



## Penerapan Algoritma VLBE Pada Aplikasi Penyimpanan File Secara Online

Sujanne Manalu

Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia  
Email: [sujanne12manalu@gmail.com](mailto:sujanne12manalu@gmail.com)

### INFORMASI ARTIKEL

#### Article History

Received : Mar 11, 2021  
Accepted : Jun 15, 2021  
Published : Jul 27, 2021

### KORESPONDENSI

Email: [sujanne12manalu@gmail.com](mailto:sujanne12manalu@gmail.com)

### A B S T R A K

Dalam perkembangan teknologi zaman sekarang ini, informasi bisa dengan cepat ditemukan melalui teknologi yang semakin pesat berkembangnya, seperti internet. Banyak masyarakat yang memiliki *file* atau data yang ingin disimpan kedalam aplikasi online. Namun besarnya suatu ukuran data mengakibatkan pemborosan dalam penggunaan media penyimpanan dan proses pengiriman data. Untuk menghemat ke dua hal itu diperlukan lah kompresi data. Kompresi data merupakan proses mereduksi ukuran suatu data untuk menghasilkan representasi digital yang padat atau mampat (*compact*) namun tetap dapat mewakili kuantitas informasi yang terkandung pada data tersebut. Ada begitu banyak algoritma untuk mengkompresi data yang ada, namun dari semua algoritma tersebut belum ada yang optimal dalam mengkompresi data teks. Dalam penelitian ini dilakukan untuk memperkecil suatu ukuran file sebelum diupload kedalam aplikasi yakni menggunakan algoritma VLBE (*Variable Length Binary Encoding*). 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*. Keuntungan algoritma VLBE ini adalah Kode variabel-panjang dapat memungkinkan sumber yang akan dikompresi dengan nol kesalahan dan dapat dibaca kembali oleh simbol-simbol salomon.

**Kata Kunci:** Kompresi; Deskripsi; Penyimpanan File; VLBE

### A B S T R A C T

In today's technological developments, information can be quickly found through increasingly rapidly developing technologies, such as the internet. Many people have files or data that they want to save into online applications. However, the large size of the data results in waste in the use of storage media and the process of sending data. To save those two things, data compression is needed. Data compression is the process of reducing the size of a data to produce a digital representation that is dense or compact but can still represent the quantity of information contained in the data. There are so many algorithms to compress existing data, but none of these algorithms is optimal in compressing text data. This research was conducted to reduce a file size before uploading it to the application using the VLBE (*Variable Length Binary Encoding*) algorithm. *Variable Length Binary Encoding Algorithm* is a data compression algorithm that encodes each character by sorting the characters by frequency and sorting stably to form a series of bits. The advantage of this VLBE algorithm is that the variable-length code can allow the source to be compressed with zero error and can be read back by Solomon symbols.

**Keywords:** Compression; Description; File Storage; VLBE

## 1. PENDAHULUAN

Perkembangan teknologi dan informasi di masyarakat telah berkembang pesat sesuai kebutuhan masyarakat dalam memperoleh informasi yang cepat. Berbagai macam fasilitas teknologi telah dikembangkan agar dapat membantu masyarakat dalam melakukan pertukaran informasi. Penyimpanan file secara *online* memudahkan transfer *file*, pesan, data, lampiran, catatan sebagainya. Masyarakat juga sangat suka menyimpan *file* secara *online* di internet seperti gambar maupun video dalam sebuah aplikasi misalkan pada aplikasi google drive, microsoft one drive.



Dalam menyimpan *file* secara online banyak kendala yang mungkin terjadi seperti ukuran atau *size* yang terlalu besar. Jika kita ingin mengupload suatu *file* kedalam aplikasi kita sebaiknya mengkompres datanya agar lebih banyak *file* yang dapat tersimpan. Jika kita ingin mendownloadnya kembali kita bisa mendekompresinya datanya agar kembali ke *file* semula. *Size* atau ukuran *file* yang kecil akan lebih mudah untuk menyimpannya. Kita sering gagal mengupload file karena masalah *size* yang tidak sesuai dengan memori yang kita miliki. Munculnya masalah ini maka data tersebut dapat diatasi dengan cara kompresi *file* yang asli menjadi lebih kecil. Sehingga sebuah aplikasi dapat menyimpan lebih banyak data dan proses pengiriman data pun semakin berjalan dengan cepat setelah data tersebut di kompresi.

