

Implementasi Algoritma Boldi Vigna Kompresi File Teks

Dini Novelia

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Jl. Sisingamangaraja No.338, Siti Rejo I, Kec. Medan Kota, Medan, Sumatera Utara, Indonesia

Email: noveliadini97@gmail.com

Email Penulis Korespondensi:-

Abstrak—Teknologi Kompresi data adalah proses Pengubahan sekumpulan data menjadi bentuk kode yang lebih rendah. Teknik kompresi biasanya digunakan untuk proses transmisi data, Kompresi data atau pemampatan data adalah sebuah cara dalam ilmu komputer untuk memerlukan ruangan penyimpanan lebih kecil, sehingga lebih efisien dalam penyimpanannya. Boldi Vigna merupakan contoh Algoritma untuk melakukan kompresi dimana keduanya melakukan kompresi data dengan cara melakukan descending sorting terhadap masing-masing karakter. Implementasi merupakan Penerapan atau pelaksanaan, sebagai tindakan untuk menjalankan rencana yang telah dibuat.

Kata Kunci: Implementasi; Boldi Vigna; File Teks.

Abstract—Data compression technology is the process of converting a set of data into a lower form of code. Compression techniques are usually used for data transmission processes, Data compression or data compression is a way in computer science to require smaller storage space, making it more efficient in storage. Boldi Vigna is an example of an algorithm for compression where both compress data by descending sorting each character. Implementation is the application or implementation, as an action to carry out the plans that have been made.

Keywords: Implementation; Boldi Vigna; Text File.

1. PENDAHULUAN

File merupakan tempat penyimpanan data dalam bentuk digital. Jumlah data yang disimpan pada file berbanding lurus dengan ukuran file itu sendiri. Ukuran file yang terlalu besar akan menjadi masalah bila file tersebut akan ditransfer atau dipertukarkan[1]. Teknik kompresi adalah teknik untuk mereduksi jumlah bit yang diperlukan untuk menyimpan data file citra sehingga ukuran file yang terkompresi lebih kecil dibanding file asal, tetapi tidak mengalami penurunan kualitas file yang signifikan. Dengan demikian maka kapasitas ruang penyimpanan yang diperlukan akan menjadi lebih kecil serta waktu dan biaya pengaksesan menjadi lebih kecil pula.

Kompresi data (pemampatan data) merupakan suatu teknik untuk memperkecil jumlah ukuran data (hasil kompresi) dari data aslinya. Pemampatan data umumnya diterapkan pada mesin komputer, hal ini dilakukan karena setiap simbol yang dimunculkan pada komputer memiliki nilai-nilai bit yang berbeda. Berkas digital (*file*) yang disimpan hasil dari proses kompresi biasanya akan berbentuk raw data, informasi pendukung mengenai berkas tersebut yang biasa dikenal dengan meta data ikut diproses juga. Informasi meta data ini menjadi hilang, hal ini disebabkan proses kompresi pada berkas akan dikerjakan mulai dari karakter awal data sampai dengan karakter terakhir data (*end of file*)[2].

Metode pengkompresian data telah berkembang dan masing-masing metode memiliki kelebihan dan kekurangan, kemampuan dari tiap-tiap metode pengkompresian data umumnya diukur dengan parameter ukuran atau kapasitas dan kecepatan kompresi, dengan kompresi atau mengecilkan ukuran file terlebih dahulu, maka kapasitas tempat penyimpanan yang diperlukan menjadi lebih kecil. Sisa dari bagian ini menjelaskan kode *zeta* (ζ), juga dikenal sebagai *Boldi-Vigna* kode, diperkenalkan oleh *Paolo Boldi dan Sebastiano Vigna* sebagai keluarga panjang variable kode yang merupakan pilihan terbaik untuk kompresi *webgraph*.

Referensi asli adalah *Boldi dan Vigna*. Kita mulai dengan hukum *Zipf*, hukum kekuatan empiris *Zipf* yang diperkenalkan oleh ahli bahasa *George K. Zipf*. Ini menyatakan bahwa frekuensi setiap kata dalam bahasa alami kurang lebih berbanding terbalik dengan posisinya di tabel frekuensi. Secara intuitif, hukum *Zipf* menyatakan bahwa kata yang paling sering dalam bahasa apa pun adalah dua kali lebih sering dari pada kata yang paling sering kedua, yang pada gilirannya dua kali lebih sering dari kata ketiga, dan seterusnya. Kode *zeta Boldi-Vigna* dimulai dengan bilangan bulat positif k yang menjadi penyusun faktor kode. Himpunan semua bilangan bulat positif dipartisi ke dalam interval $[2k - 1, 2k, 2k - 1]$, $[2k, 2k - 1]$, $[2k, 2k - 1]$, dan secara umum $[2hk, 2(h + 1)k - 1]$. Panjangnya setiap interval adalah $2(h + 1)k - 2hk$ [3].

