



Penerapan Algoritma VLBE Pada Aplikasi Kompresi File

Binaria Tarigan¹

¹Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia

Email: binariatarigan1995@gmail.com

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi : 13 Mei 2020

Revisi Akhir : 23 Mei 2020

Diterima : 30 Mei 2020

Diterbitkan Online : 08 Juli 2020

KATA KUNCI

Kompresi,
File,
PDF,
Aplikasi,
Algoritma VLBE

KORESPONDENSI

E-mail: binariatarigan1995@gmail.com

ABSTRACT

Data atau file mempunyai peranan sangat penting bagi setiap orang yang setiap harinya menggunakan computer . Pengiriman data atau file melalui jaringan ataupun media hanya lainya ukuran yang besar, sehingga memakan waktu yang lama juga memerlukan ruang penyimpanan yang besar dalam prosesnya. Pengiriman file juga memerlukan kepastian agar file yang dikirim tidak dapat dicuri atau diubah oleh orang yang tidak bertanggung jawab, sehingga diperlukan cara untuk mengatasi masalah-masalah tersebut. Teknik kompresi merupakan salah satu teknik mengurangi besar kapasitas sebuah file yang terdapat digunakan penyediaan file agar dapat mengamankan sebuah file. Proses pemampatan dan penyediaan file PDF dilakukan dengan menggunakan mengkompres data terlebih dahulu dengan algoritma Variabel Length Binary Encoding kemudian aplikasi yang dirancang akan melakukan proses-proses tersebut sehingga akan menghasilkan file yang terkompres dan disandikan.

1. PENDAHULUAN

Perkembangan *File* mempunyai peran yang sangat penting bagi setiap orang ,dan dalam dunia teknologi informasi dan komunikasi telah berkembang pesat seiring kebutuhan masyarakat dalam memperoleh informasi secara cepat, selain itu memiliki potensi dalam menyaring data dan mengolahnya menjadi sebuah informasi, teknologi juga mampu menyimpan dengan jumlah kapasitas jauh lebih banyak, berbagai macam fasilitas teknologi telah dikembangkan agar dapat membantu masyarakat melakukan banyak pertukaran informasi dalam menentukan gambar, audio, video dan teks dan lain sebagainya, dengan kapasitas yang lebih baik. Bagi masyarakat informasi berupa teks itu adalah hal yang sangat penting maka dari itu keberadaan *file* berekstensi PDF mungkin tidaklah asing bahkan banyak yang mengetahui *software* dengan fungsi PDF reader seperti *Adobe Reader*, mengingat untuk pengolahan data sudah ada *Microsoft Office*.

PDF singkatan dari *Portable Document Format*, yaitu *file* yang dibuat oleh perusahaan.Adobe pada tahun 1993 untuk pertukaran dokumen.Seiring dengan berlalu waktu, dilakukan peningkatan versi PDF.Hingga saat ini, versi PDF terakhir adalah 1.7 yang dirilis pada tahun 2009 sesuai standar ISO 32000

[1]dalam pengolahan data memiliki ukuran atau kapasitas besar sehingga ekstensi tersebut di kompresi menjadi lebih kecil agar penyimpanannya lebih sedikit. Ukuran *file* PDF yang terlalu besar selalu menjadi masalah pada penyimpanan memori. Dengan munculnya masalah ini maka data tersebut dapat diatasi dengan cara kompresi *file* PDF yang asli menjadi lebih kecil. *File* yang telah di kompresi dapat dikembalikan ke data asli. Sehingga sebuah penyimpanan dapat menyimpan lebih banyak data dan proses pengiriman data pun semakin berjalan dengan cepat setelah data tersebut di kompresi

Kompresi bertujuan untuk mengurangi ukuran data menjadi lebih kecil dari ukuran aslinya. Teknik kompresi data ada dua yaitu [2].*Lossydata compression* dan *Lossless data compression*. *Lossydata compression* masih memperbolehkan adanya kesalahan dalam proses kompresi atau dekompresi, selama kesalahan proses tersebut tidak terlalu mengubah pola pokok dari data yang dikompres. Teknik kompresi ini biasanya digunakan untuk kompresi data gambar dan suara.*Lossless data compression* digunakan jika kesalahan tidak boleh terjadi sama sekali artinya data sebelum dikompres harus sama dengan data hasil dekompresi

