



Penerapan Algoritma Punctured Elias Code pada Aplikasi Kamus Besar Bahasa Indonesia (KBBI) Berbasis Android

Zulkhairia

Program Studi Teknik Informatika, Fakultas Ilmu Komputer Dan Teknologi Informasi, Universitas Budi Darma,
Jalan Sisingamangaraja No.338, Medan, Sumatera Utara, Indonesia
Email: Zulkhairia@gmail.com

Abstrak - Kamus besar bahasa indonesia memiliki ukuran yang besar karena kualitas kamus yang dihasilkan lebih baik, maka ukuran File yang dibutuhkan untuk tersebut semakin besar. Dan proses kompresi dibutuhkan untuk mengurangi kapasitas file menjadi lebih kecil agar hemat dalam pemakaian ruang penyimpanan. Dengan ukuran data yang relatif besar, pada saat proses transfer data bisa gagal karena media penyimpanan tidak memiliki ruang yang cukup. Dengan melakukan proses kompresi pada kamus besar bahasa indonesia yang lumayan besar bisa dilakukan proses untuk memperkecil ukuran data dengan melakukan pengurangan bit pada kamus besar bahasa indonesia tersebut sehingga mempermudah untuk melakukan suatu proses pemindahan data dengan efektif. Prosedur untuk mengompresi kamus besar bahasa indonesia (KBBI) dimulai dari memilih kamus besar bahasa indonesia yang akan dikompresi, lalu buku tersebut akan dikompresi dengan algoritma punctured Elias code sehingga menghasilkan sebuah buku terkompresi yang memiliki ukuran lebih kecil. Penerapan algoritma punctured Elias code untuk kompresi kamus besar bahasa indonesia (KBBI) dilakukan dengan membaca nilai heksadesimal dari suatu kamus besar bahasa indonesia (KBBI), lalu merubah nilai heksa tersebut menjadi nilai bit baru yang berupa bilangan biner, kemudian menyusun kembali nilai biner tersebut menjadi karakter baru. Dimana hasil kompresi berdasarkan rasionya sebesar 32% berdasarkan sampel data 80 bit dengan hasil kompresi 25 Bit.

Kata Kunci: KBBI; Punctured Elias Code; Kompresi

Abstract-The large Indonesian dictionary has a large size because the quality of the resulting dictionary is better, so the file size needed for it is getting bigger. And the compression process is needed to reduce the file capacity to be smaller in order to save on the use of storage space. With a relatively large data size, during the data transfer process it can fail because the storage media does not have enough space. By performing the compression process on the large Indonesian dictionary, a process can be carried out to reduce the size of the data by reducing the bits in the large Indonesian dictionary, making it easier to carry out a data transfer process effectively. The procedure for compressing the large Indonesian dictionary (KBBI) starts from selecting the large Indonesian dictionary to be compressed, then the book will be compressed with the punctured Elias code algorithm so as to produce a compressed book that has a smaller size. The implementation of the punctured Elias code algorithm for compression of the Big Indonesian Dictionary (KBBI) is done by reading the hexadecimal value from a Big Indonesian Dictionary (KBBI), then changing the hex value into a new bit value in the form of a binary number, then rearranging the binary value into new character. Where the compression results are based on the ratio of 32% based on 80 bit data samples with 25 Bit compression results.

Keywords: KBBI; Punctured Elias Code; Compression

1. PENDAHULUAN

Kompresi data adalah proses untuk melakukan proses konversi data masukan (the source atau data asli) menjadi sebuah aliran data yang lain (the output, aliran bit atau aliran yang sudah dikompresi) sehingga memiliki ukuran yang lebih kecil. Masalah yang ada pada kompresi data adalah bagaimana melakukan proses untuk memampatkan data asli sehingga menjadi ukuran yang lebih kecil dari ukuran asli. Data yang telah dimampatkan merupakan representasi data dalam bentuk string bit.