Cara kerja sistem yang akan dirancang bangun dengan menggunakan file teks yang berukuran *byte* yang kecil dengan sistem kompresi file teks dengan menggunakan *algoritma boldi vigna* akan mengubah teks yang beradapada file tersebut menjadi sebuah bilangan angka positif yang berpartisi kedalam nilai nilai bilangan prima dengan ukuran *byte* yang kecil dan tidak membebani memori laptop dan hardisk pada komputer.

Alasan penulis memakai algoritma boldi vigna ini berkeinginan untuk melakukan penelitian tentang bagaimana melakukan enkripsi file teks mengamankan data berupa file teks yang berisikan data atau berita pribadi menjadi keamanan data yang tidak bisa dicuri oleh orang lain yang tidak bertanggung jawab dengan mengubah data asli menjadi sebuah bilangan angka primer positif dengan *algoritma boldi vigna*.

2. METODOLOGI PENELITIAN

2.1 File Teks

Dalam dunia komputer, data membawa informasi yang disimpan dalam media penyimpanan (*data storage*). Pada kasus tertentu, data bersifat penting dimana data-data lama pada data storage tidak boleh dihapus karena digunakan sebagai arsip. Disatu sisi, pertumbuhan data baru yang eksplosif perlu disimpan secara terus- menerus, selain cenderung mengurangi ruang kosong (*free space*) dalam data storage juga akan memperlambat proses pemindahan data jika diperlukan. Salah satu jenis data yang digunakan untuk menyimpan informasi penting adalah data teks.

Sebuah teks pada umumnya memiliki banyak pengulangan kata terutama pada penggunaan kata penghubung (dengan, dan, jika, dsb). Selain itu pengulangan kata juga terjadi karena sebuah teks cenderung membahas suatu topik tertentu sehingga kata-kata yang berhubungan dengan topik tersebut seringkali digunakan berulang- ulang. Dalam sistem komputer, jumlah karakter dalam file teks berbanding lurus dengan ukuran file. Setiap satu huruf berformat *ASCII* disimpan dalam 8 bit memori (1 *byte*)[1].

Selain itu pendapat ahli menyatakan File teks merupakan file yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca[2].

Kesimpulan dari penulis cara kerja sistem yang akan dirancang bangun dengan menggunakan file teks yang berukuran byte yang kecil dengan system kompresi file tekstur diri dari karakter huruf, angka, symbol dan data pada file teks berupa data document yang memiliki kapasitas tergantung pada isi file dokument tersebut.

2.2 Kompresi

Kompresi adalah pengubahan data yang berupa kumpulan karakter menjadi bentuk kode dengan tujuan untuk menghemat kebutuhan tempat penyimpanan dan waktu transmisi data. Ada beberapa faktor yang dijadikan sebagai bahan pertimbangan dalam memilih algoritma yang akan digunakan dalam kompresi data yaitu sumber daya yang dibutuhkan (memory, kecepatan *PC*), kecepatan kompresi, ukuran hasil kompresi, besarnya reduksi dan kompleksitas algoritma. Jika semakin besar ukuran data yang disimpan maka membutuhkan media penyimpanan yang besar. Oleh karena itu, kemudian muncul metode-metode yang bertujuan untuk mengkompresi data agar dapat menghemat tempat penyimpanan data. Salah satunya yaitu kompresi data teks agar bisa diperkecil ukurannya.

Metode kompresi data teks ini dapat dikelompokkan ke dalam dua kelompok besar, yaitu *Dictionary based* yang bekerja dengan mengganti kelompok simbol dalam data menjadi kode-kode dengan panjang tertentu, dengan asumsi kode-kode tersebut secara umum lebih pendek dari kelompok simbol yang digantikan dan *Statistical based* yang melakukan kompresi dengan pendekatan yang berbeda, di mana simbol-simbol yang ada diencode satu per satu, sehingga panjang dari kode output akan bervariasi tergantung dari probabilitas atau frekuensi pemunculan simbol.