File PDF dikompresi karena memiliki kapasitas besar dan algoritma yang digunakan untuk mengkompres *file* PDF adalah VLBE. Karena algoritma VLBE merupakan salah satu kompresi *file* PDF. VLBE merupakan algoritma kompresi data yang mengkodekan tiap karakter dengan mengurutkan karakter berdasarkan frekuensi dan disortir secara stabil sehingga membentuk beberapa rangkain bit. Algoritma VLBE memanfaatkan *frase-frase* yang berulang untuk melakukan kompresi. Untuk menentukan algoritma yang lebih optimal dan lebih efektif dalam kompresi teks, data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan,

Dari hasil penelitian sebelumnya yang dilakukan oleh Novelina Mancelina, Mesranaplikasi ini dirancang telah mampu melakukan kompresi string dengan menggunakan algoritma VLBE dan mampu melakukan dekompresi terhadap hasil kompresi menjadi string hasil[2].

Algoritma VLBE digunakan untuk mengkompresi *file* PDF adalah kode yang memetakan simbol sumber untuk sejumlah variable bit. Kode variable panjang dapat memungkinkan sumber yang akan dikompresi dengan nol kesalahan (kompresi data *lossless*) dan masi dapat dibaca kembali oleh simbol – simbol. Algoritma VLBE merupakan jenis kompresi *lossless* yang digunakan dalam penelitian ini, yang mana akan diukur dengan rasio kompresi (Ratio of compression dan *Compression Ratio*), *space saving*, dan waktu yang dan yang dibutuhkan selama proses kompresi[2].

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi merupakan proses perubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Saat ini terdapat berbagai tipe algoritma kompresi, antara lain Huffman, LZ77 dan variannya (LZ78,LZW,GZIP), Dynamic Markov Compression (DMC), Run Length, Shannon-Fano dan lain-lain. Kompresi data menjadi sangat penting karena dapat memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, dan memperkecil kebutuhan bandwidth[3].

2.2 Algoritma Variable Length Binary Encoding (VLBE)

Algoritma *Variable Length Binary Encoding* adalah algoritma kompresi data yang mengkodekan tiap karakter dengan mengurutkan karakter berdasarkan frekuensi dan disortir secara stabil sehingga membentuk beberapa rangkaian bit[2].

Adapun langkah-langkah dalam menyelesaikan proses kompresi dengan *Algoritma Variable Length Binary Encoding* adalah sebagai berikut:

1. Nilai awal biner dengan 0 sebagai root untuk pembentukan biner selanjutnya.
2. Nilai awal biner sebelumnya ditambahkan dengan angka 1, setiap angka didepan awal biner.
3. Jika biner n terakhir nilai awal biner diubah menjadi angka 1.

2.3 PDF (Portable Document Format)

PDF karena kebanyakan konten yang tersedia akan ditampilkan baik untuk format A4 atau surat, yang keduanya tidak mudah dibaca ketika diperkecil sesuai layar kecil. Beberapa aplikasi pembaca buku digital dapat menyusun ulang tampilnya beberapa dokumen PDF, termasuk sony PRS505, untuk mengamodasi layar kecil. Beberapa aplikasi yang ditawarkan antara lain; doPDF, PrimoPDF, PDF24 Creator, foxit Reader, atau menggunakan *Microsoft Office 2007* keatas. Bisa dilakukan konversi ke PDF secara online dan gratis. Untuk membuka *file* format PDF dibutuhkan aplikasi *foxit Reader* dan Web Browser. Format PDF merupakan format yang cukup banyak digunakan sebagai format digital yang mayoritas teks. Format ini memiliki kelebihan dalam hal format siap cetak. Bentuk format ini mirip dengan bentuk buku sebenarnya. Selain itu Format PDF memiliki fitur pencarian, daftar isi, bias memuat gambar pranalar luar, dan multimedia[1].

3. ANALISA DAN PEMBAHASAN

3.1 Analisa Masalah

Analisa adalah tahap aktifitas kreatif dimana analis berusaha memahami permasalahan secara mendalam terhadap metode yang diterapkan. ini adalah proses *iterative* yang terus berjalan hingga permasalahan dapat dipahami dengan benar. Analisis bertujuan untuk mendapatkan pemahaman metode yang diangkat secara keseluruhan tentang sistem yang akan dibuat berdasarkan masukan dari pihak- pihak dan juga pengalaman analis yang berkepentingan dengan sistem tersebut.