Kamus besar bahasa indonesia memiliki ukuran yang besar karena kualitas kamus yang dihasilkan lebih baik, maka ukuran File yang dibutuhkan untuk tersebut semakin besar. Dan proses kompresi dibutuhkan untuk mengurangi kapasitas file menjadi lebih kecil agar hemat dalam pemakaian ruang penyimpanan. Dengan ukuran data yang relatif besar, pada saat proses transfer data bisa gagal karena media penyimpanan tidak memiliki ruang yang cukup. Dengan melakukan proses kompresi pada kamus besar bahasa indonesia yang lumayan besar bisa dilakukan proses untuk memperkecil ukuran data dengan melakukan pengurangan bit pada kamus besar bahasa indonesia tersebut sehingga mempermudah untuk melakukan suatu proses pemindahan data dengan efektif.

Dikutip dari kamus besar bahasa indonesia (KBBI), buku merupakan lembar kertas yang berjilid, yang berisi tulisan ataupun kosong. Buku yang memiliki isi berupa tulisan tidak hanya menjadi panduan bagi pembaca yang mencari sesuatu yang ingin diketahui, tetapi juga memiliki kekuatan yang mampu merubah otak manusia. Buku memiliki ketebalan beragam tergantung dari isi buku tersebut. Buku atau kamus seperti KBBI memiliki halaman hingga 2040 halaman. Dengan halaman sebanyak 2040, KBBI memiliki ketebalan yang cukup besar. Maka, untuk mengatasi ketebalan tersebut dibuatlah sebuah aplikasi KBBI yang bisa diakses dimana saja dan kapan saja. Masalah tersebut dapat diatasi dengan dilakukannya pengurangan data yang berulang atau biasa disebut proses pemampatan data (kompresi).

Kompresi digunakan dalam segala keperluan, contohnya mem-back up data dan pemindahan data. Untuk mem-back up data tidak perlu untuk menyalin semua file asli, dengan adanya kompresi atau proses mengecilkan ukuran file sebelumnya, maka kapasitas yang digunakan pada tempat penyimpanan menjadi lebih kecil, jika suatu saat data yang telah dikecilkan tersebut diperlukan, maka akan dilakukan proses pengembalian ukuran ke ukuran aslinya (dekompresi file).

Penelitian sebelum Punctured Elias Codes merupakan metode yang dirancang oleh Fenwick pada sebuah



percobaan yang dilakukan untuk meningkatkan performa metode the burrows-wheeler transform. Istilah punctures berasal dari tempat pengawasan error kode-kode (ECC). ECC terdiri dari data asli yang ditambah sejumlah bilangan dari check bits. Jika check bits dihilangkan, untuk mempersingkat dapat diklarifikasi sebagai teks, suara, gambar dan video.

Kata kamus diambil dari kata dalam bahasa arab yaitu, qamus (dalam bentuk jamak qawamus). Bahasa arab mengambil kata kamus dari bahasa Yunani kuno, okeanos yang memiliki arti lautan. Bagaimana kata kamus berasal dari bahasa Yunani kuno okeanos yang berarti lautan. Sejarah mengatakan, bahwa kata kamus memiliki makna dasar yaitu wadah pengetahuan, khususnya pengetahuan bahasa yang tidak terhingga dalam dan luasnya, seluas dan sedalam lautan[1].

Menurut Chaer, kata kamus dalam bahasa Inggris yaitu dictionary, dan mulai digunakan untuk membuat karya tulis di tahun 1526 dan berasal dari kata dalam bahasa Latin, yaitu dictionarium. Kata tersebut diturunkan dari kata diction yang berarti kata ataupun berkata. Terjemahannya dalam bahasa Belanda adalah woordens boek yang berbeda dari woordens chat yang jika dalam bahasa Indonesia berarti perbendaharaan atau kosa kata[2]

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Untuk mendukung kelancaran penelitian ini, maka dilakukan tahapan penelitian sebagai berikut:

- Studi Kepustakaan (*Library Research*)
Yaitu dengan menggunakan buku, *paper* dan number ilmiah lain yang berhubungan dengan penelitian ini.
- Perancangan (*Design*)
Merancang suatu program untuk dapat melihat hasil metode dalam kompresi *file* teks.
- Pengkodean (*Coding*)
Yaitu untuk membuat aplikasi yang sudah dirancang sebelumnya.
- Pengujian (*Testing*)
Perangkat lunak akan diuji untuk mengetahui system tersebut sudah berjalan sesuai fungsi atau tidak. Jika perangkat lunak sudah berfungsi dengan baik maka akan ditarik suatu kesimpulan.
- Penerapan (*Application*)
Penerapan Algoritma *Punctured Elias Code* Aplikasi Kamus Besar Bahasa Indonesia (KBBI) Berbasis Android”
- Laporan (*Report*)
Merupakan laporan ilmiah untuk memudahkan pembaca dalam memahami langkah – langkah yang sudah dilakukan dalam penelitian dan hasilnya.

2.2 Kompresi Data

Kompresi data pada konteks ilmu komputer merupakan suatu ilmu atau seni dalam merepresentasikan sebuah informasi yang ada pada data ke dalam bentuk yang lebih kecil. Perkembangan dalam ilmu komputer dan multimedia mengakibatkan proses kompresi data menjadi hal yang penting dan sangat berguna dalam kemajuan teknologi pada saat ini[3].

Definisi dari kompresi data adalah suatu proses untuk mengkonversi sebuah data masukan (the source atau data asli mentah) menjadi suatu data lain (the Output, aliran bit, atau aliran sudah dikompresi) yang memiliki ukuran yang lebih kecil. Aliran data (stream) tersebut dapat berupa sebuah file atau buffer pada memori. Data dalam konteks kompresi melingkupi segala bentuk digital dari informasi, yang diproses oleh program komputer. Bentuk informasi tersebut secara luas diklasifikasikan sebagai teks, suara, gambar dan video[4].

Dekompresi adalah proses untuk mengembalikan sebuah data yang telah dikompresi ke bentuk aslinya. Untuk merubah data yang telah dikompres maka perlu cara yang berbeda sama seperti pada waktu proses dilakukannya kompresi[5].

2.3 Algoritma Punctured Elias Codes

Punctured Elias Codes dirancang oleh *Peter Fenwick* dalam sebuah percobaan yang dilakukan dalam proses peningkatan performa *the Burrows–Wheeler transform*. Istilah *Punctured* sendiri datang dari tempat pengawasan error kode-kode (ECC). ECC terdiri dari data yang asli kemudian ditambah sejumlah bilangan dari *check bits* [1].

Cara untuk membangun kode *Punctured Elias Codes* ini adalah sebagai berikut:

- Ambil bilangan biner dari n ,
- Reversed (balikkan bit-bitnya), dan siapkan flag untuk menunjukkan jumlah bit yang bernilai 1 di dalam n .
- Untuk setiap bit 1 di dalam n kita siapkan flag dari 1 dan akhiri flag dengan 0.
- Gabungkan flag dengan bilangan biner yang sudah dibalikkan (reversed).

n	Binary	Reserved	Flag	Flag Reserved	P1
0	0	-	-	-	0
1	1	1	10	10 1	101
2	10	01	10	10 01	1001
3	11	11	110	110 11	11011
4	100	001	10	10 001	10001
5	101	101	110	110 101	110101
6	110	011	110	110 011	110011
7	111	111	1110	1110 111	1110111
8	1000	0001	10	10 0001	100001
9	1001	1001	110	110 1001	1101001
10	1010	0101	110	110 0101	1100101
11	1011	1101	1110	1110 1101	11101101
12	1100	0011	110	110 0011	1100011
13	1101	1011	1110	1110 1011	11101011
14	1110	0111	1110	1110 0111	11100111

