untuk dokumen berwarna bisa dilihat pada tabel 7, 8 dan 9 berikut:

Tabel 7. Hasil Pengujian Dokumen Berwarna Resolusi 72 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 1417 kByte |
| JPG | 163 kByte |
| DjVu | 20 kByte |

Tabel 8. Hasil Pengujian Dokumen Berwarna Resolusi 100 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 2604 kByte |
| JPG | 280 kByte |
| DjVu | 31 kByte |

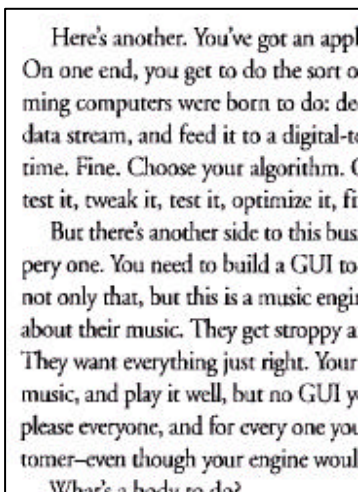
Tabel 9. Hasil Pengujian Dokumen Berwarna Resolusi 300 dpi

| Format File | Ukuran File |
|-------------|-------------|
| TIFF | 24357 kByte |
| JPG | 1737 kByte |
| DjVu | 154 kByte |

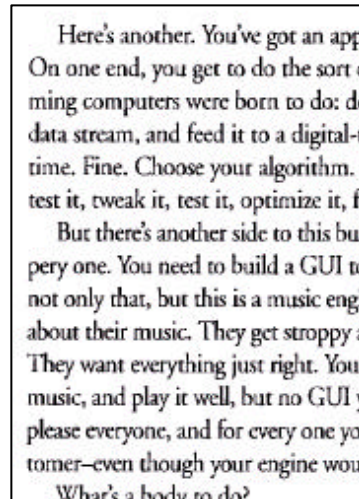
Hasil yang didapatkan sekali lagi menunjukkan bahwa format DjVu lebih unggul dibandingkan dengan JPG. Terlihat dari tabel terakhir, file DjVu 150 kali lebih kecil dibandingkan dengan file TIFF.

Untuk hasil pengujian kualitas gambar didapatkan hasil yang cukup mengesankan. Gambar dilihat sampai dengan kondisi zoom 300 %, dan masih menunjukkan kualitas yang baik.

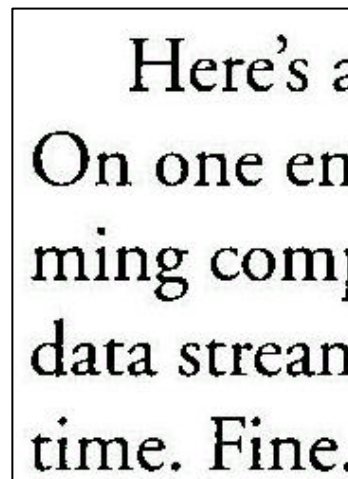
Contoh tampilan bisa dilihat pada gambar-gambar 3,5,7 dan 9 untuk file JPG dan gambar-gambar 4,6,8 dan 10 untuk file DjVu.



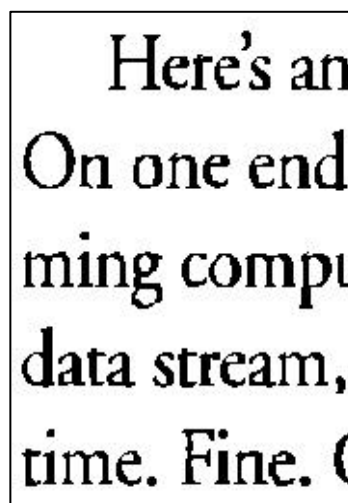
Gambar 3. Dokumen Berisi Teks⁵, Format JPG, Zoom 100%



Gambar 4. Dokumen Berisi Teks⁵, Format DjVu, Zoom 100%



Gambar 5. Dokumen Berisi Teks⁵, Format JPG, Zoom 300%



Gambar 6. Dokumen Berisi Teks⁵, Format DjVu, Zoom 300%



Gambar 7. Dokumen Berwarna⁵, Format JPG, Zoom 100%



Gambar 10. Dokumen Berwarna⁵, Format DjVu, Zoom 300%

Terlihat dari gambar-gambar di atas, kualitas format DjVu terlihat cukup bagus dibandingkan dengan format JPG.



Gambar 8. Dokumen Berwarna⁵, Format DjVu, Zoom 100%



Gambar 9. Dokumen Berwarna⁵, Format JPG, Zoom 300%

6. KESIMPULAN DAN SARAN

Kesimpulan

Dari studi yang sudah dilakukan akhirnya dapat ditarik kesimpulan-kesimpulan sebagai berikut:

1. Format DjVu sangat cocok untuk dipakai menyimpan dokumen yang direproduksi dalam bentuk elektronik. Terutama jika diinginkan resolusi tinggi.
2. Dengan ukuran file yang kecil, otomatis kecepatan menampilkan gambar juga lebih cepat dibandingkan dengan file lain.
3. Format DjVu di masa mendatang mempunyai kesempatan besar untuk menjadi standar.

Saran

Untuk pengembangan lebih lanjut bisa diberikan fasilitas-fasilitas tambahan dalam format DjVu. Antara lain fasilitas *hyperlink* dan *multipage*. Sehingga bisa jadi format ini bisa menjadi alternatif dari format PDF.

DAFTAR PUSTAKA

1. http://www.cica.indiana.edu/graphics/image_specs/ilbm.format.txt

2. <http://www.djvu.att.com/open>
3. <http://djvu.research.att.com/>
4. <http://www.jpeg.org/public/jbighomepage.htm>
5. _____, *TinTin: Tawanan Dewa Matahari*, Penerbit Indira, 1986.