Sistem ini dirancang untuk melakukan proses kompresi pada *file* PDF. Proses kompresi menggunakan algoritma *Variable Length Binary Encoding* dengan tujuan untuk mengetahui hasil kinerja dari algoritma tersebut dalam mengkompresi sebuah *file* PDF. Dan sistem ini dirancang menggunakan bahasa pemograman *Microsoft Visual Basic 2008*.

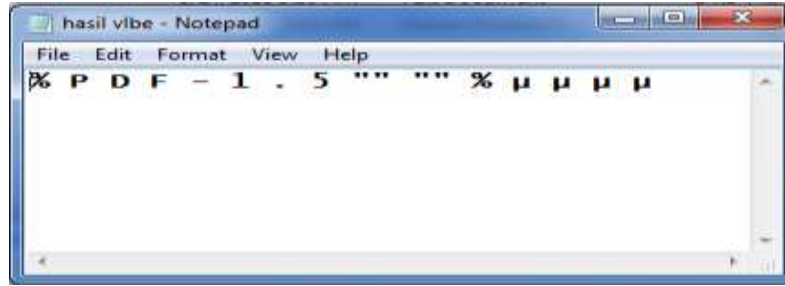
3.2 Penerapan Algoritma Variable Length Binary Encoding (VLBE)

Sebelum *file* dikompresi. Terlebih dahulu dilakukan pembacaan pada *file* PDF untuk mendapatkan data berupa biner. Membaca Biner yang terdapat pada *file* PDF menggunakan aplikasi Binary Viewer untuk mencari nilai biner pada aplikasi *file* PDF.

Tabel 1. Sempel *File* PDF yang akan dikompresi

| | |
|------------------|-----------------|
| Nama <i>File</i> | BINARIA TARIGAN |
| Type | <i>File</i> PDF |
| Size | 81,914Bytes |

Selanjutnya proses dekomposisi *Variabel Length Binary Encoding* yang akan dilakukan mengubah hasil seluruh biner menjadi karakter yang terdapat pada *File* awal.



Gambar 4. Hasil Dekomposisi

Selanjutnya proses Dekomposisi dilakukan dengan algoritma *Variabel Length Binary Encoding* yang dilakukan mengubah hasil seluruh biner menjadi string bit semula yang terdapat pada *File* awal

Tabel 6. Dekomposisi

| No | <i>Variabel Length Binary Encoding</i> | Nilai Biner |
|----|--|-------------|
| 1 | 0 | 00100101 |
| 2 | 10 | 01010000 |
| 3 | 110 | 01000100 |
| 4 | 1110 | 01000110 |
| 5 | 11110 | 00101101 |
| 6 | 111110 | 00110001 |
| 7 | 1111110 | 00101110 |
| 8 | 11111110 | 00110101 |
| 9 | 111111110 | 00001101 |
| 10 | 1111111110 | 00001010 |
| 11 | 11111111110 | 00100101 |
| 12 | 111111111110 | 00100101 |
| 13 | 1111111111110 | 10110101 |
| 14 | 11111111111110 | 10110101 |
| 15 | 111111111111111 | 10110101 |

Berdasarkan pada tabel dekomposisi maka diambil seluruh nilai bit digabungkan menjadi sebagai berikut.
 ” 0010010101010000010001000100011000101101001100010010111000110101
 0000110100001010001001001001011011010110110101”

Maka semua bit yang sudah terdekomposisi didapatkan hasil sampel awal biner viewer PDF sebagai berikut:

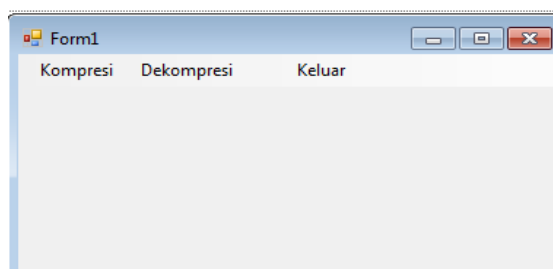
Tabel 7. Biner Viewer Dekomposisi

| | | | | |
|----------|----------|----------|----------|----------|
| 00100101 | 01010000 | 01000100 | 01000110 | 00101101 |
| 00110001 | 00101110 | 00110101 | 00001101 | 00001010 |
| 00100101 | 00100101 | 10110101 | 10110101 | 10110101 |