Perancangan aplikasi kompresi file teks dengan menggunakan *Microsoft Visual Basic 2008* memudahkan proses kompresi hanya dengan menginputkan file teks maka akan di proses sehingga menghasilkan file hasil kompresi beserta *file headernya*. Hasil file teks sebelum dan sesudah dikompresi setelah dibandingkan telah mencapai 50% rasio perbandingannya[3].

Pendapat para ahli menyatakan kompresi data merupakan suatu upaya untuk mengurangi jumlah bit yang digunakan untuk menyimpan atau mentransmisikan data. Kompresi data meliputi berbagai teknik kompresi yang diterapkan dalam bentuk perangkat lunak (software) maupun perangkat keras (hardware). Bila ditinjau dari sisi penggunaannya, kompresi data bisa bersifat umum untuk segala keperluan atau bersifat khusus untuk keperluan tertentu.

Keuntungan data yang terkompresi antara lain: mengurangi *bottleneck* pada proses *I/O* dan transmisi data, penyimpanan data lebih hemat ruang, mempersulit pembacaan data oleh pihak yang tidak berkepentingan, dan memudahkan distribusi data dengan media removable seperti *flash disk*, *CD*, *DVD*, dll[4]. Kesimpulan dari penulis *kompresi* saat ini telah berkembang dengan pesat, sehingga teknik algoritma untuk mengkompresi data untuk menurunkan beban dari kapasitas tempat penyimpanan data, dimana dengan kompresi data yang di kurangi jumlah bit dapat memberikan ruang penyimpanan data yang lebih hemat dan pengamanan data dari orang yang ingin membuka sebuah dokumen yang telah berubah data algoritmanya.

2.3 Algoritma Boldi Vigna

Algoritma *Boldi-Vigna* dan *Taboo* merupakan contoh-contoh algoritma untuk melakukan kompresi dimana keduanya melakukan kompresi data dengan cara melakukan *descending sorting* terhadap masing-masing karakter, lalu diubah biner masing-masing karakter dengan pola yang sudah ditentukan. Pengujian dilakukan pada teks 2.148 karakter, 21.157 karakter, dan 211.571 karakter.

Sisa dari bagian ini menjelaskan kode *zeta* (ζ), juga dikenal sebagai *Boldi-Vigna kode*, diperkenalkan oleh *Paolo Boldi* dan *Sebastiano Vigna* sebagai keluarga panjang variabel kode yang merupakan pilihan terbaik untuk kompresi *webgraph*. Referensi asli adalah *Boldi dan Vigna*. Kita mulai dengan hukum *Zipf*, hukum kekuatan empiris *Zipf* yang diperkenalkan oleh ahli bahasa *George K. Zipf*. Ini menyatakan bahwa frekuensi setiap kata dalam bahasa alami kurang lebih berbanding terbalik dengan posisinya di tabel frekuensi. Secara intuitif, hukum *Zipf* menyatakan bahwa kata yang paling sering dalam bahasa apa pun adalah dua kali lebih sering daripada kata yang paling sering kedua, yang pada gilirannya dua kali lebih sering dari kata ketiga, dan seterusnya. Kode *zeta Boldi-Vigna* dimulai dengan bilangan bulat

positif k yang menjadi penyusutfaktor kode. Himpunan semua bilangan bulat positif dipartisi ke dalam interval $[20, 2k - 1]$, $[2k, 22k - 1]$, $[22k, 23k - 1]$, dan secara umum $[2hk, 2(h + 1)k - 1]$. Panjangnya setiap interval adalah $2(h + 1)k - 2hk$. Pengkompresian data telah berkembang dan masing-masing metode memiliki kelebihan dan kekurangan, kemampuan dari tiap-tiap metode pengkompresian data umumnya diukur dengan parameter ukuran atau kapasitas dan kecepatan kompresi, dengan kompresi atau mengecilkan ukuran file terlebih dahulu, maka kapasitas tempat penyimpanan yang diperlukan menjadi lebih kecil.