Salah satu kegunaan kompresi adalah untuk memperkecil kapasitas kosong dalam memori media penyimpanan, agar kita tidak terlalu boros menggunakan media penyimpanan tersebut. Kompresi data berarti suatu teknik untuk memampatkan data agar diperoleh data dengan ukuran yang lebih kecil daripada ukuran aslinya sehingga lebih efisien dalam menyimpannya serta mempersingkat waktu pertukaran data tersebut[1].

Algoritma *Variable Length Binary Encoding* (VLBE) adalah jenis kompresi lossless yang digunakan dalam penelitian ini, akan diukur kinerja oleh *Compression Ratio* (CR), *Ratio of compression* (RC), Waktu kompresi (milidetik) dan waktu Dekompresi (milidetik). 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]. Algoritma VLBE digunakan untuk mengkompres data seperti text, video, gambar, audio dan lainnya.

Dari hasil penelitian sebelumnya yang dilakukan oleh D Viliana disimpulkan bahwa aplikasi yang dirancang telah mampu melakukan kompresi dalam waktu yang lebih sedikit untuk mengembalikan file teks hasil kompresi ke file teks semula, dengan rata-rata 0.153 milisekon[3].

## 2. METODOLOGI PENELITIAN

### 2.1 Penyimpanan Online

Saat ini banyak sekali media untuk menyimpan data antara lain, hardisk, plasdisk, CD, DVD, dan lainnya. Selain media penyimpanan fisik yang berbentuk, anda juga dapat menyimpan data pada media penyimpanan lain yang bersifat virtual yaitu internet. Salah satu keuntungan yang dapat diperoleh dari penyimpanan online di internet adalah kemudahan untuk diakses dari berbagai tempat.

### 2.2 Metode VLBE (*Variable Length Binary Encoding*)

Algoritma *Variable Length Binary Encoding* (VLBE) dalam teori pengkodean adalah kode yang memetakan simbol sumber untuk sejumlah variabel bit. Kode variabel-panjang dapat memungkinkan sumber yang akan dikompresi dengan nol kesalahan (kompresi data *lossless*) dan masih dapat dibaca kembali oleh simbol-simbol Salomon. 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* Salomon [4].

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

Cara untuk membangun kode *Variable Length Binary Encoding* (VLBE) adalah sebagai berikut:

1. Nilai awal biner dimulai dari 0 sebagai root untuk pembentukan biner selanjutnya.
2. Nilai awal biner sebelumnya ditambahkan dengan angka 1, setiap angka 1 diletakkan didepan nilai awal biner.
3. Jika biner *n* terakhir maka nilai awal biner diubah menjadi angka 1.
4. Kemudian jumlah bit dari karakter dikalikan dengan frekuensi kemunculan tersebut.
5. Nilai setiap hasil perkalian dari bit dan frekuensi dijumlahkan.

Kode *Variable Length Binary Encoding* (VLBE) pada tabel 1.

Tabel 1. kode *Variable Length Binary Encoding* (VLBE)

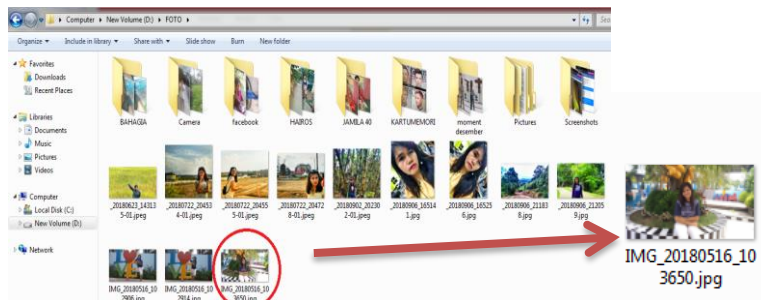
No	<i>Variable Length Binary Encoding</i>
1	0
2	10
3	110
4	1110
5	11110
6	111110