4. IMPLEMENTASI

4.1 Implementasi Sistem

Penerapan algoritma VLBE pada aplikasi kompresi *file pdf* ,telah dirancang dan dibuat menggunakan aplikasi *Microsoft Visual Studio 2008* dengan bahasa pemrograman *visual basic*. Adalah menu yang pertama kali muncul disaat aplikasi dijalankan. Menu utama memiliki 3 menu bar yaitu kompresi, dekomposisi dan keluar, berikut adalah tampilannya :



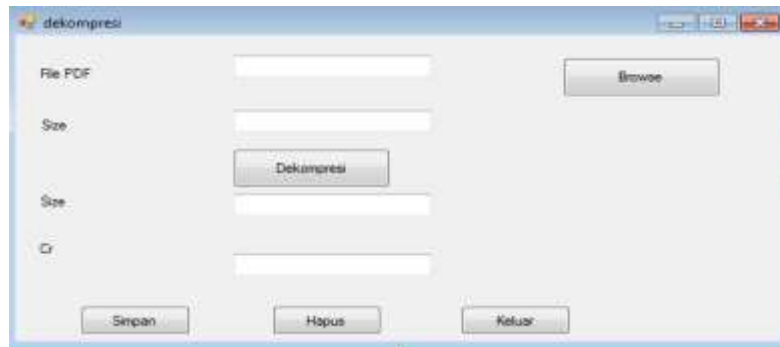
Gambar 5. Menu utama

Form ini digunakan untuk melakukan kompresi terhadap *file* yang dipilih pertama sekali membuka *file* yang akan dikompres dengan menggunakan tombol browse. Proses kompresi kemudian dimulai dengan menggunakan tombol kompresi hasil kompresi akan disimpan kembali dalam format *file* menggunakan tombol simpan.



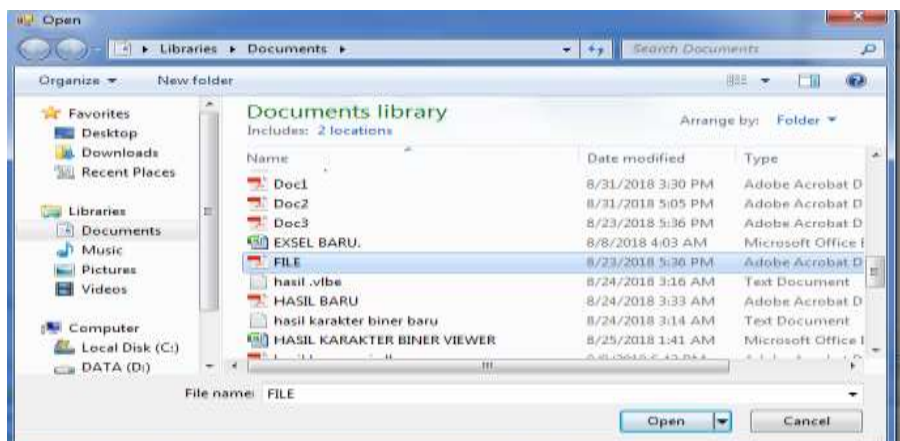
Gambar 6. Form Kompresi

Form ini digunakan untuk melakukan kompresi terhadap *file* yang dipilih pertama sekali membuka *file* yang akan dikompres dengan menggunakan tombol browse. Proses dekompresi kemudian dimulai dengan menggunakan tombol dekompresi hasil dekompresi akan disimpan kembali dalam format *file* menggunakan tombol simpan.