Gambar 1. Tabel *Punctured Elias Code*

a. Ratio of Compression (Rc)

$$Rc = \frac{\text{Ukuran data sebelum dikompres}}{\text{Ukuran data setelah dikompres}}$$

b. Compression ration (Cr)

$$Cr = \frac{\text{Ukuran data setelah dikompres}}{\text{Ukuran sebelum dikompres}} \times 100\%$$

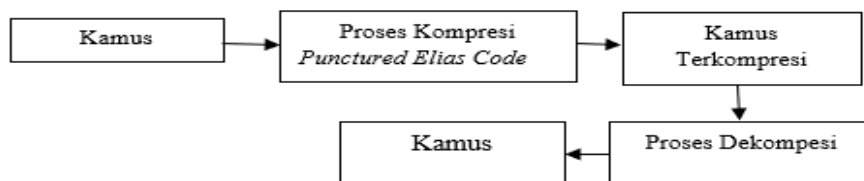
c. Space Saving (SS)

$$SS = \text{Ukuran data sebelum dikompresi} - \text{Ukuran data setelah dikompresi}$$

3. HASIL DAN PEMBAHASAN

Berdasarkan penelitian ini, akan dilakukan analisa dan perancangan perangkat lunak pengkompresian *file* dengan menggunakan algoritma *Punctured Elias Code*. Algoritma *Punctured Elias Code* merupakan salah satu algoritma kompresi yang mengkompres suatu data ke dalam bentuk data yang lain dengan lebih sering menggunakan sedikit bit dan jarang menggunakan lebih banyak bit karakter. Penggunaan algoritma *Punctured Elias Code* dalam aplikasi kompresi file teks yaitu untuk merubah ukuran file asli menjadi file yang lebih kecil ukurannya. Prosedur untuk mengkompresi file dimulai dari memilih file teks yang akan dikompresi, lalu file tersebut akan dikompresi dengan algoritma *Punctured Elias Code* sehingga menghasilkan sebuah file terkompresi yang memiliki ukuran lebih kecil. Sedangkan untuk proses dekompresinya yang menjadi output adalah file terkompresi lalu file tersebut akan didekompresi menggunakan algoritma *Punctured Elias Code* sehingga menghasilkan file yang teks memiliki ukuran yang sebelum dikompresi atau balik menjadi file awal. Untuk mengetahui prosedur kompresi dan dekompresi suatu *file* dapat dilihat pada gambar 2.

Permasalahan yang dibahas adalah membuat suatu pengkodean karakter kompresi dengan menggunakan algoritma *Punctured Elias Code*. Berikut ini contoh pengaplikasian kompresi dan dekompresi dengan menggunakan algoritma *Punctured Elias Code*. Berikut merupakan prosedur kompresi dan dekompresi *record database* kamus teknologi informasi, Pada kamus akan diproses kompresi sehingga kamus besar bahasa indonesia (KBBI) terkompresi jika sudah terkompresi maka simpan kedalam terkompresi menggunakan algoritma *Punctured Elias Code*, terkompresi maka aplikasi kamus teknologi informasi yang menggunakan algoritma *Punctured Elias Code* akan membaca *record database* kamus, kamus dibaca maka akan melakukan proses dekompresi setelah dekompresi dilakukan maka akan menampilkan isi kamus.