## 3. HASIL DAN PEMBAHASAN

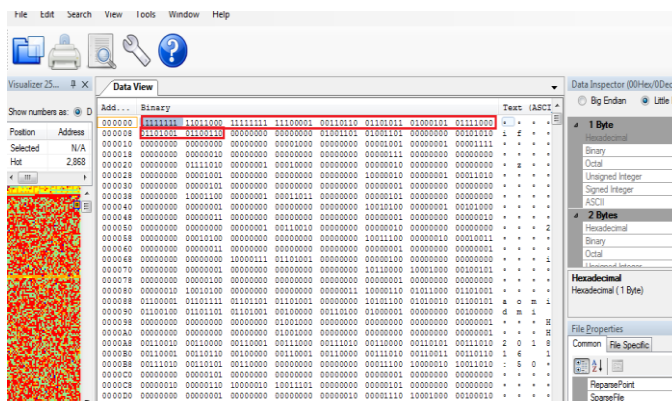
Dalam penyimpanan suatu *file* memiliki tentu harus melihat ukuran file sebelum mengupload. Dengan melakukan kompresi data, data yang berukuran besar akan dikompresi menjadi ukuran yang kecil dan akan mengurangi alokasi penyimpanan. Dalam penelitian ini, akan dibahas 2 proses utama yaitu proses kompresi dan dekompresi, dan peneliti akan mengkompresi sebuah penyimpanan *file* dengan algoritma VLBE (*Variable Length Binary Encoding*).

**a. Proses Kompresi**

Contoh *file* yang akan disimpan adalah *file* gambar yang akan dilakukan sebuah proses percobaan kompresi pada *file* gambar ini. Sebelum *file* dikompresi, terlebih dahulu dilakukan pembacaan pada *file* gambar untuk mendapatkan data berupa biner. Membaca biner yang terdapat pada *file* gambar menggunakan aplikasi *Binary Viewer* untuk mencari biner pada aplikasi *file* gambar. Berikut adalah contoh *file* gambar.



**Gambar 1.** *File* yang akan dikompres



**Gambar 2.** Kode Binary Dari Hasil *Binary Viewer*

Berdasarkan contoh diatas di dapat nilai biner viewer, adapun nilai binary pada file contoh D:\FOTO\IMG\_20180516\_103650 tersebut adalah:

**Tabel 2.** UrutanTabel Berdasarkan Binary Viewer

11111111	11011000	11111111	11100001	00110110
01101011	01000101	01111000	01101001	01100110

Berdasarkan pada tabel di atas nilai biner pada *file* adapun nilai binary pada file sempel D:\FOTO\IMG\_20180516\_103650 tersebut adalah:

111111111101100011111111110000100110110011010110100010101111000011010010110011

1. Buat daftar frekuensi kemunculan tiap-tiap simbol

Berdasarkan tabel diatas nilai awal biner diambil dari 0 sebagai pembentukan root selanjutnya dan setiap awal biner sebelumnya ditambahkan angka 1, setiap angka 1 diletakan awal biner. Jika biner n terakhir maka nilai awal biner dirubah menjadi angka 1.

**Tabel 3.** Pendataan Frekuensi dan Bit

No	Binary	Bit	Frekuensi	Frek x bit
1	11111111	8	2	16
2	11011000	8	1	8
3	11100001	8	1	8
4	00110110	8	1	8
5	01101011	8	1	8
6	01000101	8	1	8
7	01111000	8	1	8
8	01101001	8	1	8
9	01100110	8	1	8
Jumlah				80

Algoritma VLBE dalam teori pengkodean adalah kode yang memetakan simbol sumber untuk sejumlah variable bit. Kode variable panjang dapat memungkinkan sumber yang akan dikompresi dengan nol kesalahan dan masih dapat dibaca kembali oleh simbol-simbol. Algoritma VLBE adalah algoritma kompresi data yang mengkodekan tiap karakter dengan mengurutkan karakter berdasarkan frekuensi dan disortir secara stabil sehingga membentuk beberapa rangkaian bit.