Gambar 7. Form Dekompresi

Tampilan output adalah tampilan proses ketika *file* pdf akan dikompres hingga menampilkan hasil kompresi serta proses dekompresi pdf. Adapun proses tersebut dilihat pada gambar di bawah ini:



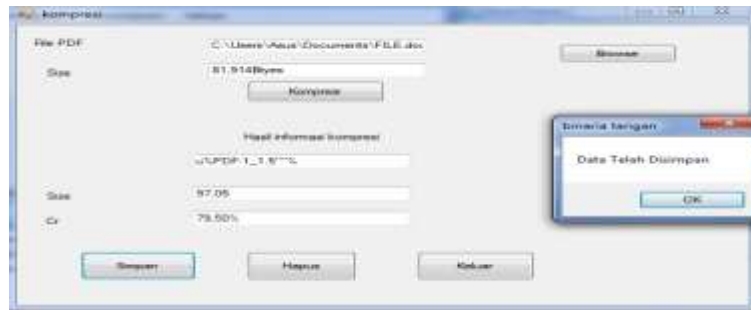
Gambar 8. Tampilan pemilihan *file* pdf

Berdasarkan pada gambar diatas,proses yang harus dilakukan ketika user melakukan kompresi pdf adalah memiliki tombol browse di menu aplikasi kompresi. Kemudian *file* yang dipilih pada gambar dibawah ini:



Gambar 9. Tampilan Proses Kompresi

Tampilan kompresi *file pdf* adalah *form* yang menampilkan hasil dari kompresi biner viewer berupa karakter. Adapun tampilan hasil kompresi dapat dilihat pada gambar dibawah ini.



Gambar 10. Tampilan Hasil Kompresi

Berdasarkan pada gambar diatas, untuk memulai proses kompresi dapat menekan tombol kompresi, hasil kompresi *file* sebelumnya mengalami perubahan pada ukuran.



Gambar 11. Tampilan Dekompresi

Berdasarkan pada gambar diatas, proses dekompresi dimulai dengan memilih tombol dekompresi. Hasil dekompresi adalah *file* awal saat sebelum proses dekompresi. Hasil *file* dekompresi dapat disimpan dengan memulai tombol simpan.

5. KESIMPULAN

Berdasarkan uraian dari bab-bab sebelumnya, maka penulis dapat memberikan kesimpulan sebagai berikut:

1. Proses mengkompresi *file pdf* telah berhasil melakukan kompresi *file pdf* dengan menggunakan algoritma VLBE, sehingga proses kompresi dapat berjalan sesuai yang diharapkan.
2. Aplikasi yang dirancang dapat melakukan Proses kompresi *file pdf*, sehingga *file pdf* yang memiliki ukuran yang besar dapat dikompresi menjadi ukuran yang lebih kecil dengan menggunakan algoritma VLBE dan melakukan dekompresi terhadap hasil kompresi.
3. Dengan menerapkan algoritma VLBE *file pdf* yang memiliki ukuran besar dapat diperkecil dengan menerapkan algoritma VLBE.

REFERENCES

- [1] I. Saefullah, MEMBUAT BUKU DIGITAL MANDIRI .
- [2] N. Mancelina, "PENERAPAN METODE PERBANDINGAN EKSPONENSIAL DALAM MENGANALISA KINERJA ALGORITMA FIXED LENGTH BINARY ENCODING DENGAN VARIABLE LENGTH BINARY ENCODING," vol. 2, no. 1, pp. 1–4, 2017.
- [3] A. Shannon, F. Menggunakan, J. S. Informasi, F. I. Komputer, and U. M. Buana, "Perancangan Aplikasi Kompresi SMS dengan," 2010.

- [4] S. s. M.K.D.T. Sutoyo, Teori pengolahan Citra Digital. CV ANDI OFFSET. 2009.
- [5] R.A.S and M.Shalahuddin, Rekayasa Perangkat Lunak. Informatika Bandung, 2015.
- [6] Hendrayudi, Dasar-Dasar Pemrograman Microsoft Visual Basic 2008. Bandung: PT. SARANA TUTORIAL NURANI SEJAHTERA, 2011.
- [7] R. Priyanto, Langsung Bisa VISUAL BASIC NET 2008. yogyakarta .ANDI OFFSET,2009.
- [8] U. Rusmawan ,komksi program VB.net untuk tugas akhir dan skripsi.jakarta PT ELEX media komputindo,2014.
- [9] B. J. Hutapea, M. A. Hasmi, and A. Karim, “Sistem Pendukung Keputusan Penentuan Jenis Kulit Terbaik Untuk Pembuatan Sepatu Dengan Menggunakan Metode Vikor,” *JURIKOM(Jurnal Ris. Komputer)*, vol. 5, no. 1, pp. 6–12, 2018.