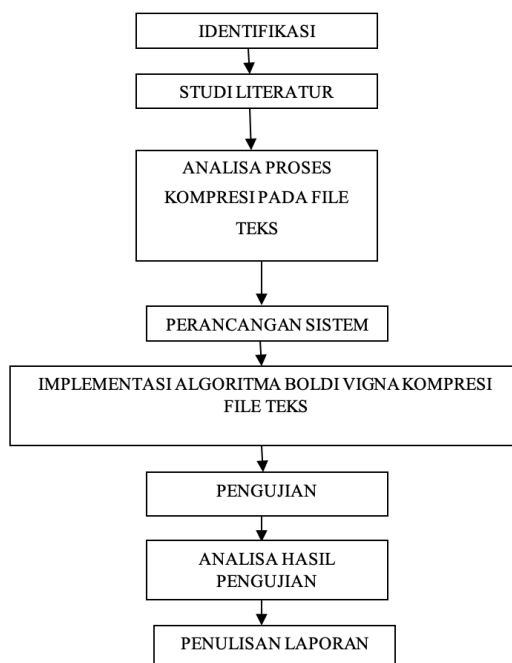
Cara kerja sistem yang akan dirancang bangun dengan menggunakan file teks yang berukuran byte yang kecil dengan sistem kompresi file teks dengan menggunakan algoritma boldi vigna akan mengubah teks yang berada pada file tersebut menjadi sebuah bilangan angka positif yang berpartisi kedalam nilai – nilai bilangan prima dengan ukuran byte yang kecil dan tidak membebani memory laptop dan hardisk pada komputer[5].

Pendapat lain dari para ahli menyatakan Teks dalam format *.txt merupakan format yang umum digunakan. Teks adalah kumpulan dari karakter-karakter atau string yang menjadi satu kesatuan. Teks yang memuat banyak karakter didalamnya selalu menimbulkan masalah pada media penyimpanan dan kecepatan waktu pada saat transmisi data. Media penyimpanan yang terbatas, membuat semua orang mencoba berpikir untuk menemukan sebuah cara yang dapat digunakan untuk mengompres teks. Algoritma Boldi-Vigna dan Taboo merupakan contoh-contoh algoritma untuk melakukan kompresi dimana keduanya melakukan kompresi data dengan cara melakukan *descending sorting* terhadap masing-masing karakter, lalu diubah biner masing-masing karakter dengan pola yang sudah ditentukan[6].

Kesimpulan dari penulis memakai algoritma boldi vigna ini berkeinginan untuk melakukan penelitian tentang bagaimana melakukan enkripsi file teks mengamankan data berupa file teks yang berisikan data atau berita pribadi menjadi keamanan data yang tidak bisa dicuri oleh orang lain yang tidak bertanggung jawab dengan mengubah data asli menjadi sebuah bilangan angka primer positif dengan algoritma bold ivigna.

2.4 Kerangka Kerja Penelitian

Pada penelitian ini, akan dilakukan analisa dan perancangan perangkat lunak pengkompresian file teks dengan menggunakan metode BOLDI VIGNA. Tahapan penelitian yang akan dilakukan akan dapat dilihat dalam kerangka kerja yang ada pada gambar 1.



Gambar 1. Kerangka Kerja Penelitian

Berdasarkan gambar 1 maka uraian kerja dapat dituliskan sebagai berikut:

- Mengidentifikasi Masalah
Pada tahap penelitian ini, yang akan dilakukan adalah menganalisis masalah yang terjadi pada pengkompresian file teks. Dimana masalah yang terkait seperti kapasitas penyimpanan data, dan proses transfer data lama karena kapasitas data yang besar.
- Studi Literatur
Pada tahap Studi Literatur, dilakukan pencarian data dan memperoleh informasi yang terkait dengan kompresi file teks dan Metode BOLDI VIGNA melalui jurnal, buku, atau melalui prosiding.
- Analisa Proses Kompresi Pada File Teks
Pada tahap analisa proses dekompresi pada file teks menggunakan metode BOLDI VIGNA untuk mengetahui suatu kinerja kompresinya.