**Tabel 4.** Kode *Variable Length Binary Encoding* (VLBE)

No	<i>Variable Length Binary Encoding</i>
3	110
4	1110
5	11110
6	111110

- Berdasarkan tabel diatas nilai awal biner diambil dari 0 sebagai pembentukan root selanjutnya dan setiap awal biner sebelumnya ditambahkan angka 1, setiap angka 1 diletakan awal biner. Jika biner n terakhir maka nilai awal biner dirubah menjadi angka 1.

**Tabel 5.** Kompresi Menggunakan Kode VLBE

No	Binary	Frekuensi	VLBE	Jumlah Bit	Frek x bit
1	11111111	2	0	1	2
2	11011000	1	10	2	1
3	11100001	1	110	3	3
4	00110110	1	1110	4	4
5	01101011	1	11110	5	5
6	01000101	1	111110	6	6
7	01111000	1	1111110	7	7
8	01101001	1	11111110	8	8
9	01100110	1	11111111	9	9
Jumlah					45

- Berdasarkan pada tabel menjadi nilai bit biner dari kode VLBE di atas dapat dibentuk nilai bit baru hasil dikompres dari susunan nilai biner pada *file* sampel awal sebelumnya awal sebelum kompresi yaitu:  
 "11111111 11011000 11100001 00110110 01101011 01000101 01111000 01101001 01100110"
- Selanjutnya Biner diatas disesuaikan dengan kode VLBE sebagai berikut:  
 01011011 10111101 11110111 11101111 11101111 1111
- Dari string bit diatas maka dapat disimpulkan seperti tabel dibawah ini

**Tabel 6.** Nilai Desimal dan *Codeword*

Codeword	Desimal
01011011	91
10111101	189
11110111	247
11101111	239
11101111	239
1111	15

Berdasarkan pada tabel di atas dapat dibentuk nilai bit baru hasil kompresi dari susunan nilai desimal sampel awal sebelum kompresi yaitu, 91, 189, 247, 239,239,15 (tanpa tanda koma dan spasi) menjadi nilai *bit* biner "0101101110111101111101111101111101111110111111111"

Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan *string bit* itu sendiri yaitu *padding* bit dan *flag* bit. Jika sisa bagi panjang *string* bit terhadap 8 adalah 0 maka tambahan 00000001. Nyatakan dengan bit akhir. Sedangkan jika sisa bagi panjang *string* bit terhadap 8 adalah n(1,2,3,4,5,6,7) maka tambahkan 0 sebanyak  $7 - n + "1"$  di akhir *string bit*. Nyatakan dengan L. Lalu tambahkan bilangan biner dari  $9 - n$ . nyatakan dengan bit akhir. karena jumlah string bit 44 tidak habis dibagi delapan dan sisanya 4 bit, nyatakan sisa bagi tersebut dengan nilai n. maka tambahkan 0 sebanyak  $7 - n + "1"$  di akhir string bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari  $9 - n$ . Nyatakan dengan bit akhir.

$$7 - n + "1"$$

$$7 - 4 + "1" = 0001$$

Bit Akhir  $9 - n$

$$\text{Bit Akhir} = 9 - 4 = 5 = 00010000$$

"0101101110111101111101111101111111000100010000". Total panjang bit keseluruhan setelah ada penambahan bit adalah  $44+4+8=56$ . Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti tabel di bawah ini.

**Tabel 7.** Pembagian Bit

01011011	10111101	11110111	11101111
11101111	11110001	00010000	

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 9 kelompok nilai biner baru yang sudah terkompresi beserta nilai biner penambahan bit. Setelah pembagian dilakukan, maka nilai yang sudah dibagi dirubah

kedalam suatu karakter dengan terlebih dahulu mencari nilai desimal dari *string bit* tersebut menggunakan kode ASCII untuk mengetahui nilai yang sudah terkompresi. Adapun nilai simbol yang sudah terkompresi dapat dilihat pada tabel di bawah ini:

**Tabel 8.** Hasil Biner, Desimal dan Simbol

Codeword	Nilai Desimal terkompresi	Karakter
01011011	91	[
10111101	189	]
11110111	247	≈
11101111	239	∩
11101111	239	∩
11110001	241	±
00010000	16	DLE

Dari hasil Kompresi *Variabel Length Binary Encoding* diatas dapat dihitung kinerja kompresinya menurut parameter yang sudah ditentukan yaitu:

*Compression Ratio* dari contoh menggunakan algoritma *Variabel Length Binary Encoding (VLBE)*

$$C_R = \frac{44}{80} * 100\% = 55\%$$

Adapun keseluruhan hasil kompresi dapat dilihat berikut:

[ ] ≈ ∩ ∩ ≡ DLE

Berdasarkan gambar diatas maka diambil seluruh biner dan di gabungkan menjadi:

“01011011101111011111011111101111111011111111000100010000”.