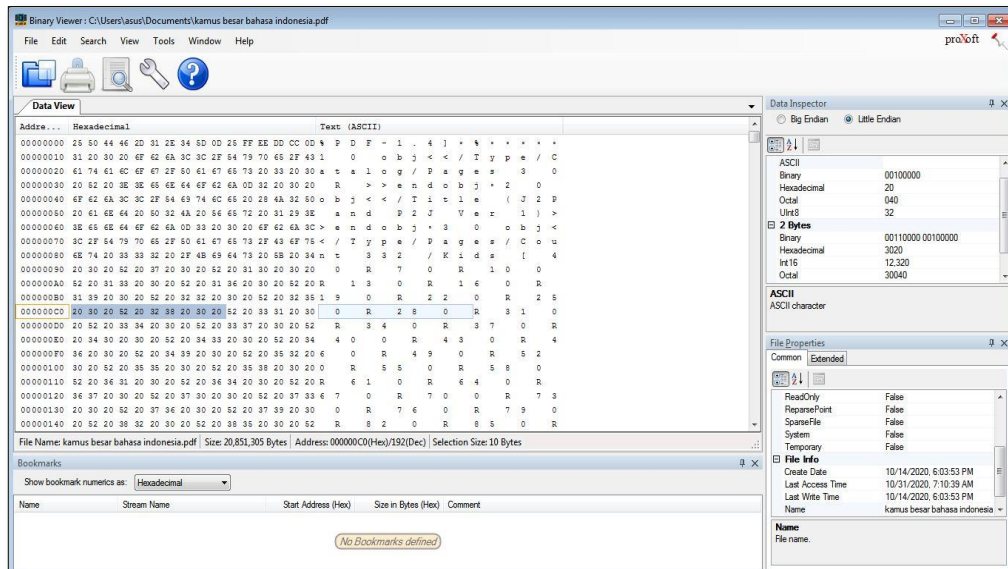
Gambar 2. Prosedur Kompresi dan Dekompresi

Berdasarkan analisa, *kompresi file* memiliki ukuran lebih besar ukuran yang lainnya. Dengan melakukan kompresi file, file 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 file dengan algoritma stout codes

Contoh file kamus besar bahasa indonesia yang akan dikompres adalah file berektensi FDF yang akan dilakukan sebuah proses kompresi. Sebelum file dikompresi, terlebih dahulu dilakukan pembacaan pada file untuk mendapatkan file berupa biner. Membaca biner yang terdapat pada file FDF menggunakan aplikasi Binary Viewer untuk mencari biner pada aplikasi file. Berikut adalah contoh file kamus besar bahasa indonesia yang akan dikompresi dan dekompresi.

Nama File kamus besar bahasa indonesia
 Extension File FDF.
 Size 19.8 Mb



Gambar 3. File Teks Yang Akan Dikompresi

Berdasarkan contoh diatas di dapat nilai biner viewer, adapun nilai hexa pada file Teks sempel C:\Uses\asus\Documents\file apijuddin\Apijuddin[1].docx tersebut adalah:

Tabel 2. Nilai Hexadesimal berdasarkan binary viewer

20	30	20	52	20	32	38	20	30	20
----	----	----	----	----	----	----	----	----	----

3.1 Kompresi Algoritma Stout Codes

Dalam proses kompresi algoritma stout codes yaitu dengan mengurutkan frekuensi dari karakter yang muncul lalu merubah karakter tersebut menjadi codeword. Berikut ini langkah-langkah untuk mengkompresi file teks menggunakan algoritma stout codes :

- a. Buat daftar frekuensi kemunculan tiap-tiap karakter dan mengurutkan dari frekuensi terbesar sampai frekuensi terkecil.

Tabel 3. Daftar Frekuensi Kemunculan Karakter

n	Hexsa	ASCII Biner	Bit	Frekuensi	Bit x Frekuensi
1	20	10000000	8	5	40
2	30	11000000	8	2	16
3	52	10100100	8	1	8
4	32	11001000	8	1	8
5	38	11100000	8	1	8
Total					80

- b. Melakukan proses kompresi

Tabel 4. Proses kompresi dengan algoritma Punctured elias code

No	Karakter	Frekuensi	Punctured Elias Code (P1)	Bit	Frekuensi x Bit
1	20	5	0	1	5
2	30	2	101	3	6
3	52	1	1001	4	4
4	32	1	11011	5	5

5	38	1	10001	5	5
Total			25		

Berdasarkan pada tabel di atas dapat dibentuk nilai bit baru hasil kompresi dari susunan nilai hexadecimal sampe awal sebelum kompresi yaitu, 20, 30, 20, 52, 20, 32, 38, 20, 30, 20 (tanpa ada tanda koma dan spasi) menjadi nilai *bit biner* "0000010110110011101110001",