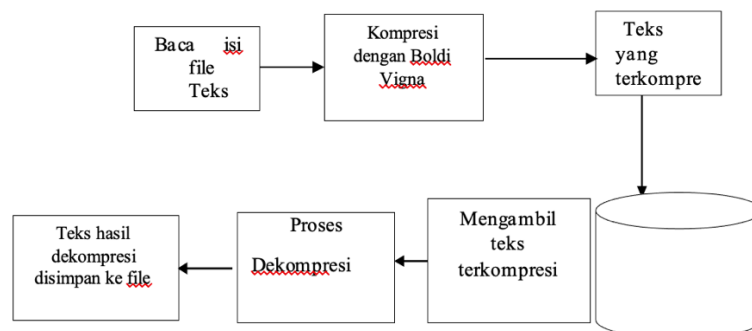
- d. Perancangan Sistem
Perancangan sistem dilakukan dengan merancang atau mendesain sebuah aplikasi kompresi dengan menggunakan pemograman Microsoft Visual Basic 2008.
- e. Penerapan Metode BOLDI VIGNA Pada Aplikasi yang Dirancang
Proses penerapan yang dilakukan dengan memasukkan metode BOLDI VIGNA ke dalam aplikasi kompresi yang telah dirancang sebelumnya, dan berupa kode program.
- f. Pengujian
Tahap pengujian ini, melakukan pengujian aplikasi yang telah selesai.
- g. Analisa Hasil Pengujian
Pada tahap ini, dilakukan analisa terhadap pengujian yang dilakukan secara manual dengan hasil pengujian yang dilakukan pada aplikasi yang telah dirancang.
- h. Penulisan Laporan
Penulisan laporan yang dilakukan untuk mendokumentasikan keseluruhan kegiatan penelitian dalam bentuk skripsi yang nantinya akan dibuat dalam bentuk artikel ilmiah dan akan dipublikasikan.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Sistem

Analisa sistem merupakan tahap awal dalam sebuah penelitian yang bertujuan mengetahui masalah terkait dalam pembuatan sebuah sistem dan menggambarkan proses-proses yang ada di dalam sistem untuk menghasilkan keluaran yang sesuai dengan kebutuhan pemakai. Pada tahap ini akan dijelaskan secara umum cara kerja Algoritma *boldi vigna* dalam kompresi file teks. Algoritma *boldi vigna* merupakan teknik kompresi dengan cara melakukan pengkodean dalam bentuk bit untuk mewakili data.

Analisa kebutuhan sistem membahas secara garis besar apa saja yang dibutuhkan dalam membangun sistem tersebut. Ada beberapa kebutuhan yang harus dimiliki sistem yaitu membaca file string dan file hasil kompresi, melakukan proses kompresi dan dekomposisi file teks berdasarkan algoritma Algoritma *boldi vigna*.



Gambar 2. Prosedur Kompresi File Teks

Langkah-langkah proses kompresi dengan *algoritma Boldi Vigna* adalah sebagai berikut:

- a. Membaca string yang terdapat pada file teks.
- b. Merubah karakter menjadi huruf kecil
- c. Melakukan proses kompresi
 1. Mengganti pasangan karakter yang muncul lebih dari satu kali dengan simbol non-terminal (A, B, C,...)
 2. Setelah tidak ada lagi pasangan karakter yang muncul lebih dari satu kali, maka lakukan proses perhitungan
- d. Hasil dari tahap 3.a dilanjutkan ke perhitungan
- e. Karakter yang sudah diganti dengan bit selanjutnya disimpan kembali ke fileteks.

Berikut adalah langkah-langkah proses kombinasi dekomposisi dengan algoritma Boldi Vigna:

- a. Baca isi *file* teks yang telah terkompresi.
- b. Selanjutnya hapus *flag bit* dan *padding* dari nilai keseluruhan *bit*.
- c. Pengecekan bit yaitu dengan cara melakukan cek bit dari bit pertama dengan tabel kode. Jika ditemukan bit yang sesuai dengan tabel kode maka ubah nilai string yang sesuai sehingga akan mendapatkan hasil semula sebelum dikompresi selanjutnya setelah dari dekomposisi, lakukan pengembalian karakter non-terminal ke pasangan karakter yang ada pada tabel kamus.

3.2 Penerapan Algoritma Boldi Vigna

Pada tahap ini akan dilakukan kompresi dan dekomposisi beberapa string dengan algoritma Boldi Vigna. Berikut ini merupakan contoh proses kompresi dengan Algoritma Boldi Vigna.

- a. Input karakter yang akan dijadikan teks sebagai berikut.
String : JULEHA AKANMAKAN

Jumlah : 17
 Karakter : {J, U, L, E, H, A, Sp, K, N, M}

- b. Membuat tabel karakter dan frekuensi Pada langkah ini akan dilakukan proses perhitungan ukuran bit awal dari string di atas.