#### b. Proses dekompresi file dengan algoritma VLBE

Selanjutnya proses dekompresi hal yang dilakukan adalah menganalisa keseluruhan bit hasil dari kompresi sebelumnya. Adapun bit keseluruhan hasil kompresi dapat dilihat pada tabel berikut :

**Tabel 9.** Hasil Simbol, Desimal dan Codeword

Codeword	Nilai Desimal terkompresi	Karakter
01011011	91	[
10111101	189	]
11110111	247	≈
11101111	239	∩
11101111	239	∩
11110001	241	±
00010000	16	DLE

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi “01011011101111011111011111101111111011111111000100010000”.

Selanjutnya adalah dengan mengembalikan *binary* menjadi *string bit* semula dengan menghilangkan biner yang ditebalkan. Untuk mengembalikan *binary* menjadi *string bit* semula dapat dilakukan melalui langkah berikut ini. Lakukan pembacaan pada 8 bit terakhir, hasil pembacaan berupa bilangan desimal. Nyatakan hasil pembacaan dengan n. Hilangkan bit pada bagian akhir sebanyak 7+n. Setelah dilakukan perhitungan pembacaan bit akhir . Nilai biner yang dihilangkan sebanyak 8 bit pada akhir. n = 1. Hilangkan 7 + n atau 7+8 = 15. Penjelasan diatas menunjukan bahwa bit akhir harus dihilangkan. Hasil pengembalian *binary* menjadi *string bit* semula dapat dilihat sebagai berikut ini:

“01011011101111011111011111101111111” pada diatas berjumlah 44 bit seperti diawal sehingga dilakukan pembacaan *string bit* awal. Adapun tabel hasil perhitungan diatas adalah sebagai berikut:

**Tabel 10.** Pencocokan Bit

Indeks	Pengecekan bit	Keterangan
1	0	Tidak ada
2	01	Tidak ada
3	010	Tidak ada
4	0101	Tidak ada
5	01011	Tidak ada
8	01011011	Ada pada tabel
9	0	Tidak ada
10	10	Tidak ada
11	101	Tidak ada
12	1011	Tidak ada
13	10111	Tidak ada

Indeks	Pengecekan bit	Keterangan
14	101111	Tidak ada
15	1011110	Tidak ada
16	10111101	Ada pada tabel
17	11110111	Tidak ada
18	1	Tidak ada
19	11	Tidak ada
20	111	Tidak ada
21	1110	Tidak ada
22	11101	Tidak ada
23	111011	Tidak ada
24	1110111	Tidak ada
25	11101111	Ada pada tabel
26	1	Tidak ada
27	11	Tidak ada
28	111	Tidak ada
29	1110	Tidak ada
30	11101	Tidak ada
31	111011	Tidak ada
32	1110111	Tidak ada
33	11101111	Ada pada tabel
34	1	Tidak ada
35	11	Tidak ada
36	111	Tidak ada
37	1111	Ada pada tabel

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

[  $\approx \cap \cap \equiv$  DLE

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

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 11.** Dekomposisi

No	<i>Variabel Length Binary Encoding</i>	Nilai Biner
1	0	11111111
2	10	11011000
3	110	11111111
4	1110	11100001
5	11110	00110110
6	111110	01101011
7	1111110	01000101
8	11111110	01111000
9	111111110	01101001
10	111111111	01100110

### 3.1 Implementasi Program

Tampilan unggah dan kompresi *file* yaitu menu dokumen yang menampilkan hasil dari unggah dan kompresi *file*. Ada tampilan hasil kompresi gambar dapat dilihat pada gambar 8.