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 25 tidak habis dibagi delapan dan sisanya 6 bit, nyatakan sisa bagi tersebut dengan nilai n . maka tambahkan 0 sebanyak 0 sebanyak $7 - n + "1"$ di akhir string bit. Nyatakan dengan L. Lalu tambahkan bilangan biner dari $9 - n$. Nyatakan dengan bit akhir

$$7 - n + "1"$$

$$7 - 2 + "1" = 000001$$

Bit Akhir $9 - n$

Bit Akhir = $9 - 2 = 7 = 00000111$

Gambar 5. Perhitungan Penambahan Bit

00000101,10110011,10111000,10000000,00000111Total panjang bit keseluruhan setelah ada penambahan bit adalah $25+7+8=40$. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti gambar di bawah ini.

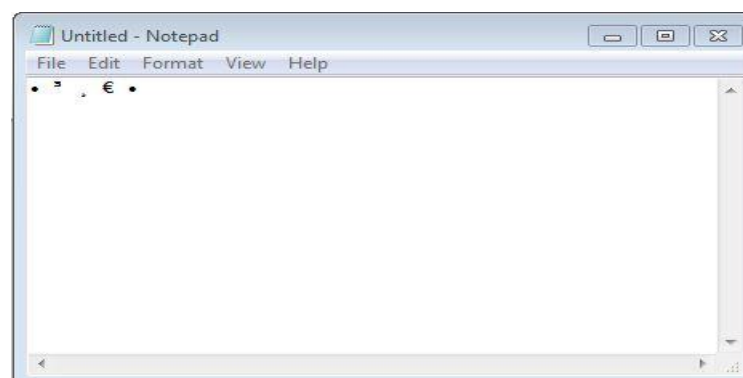
00000101 10110011 10111000 10000000 00000111

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 5 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 5. Nilai Desimal terkompresi

Biner	Nilai Desimal Terkompresi	Simbol
00000111	7	•
10110011	179	3
10111000	184	,
10000000	128	€
00000111	7	•

Setelah nilai desimal diketahui, maka mengubah nilai desimal kedalam suatu karakter. Karakter hasil dari proses kompresi yang dihasilkan tersimpan dalam suatu *file* dengan ekstensi ".sc", dan jika *file* tersebut dibuka dengan aplikasi notepad, maka akan tampil karakter seperti gambar berikut :



Gambar 6. Hasil Karakter Kompresi

Setelah mengetahui hasil kompresi tersebut, hasil kompresi dapat di ukur atau di hitung sebagai berikut :

$$Cr = 100\% - \frac{\text{ukuran bit data setelah dikompresi}}{\text{ukuran bit data sebelum dikompresi}} \times 100\%$$



$$Cr = 100\% - \frac{25bit}{80 bit} \times 100\%$$

$$Cr = 100\% - 45\%$$

$$Cr = 32\%$$

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 6. Hasil Telkompres

Simbol	Nilai Desimal Terkompresi	Biner
•	7	00000111
³	179	10110011
,	184	10111000
€	128	10000000
•	7	00000111

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi “00000101,10110011,10111000,10000000,00000111”

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+6= 12. Penjelasan diatas menunjukan bahwa bit akhir harus dihilangkan. Hasil pengembalian binary menjadi string bit semula dapat dilihat sebagai berikut ini: “0000010110110011101110001”

Berdasarkan perhitungan dengan algoritma punctured code string bit pada diatas berjumlah 25 bit seperti diawal sehingga dilakukan pembacaan string bit awal. Adapun tabel hasil perhitungan diatas adalah sebagai berikut:

Tabel 7. Pengecekan Bit

Indeks	Nilai	Keterangan
1	0	Ada pada tabel
2	0	Ada pada tabel
3	0	Ada pada tabel
4	0	Ada pada tabel
5	0	Ada pada tabel
6	1	Tidak Ada
7	10	Tidak Ada
8	101	Ada pada tabel
9	1	Tidak Ada
10	10	Tidak Ada
11	101	Ada pada tabel
12	1	Tidak Ada
13	10	Tidak Ada
14	100	Tidak Ada
15	1001	Ada pada table
16	1	Tidak Ada
17	11	Tidak Ada
18	110	Tidak Ada
19	1101	Tidak Ada
20	11011	Ada pada table
21	1	Tidak Ada
22	10	Tidak Ada
23	100	Tidak Ada
24	1000	Tidak Ada
25	10001	Ada pada tabel

Maka dari penjabaran diatas dapat dibentuk tabel punctured code string bit dan nilai hexa awal. Berdasarkan hasil dekompresi di atas didapati nilai sampel hexadesimal berdasarkan binary viewer sehingga dihasilkan string semula seperti tabel di bawah



Tabel 8. Tabel *Hexa* dan *stout codes*

N	Hexsa	Codeword	Bit	Frekuensi	Bit x Frekuensi
1	20	0	1	5	5
2	30	101	3	2	6
3	52	1001	4	1	4
4	32	11011	5	1	5
5	38	10001	5	1	5
Total					25

4. KESIMPULAN

Kesimpulan yang dapat diambil setelah melakukan perancangan aplikasi kamus besar bahasa indonesia (KBBI) dengan menerapkan algoritma punctured Elias code dimana prosedur untuk mengkompresi kamus besar bahasa indonesia (KBBI) dimulai dari memilih kamus besar bahasa indonesia yang akan dikompresi, lalu buku tersebut akan dikompresi dengan algoritma punctured Elias code sehingga menghasilkan sebuah buku terkompresi yang memiliki ukuran lebih kecil. Penerapan algoritma punctured Elias code untuk kompresi kamus besar bahasa indonesia (KBBI) dilakukan dengan membaca nilai heksadesimal dari suatu kamus besar bahasa indonesia (KBBI), lalu merubah nilai heksa tersebut menjadi nilai bit baru yang berupa bilangan biner, kemudian menyusun kembali nilai biner tersebut menjadi karakter baru. Dimana hasil komresi berdasarkan rasionya sebesar 32% berdasarkan sampel data 80 bit dengan hasil kompresi 25 Bit.

REFERENCES

- [1] R. D. Pratiwi, S. D. Nasution, and Fadlina, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algoritma Fixed Length Binary Encoding (Flbe)," *Media Inform. Budidarma*, vol. 2, no. 1, pp. 10–14, 2018.
- [2] M. R. Iriansyah, S. D. Nasution, and K. Ulfa, "Penerapan Metode Deflate Dan Algoritma Goldbach Codes Dalam Kompresi File Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 1, no. 1, pp. 186–189, 2017.
- [3] N. Dengen, "Perancangan Sistem Informasi Terpadu Pemerintah Daerah Kabupaten Paser," vol. 4, no. 1, pp. 47–54, 2009.
- [4] Marjito and G. Tesaria, "APLIKASI PENJUALAN ONLINE BERBASIS ANDROID (STUDI KASUS: DI TOKO HOAX MERCH)," *J. Comput. Bisnis*, vol. 10, no. 1, pp. 40–49, 2016.
- [5] D. Salomon and G. Motta, *Handbook of Data Compression*. Springer, 2010.
- [6] A. Muhammad dan A. Soeb, "Penarapan Algoritma Prefix Code Dalam Kompresi File Video," dalam *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, Medan, 2021.
- [7] A. Nur dan A. Soeb, "Penerapan Algoritma Elias Omega Code Pada Kompresi File Audio Aplikasi Murottal Muzzamil Hasbalah," *Pelita Informatika: Informasi dan Informatika*, vol. 9, no. 2, pp. 113-119, 2020.