Tabel 1. Karakter Dan Frekuensi

| Char | Ascii Code | Ascii Binary | Bit | Frekuensi | Bit x Frekuensi |
|--------------|------------|--------------|-----|-----------|-----------------|
| J | 74 | 01001010 | 8 | 1 | 8 |
| U | 85 | 01010101 | 8 | 1 | 8 |
| L | 76 | 01001100 | 8 | 1 | 8 |
| E | 69 | 01000101 | 8 | 1 | 8 |
| H | 72 | 01001000 | 8 | 1 | 8 |
| A | 65 | 01000001 | 8 | 5 | 40 |
| Sp | 32 | 00010000 | 8 | 2 | 16 |
| K | 75 | 01001011 | 8 | 2 | 16 |
| N | 78 | 01001110 | 8 | 2 | 16 |
| M | 77 | 01001101 | 8 | 1 | 8 |
| Total | | | | | 136 |

- c. Kompresi dengan Algoritma Boldi Vigna
 Pada tahapan ini string di atas akan kita proses dengan menggunakan algoritma Boldi Vigna. Berikut adalah tahapan algoritma Boldi Vigna:
1. Baca string, dan ubah semua karakter menjadi huruf kecil (*lowercase*).
 2. Mencari pasangan karakter yang muncul lebih dari 1 kali.
 3. Ganti pasangan karakter yang muncul lebih dari 1 kali dengan simbol non-terminal (A, B, C,..).
 4. Ulangi langkah ke-2 (b) sampai tidak ada pasangan karakter lagi.

Tabel 2. Kompresi

| Proses | String | Pasangan Karakter | Penggantian Simbol | Aturan | String Hasil |
|--------|-----------------|-------------------|--------------------|----------------------|---------------|
| 1 | Julehaakanmakan | Ak | A | A = ak | JulehaAanmAan |
| 2 | julehaAanmAan | Aa | B | A = ak B = Aa | julehaBnmBn |
| 3 | julehaBnmBn | Bn | C | A = ak B = Aa C = Bn | Juleha C mC |

Dari tabel 2 proses sequitur didapat hasil akhir string adalah "juleha C mC" Melihat dari algoritma kompresi yang akan menghitung ukuran akhir dari string setelah diproses, maka dalam penelitian ini tidak akan dilakukan perhitungan akhir. Dari tabel tersebut dapat dibentuk bit dan string sebelum dikompresi yaitu "juleha C mC" Menjadi string bit "1001, 11011, 10001, 110101, 110011, 1110111, 0, 101, 0, 100001, 101". Sebelum ditulis ke sebuah file hasil kompresi dilakukan penambahan string bit itu sendiri apakah habis dibagi 8 dan berapa sisanya jika dibagi 8. $100111011100011101011100111101110101010001101$. Jumlah string 47 jika dibagi 8 maka memiliki sisa 7 (dinyatakan sebagai n) agar bit dapat hasil dibagi 8 maka dapat ditambahkan padding bit 0 sebanyak $7 - n + "1"$, maka $7 - 7 + "1" = 1$. Karena 7-7 habis, maka hanya menambah padding "1" pada akhir *string bit*.

Sehingga hasilnya:

10011101 11000111 01011100 11111011 10101010 00011011

Kemudian tambahkan *flag bit* yaitu *biner* dari *decimal* dengan rumus $9 - n = 9 - 7 = 2 = 00000010$, sehingga menjadi:

10011101 11000111 01011100 11111011 10101010 0001101100000010

Lalu ubah masing-masing 8 bit menjadi karakter kembali dan disimpan pada *file*.

10011101 = •
 11000111 = Ç
 01011100 = \
 11111011 = û
 10101010 = a
 00011011 =
 00000010 =

Dan dapat dihitung kinerja kompresinya menurut parameter yang sudah ditentukan yaitu:

Compression ratio (Cr)

$$Cr = \frac{\text{Ukuran data setelah di kompresi}}{\text{Ukuran data sebelum di kompresi}} * 100\%$$

$$= \frac{47 \text{ bit}}{136 \text{ bit}} * 100\%$$

= 34,55%

Maka dari hasil *compression ratio* didapatkan bahwa sebanyak 34.55% sudah berhasil dimampatkan.

d. Dekompresi

Proses dekompresi dilakukan dengan merubah karakter huruf menjadi *biner* sehingga didapatkan string bit kembali, lalu hilangkan padding dan *flagbit* pada string bit. 10011101 11000111 01011100 11111011 10101010 00011011 00000010. Baca 8 bit terakhir pada string bit lalu rubah ke desimal, nyatakan sebagai n. 8 bit terakhir: 00000010, desimal = 2, selanjutnya dihitung dengan rumus $7 + n = 7 + 2 = 9$, maka dihilangkan sebanyak 9 bit pada akhir dari string bit, sehingga hasil akhir setelah dihilangkan padding dan flagbit adalah:

10011101 11000111 01011100 11111011 10101010 0001101

Lalu, lakukan pembacaan bit dari kiri ke kanan dan bandingkan dengan tabel 3 sehingga dapat string awal kembali:

Tabel 3. Dekompresi

| Bit | Keterangan |
|---------|------------|
| 1 | - |
| 10 | - |
| 100 | - |
| 1001 | J |
| 1 | - |
| 11 | - |
| 110 | - |
| 1101 | - |
| 11011 | U |
| 1 | - |
| 10 | - |
| 100 | - |
| 1000 | - |
| 10001 | L |
| 1 | - |
| 11 | - |
| 110 | - |
| 1101 | - |
| 11010 | - |
| 110101 | E |
| 1 | - |
| 11 | - |
| 110 | - |
| 1100 | - |
| 11001 | - |
| 110011 | H |
| 1 | - |
| 11 | - |
| 111 | - |
| 1110 | - |
| 11101 | - |
| 111011 | - |
| 1110111 | A |
| 0 | Spasi |
| 1 | - |
| 10 | - |
| 101 | C |
| 0 | Spasi |
| 1 | - |
| 10 | - |
| 100 | - |
| 1000 | - |
| 10000 | - |

| Bit | Keterangan |
|--------|------------|
| 100001 | M |
| 1 | - |
| 10 | - |
| 101 | C |

Setelah dilakukan pembacaan maka didapat hasil dari dekompresi adalah “juleha C mC”. Dekompresi pada algoritma Boldi Vigna yaitu mengubah karakter simbol non-terminal kembali ke pasangan karakter menggunakan tabel 3.4 melihat dari aturan yang telah ada, maka didapat hasil dekompresi yaitu:

Tabel 4. Dekompresi

| Proses | String Hasil | Aturan | Simbol non-terminal | Perubahan symbol kepasangan karakter | String Awal |
|--------|---------------|----------------------------|---------------------|--------------------------------------|-----------------|
| 1 | juleha C mC | A = ak B = Aa C = Bn | C | Bn | julehaBnmBn |
| 2 | julehaBnmBn | A = ak B = Aa | B | Aa | JulehaAanmAan |
| 3 | julehaAanmAan | A = ak | A | Ak | Julehaakanmakan |

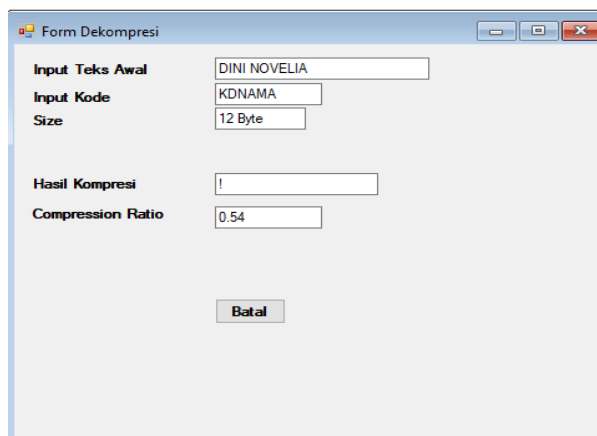
Setelah dilakukan proses perubahan simbol non-terminal ke pasangan karakter berdasarkan aturan pada tabel 4 maka didapat hasil akhir yaitu “juleha akan makan”.

3.3 Tampilan Program

Pada tampilan program ini yaitu tampilan dari aplikasi dengan penerapan algoritma *boldi vigna* pada aplikasi kompresi file teks berbasis desktop. Aplikasi yang akan dijalankan dirancang dengan menggunakan aplikasi *Microsoft Visual Basic 2008*. Tampilan yang dibutuhkan yaitu tampilan *input* dan tampilan *output*.

a. Proses kompresi

Pada proses kompresi maka yang pertama sekali dilakukan adalah menginput data teks yang ingin di kompres. Adapun tampilannya adalah dapat dilihat pada gambar 3 sebagai berikut:



Gambar 3. Proses Kompresi

b. Proses Dekompresi

Pada proses dekompresi maka yang pertama kali dilakukan yaitu menginput kode pemanggilan. Adapun tampilan dari proses dekompresi adalah sebagai berikut:



Gambar 4. Proses Dekompresi

4. KESIMPULAN

Kesimpulan dari implementasi dan pengujian kompresi file teks dengan algoritma Boldi Vigna menunjukkan bahwa algoritma ini bekerja dengan mengubah ukuran data menjadi lebih kecil menggunakan kode zeta, dimulai dengan bilangan bulat positif k yang berperan sebagai penyusun kode. Selain itu, kompresi file teks ini memungkinkan pembuatan partisi pada harddisk untuk meminimalisir risiko bad sector, di mana partisi membantu menjaga data tetap aman meskipun ada kerusakan pada salah satu bagian. Aplikasi kompresi file teks ini dirancang menggunakan Microsoft Visual Basic, yang mempermudah proses kompresi hanya dengan menginput file teks yang akan dikompres, menghasilkan file terkompresi beserta file header-nya.

REFERENCES

- [1] P.N Primadani, "Kompresi Data Text Menggunakan Metode 'Duplicate Word Indexing,'" vol. 03, no. 01 , pp 60-62, 2018.
- [2] S. D. Nasution, "Perancangan Aplikasi Kompresi File Teks Dengan Menggunakan Algoritma Goldbach Codes," Jurnal Infotek STIEKOM, vol. 1, no. Februari, pp. 2008-2010, 2016.
- [3] B. David, Handbook of data Compression, Estudio Ca. London: ISBN 978-1-84882-902-2 e-ISBN 978-1-848882-903-9, 2010.
- [4] D. R. DE SOUZA, "Algorithm Boldi-Vigna(ζ 4) Codes" IOSR J. Econ. Finance., vol. 3, no. 1, p. 56, 2016, doi: <https://doi.org/10.3929/ethz-b000238666>.
- [5] D. Atmodjo, "Peningkatan Keamanan Data dengan metode Cropping Selection Pseudorandom," vol. 4, no. 3, pp. 132-138, 2016
- [6] U. Rusmawan, Pemograman Aplikasi DekstopVB.NET, PT Elex Me. Jakarta: Gramedia, 2011.
- [7] S. Santoso and R. Nurmalina, "Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut)," J. Integr., vol. 9, no. 1, pp. 84-91, 2017.
- [8] R. A. Sukamto and M. Shalahuddin, Rekayasa Perangkat Lunak (Arsitektur dan Berorientasi Objek), Bandung: Informatika, 2014.
- [9] R. Y. Lubis and M. Syahril, "Pemilihan Editor Berita Terbaik Menggunakan Metode Complex Proportional Assessment (COPRAS)," vol. 3, no. 4, pp. 738–747, 2020.
- [10] D. Alamsyah, R. Nuraini, and M. Bagir, "Implementasi Metode Complex Proportional Assessment (COPRAS) Pada Sistem Pendukung Keputusan Pemilihan Bluetooth Audio Transmitter," vol. 3, no. 3, pp. 123–132, 2022, doi: 10.47065/josyc.v3i3.1695.
- [11] J. H. Lubis, S. Esabella, Mesran, Desyanti, and D. M. Simanjuntak, "Penerapan Metode Multi Attribute Utility Theory (MAUT) Dalam Pemilihan Karyawan yang di Non-Aktifkan di Masa Pandemi," vol. 6, no. April, pp. 969–978, 2022, doi: 10.30865/mib.v6i2.3909.
- [12] S. Manurung, "Sistem Pendukung Keputusan Pemilihan Guru Dan Pegawai Terbaik Menggunakan Metode Moora," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 1, pp. 701–706, 2018, doi: 10.24176/simet.v9i1.1967.
- [13] S. R. Tanjung and M. V Siagian, "Penerapan Metode COPRAS dan ENTROPY dalam Pemilihan Anggota Badan Pengawas Pemilihan Umum (BAWASLU)," vol. 1, no. 2, pp. 48–59, 2021.
- [14] A. Hia and T. Syahputra, "Sistem Pendukung Keputusan Menentukan Performance Cleaning Service Menggunakan Metode COPRAS," vol. 1, pp. 157–171, 2022.
- [15] A. G. Simorangkir and Mesran, "SPK Pemilihan Konten YouTube Layak Tonton untuk Anak-Anak Menerapkan Metode ROC (Rank Order Centroid) dan COPRAS (Complex Proportional Assessment)," 2022.
- [16] L. Irvana and N. Mariana, "Penerapan Metode COPRAS Untuk Pemilihan SMK Jurusan TKJ Kota Semarang," vol. 11, pp. 201–207, 2022.