**Gambar 3.** Tampilan *File* Hasil Unggah dan Kompresi


### 3.3 Pengujian Aplikasi

Pengujian *file* gambar merupakan salah satu tujuan dari diterapkannya algoritma *Inverted Elias Delta* kedalam aplikasi penyimpanan *file* terkompresi. Berikut adalah hasil pengujian unggah dan kompresi *file* gambar yang sudah diterapkan kedalam aplikasi penyimpanan *file* terkompresi.

#### 1. Hasil Pengujian Unggah dan Kompresi

Adapun hasil pengujian *file* yang dilakukan dapat dilihat pada tabel 4.1 berikut:

**Tabel 12.** Hasil Pengujian kompresi *File* Gambar

No	Sebelum Kompresi		Sesudah Kompresi			
	<i>File</i>	Ukuran	Nama <i>File</i>	Ukuran	Nama <i>File</i>	Cr
1		1.92 MB	IMG_20180516_103650.jpg	680 KB	IMG_20180516_103650.jpg	55%

Berdasarkan pada hasil pengujian pada tabel 4.1, menunjukkan bahwa hasil kompresi *file* memiliki perbedaan ukuran ketika *file* sebelum di kompresi dan setelah dikompresi. Hal ini dapat dilihat dari *Compression ratio* (CR) yang memiliki nilai 55%.

#### 2. Hasil Pengujian Dekompresi *File*

Adapun hasil pengujian dekomposisi dari *file* yang sudah dilakukan kompresi dapat dilihat pada tabel di bawah ini:

**Tabel 13.** Hasil Pengujian Dekompresi *File* Gambar

No	Gambar Kompresi		Sesudah Dekompresi	
	Nama <i>File</i>	Ukuran	<i>File</i>	Ukuran
1	IMG_20180516_103650.jpg	680 KB		1.92B

Berdasarkan pada hasil pengujian pada tabel di atas, menunjukkan bahwa hasil dekomposisi mengembalikan ukuran gambar menjadi ukuran awal sebelum terjadinya kompresi.

## 4. KESIMPULAN

Kesimpulan yang diperoleh dari penelitian bahwa aplikasi yang dirancang telah mampu melakukan dengan menggunakan algoritma VLBE. Aplikasi dapat menghemat waktu dan lebih sedikit memori yang terpakai dalam mengupload suatu *file*. Aplikasi mampu melakukan dekomposisi atau pengembalian data ke bentuk aslinya atau kompresi.

## REFERENCES

- [1] D. Stmik, B. Sarana, M. Stmik, B. Sarana, and I. I. L. A. T. Eori, "Aplikasi Kompresi File Dengan Metode," vol. 4, no. 1, pp. 1–4, 2014.
- [2] V. Lusiana, "Teknik Kompresi Citra Digital untuk Penyimpanan File menggunakan Format Data XML," vol. 19, no. 2, pp. 112–119, 2014.
- [3] S. D. Nasution, "PERANCANGAN APLIKASI KOMPRESI FILE TEKS DENGAN MENERAPKAN," no. May 2016, pp. 1–4, 2017.
- [4] J. Emitter *et al.*, "Kompresi citra dengan menggunakan metode delta modulation," vol. 14, no. 1, pp. 1–12.
- [5] B. P. Silalahi, F. Bukhari, and S. Nurhudayani, "PENGKODEAN ARITMETIKA UNTUK KOMPRESI DATA TEKS," pp. 1–10, 1987.
- [6] C. T. Utari, P. Studi, M. Teknik, U. S. Utara, and K. Citra, "IMPLEMENTASI ALGORITMA RUN LENGTH ENCODING UNTUK PERANCANGAN APLIKASI KOMPRESI DAN DEKOMPRESI," vol. V, no. 2, pp. 24–31, 2016.
- [7] U. S. Utara and U. S. Utara, "Perbandingan Algoritma Kompresi Variabel Length Binary Encoding Dan Punctured Elias Codes P2 Untuk Kompresi Audio Wav," 2016.
- [8] Hengky Alexander Mangkulo, "PENYIMPANAN ONLINE GRATIS DI INTERNET", 2011.
- [9] Andi Publisher, "KUPAS TUNTAS ADOBE DREAMWEAVER CS5 DENGAN PEMROGRAMAN PHP & MYSQL ", 2011.