

Penerapan Algoritma Inverted Elias Delta Untuk Kompresi Konten Pada Aplikasi Psikologi Berbasis Android

Krisna Monita R Hutahaean

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: khrisnahutahaean@gmail.com

Abstrak—Dalam era teknologi yang semakin canggih seperti saat ini, tingginya aktivitas penyimpanan dan pengiriman data menimbulkan suatu masalah dimana diperlukannya tempat penyimpanan yang besar. Ukuran data yang semakin besar dapat mengakibatkan pemborosan ruang penyimpanan dan lambatnya proses pengiriman data secara *online* ataupun pengirimian secara *offline*. Penyimpanan data yang semakin lama akan mengakibatkan menumpuknya data—data yang semakin banyak pula. Penumpukan data ini menjadi masalah terutama dari segi ketersediannya ruang penyimpanan. Seiring perkembangan teknologi saat ini maka semakin berkembangnya kemampuan dalam mengolah data yaitu dengan memanfaatkan metode kompresi yang diharapkan dapat menjawab masalah tersebut. Algoritma *Inverted Elias Delta* digunakan untuk mengkompresi *record database* pada aplikasi Psikologi agar memperkecil ukuran *bit* dari suatu karakter. Sehingga dengan menerapkan algoritma *Inverted Elias Delta* untuk mengkompresi *record database* dapat mengurangi kapasitas penyimpanan secara optimal pada *smartphone* (*handphone*).

Kata Kunci: Kompres; Record Database; Inverted Elias Delta

Abstract—In an era of increasingly sophisticated technology like today, the high activity of data storage and transmission creates a problem where a large storage space is required. The larger data size can lead to wasteful storage space and slow data transmission process online or offline. Data storage that is getting longer will result in piling up more and more data. This data accumulation is a problem, especially in terms of availability of storage space. Along with current technological developments, the ability to process data is increasingly developing, namely by utilizing compression methods which are expected to answer these problems. The Inverted Elias Delta algorithm is used to compress database records in the Psychology application in order to reduce the bit size of a character. So, applying the Inverted Elias Delta algorithm to compress the record database can reduce storage capacity optimally on a smartphone (*handphone*).

Keywords: Compression; Record Database; Inverted Elias Delta

1. PENDAHULUAN

Pada perkembangan teknologi zaman sekarang ini sangat berperan dalam kehidupan sehari-hari. Sehingga kebutuhan data digital yang semakin meningkat. Dengan menggunakan teknologi informasi dapat membantu dan mempermudah dalam melakukan suatu pengolahan data dan informasi dengan menggunakan alat bantu *smartphone* (*handphone*). Kemajuan teknologi *smartphone* (*handphone*) kian hari semakin pesat, tidak hanya untuk komunikasi tetapi manfaat *smartphone* (*handphone*) juga telah menjadi sebuah alat yang mendukung kinerja dan berbagai aktifitas manusia. Salah satunya adalah aplikasi psikologi.

Aplikasi psikologi dibutuhkan diberbagai ilmu pengetahuan untuk mengerti dan memahami kejiwaan seseorang. Psikologi juga merupakan salah satu bidang ilmu pengetahuan yang mempelajari tentang perilaku, fungsi mental dan proses mental manusia melalui prosedur ilmiah. Tetapi tidak semua ahli psikologi dan mahasiswa psikologi memiliki buku psikologi dikarenakan buku yang tersedia terbatas dan harganya yang mahal, bukan hanya itu saja namun ketika mahasiswa membawa buku maka hal itu akan sangat mempersulitkan mahasiswa karena membawa buku psikologi tersebut, sehingga mereka membutuhkan aplikasi psikologi di *smartphone* (*handphone*) sebagai alat penunjang pembelajaran di rumah atau sebagai referensi untuk mahasiswa. Tidak hanya di lingkungan mahasiswa psikolog namun di lingkungan lain juga diperlukan aplikasi psikologi untuk jurusan tertentu. Masalah yang ada pada aplikasi psikologi adalah tentang terbatasnya kapasitas penyimpanan atau yang dikenal dengan memori penuh, terbatasnya kapasitas kosong pada media penyimpanan digital yang menumpuk mengakibatkan penuhnya ruang kosong didalam media penyimpanan digital. Salah satu cara untuk memaksimalkan ruang kosong pada media penyimpanan digital adalah dengan mengalihkan tempat penyimpanan internal ke *memori card*. Namun tidak semua orang memiliki memori card atau ingin menggunakannya. Aplikasi yang berukuran besar tentu akan membutuhkan ruang yang besar, biasanya memori akan penuh bila kita banyak mengunduh aplikasi atau banyaknya file dalam *smartphone*. Oleh karena itu dilakukan kompresi konten pada aplikasi psikologi untuk memperkecil ukurannya. Karena *smartphone* yang memiliki kapasitas penyimpanan kecil juga perlu ruang penyimpanan agar dapat menginstal atau menggunakan aplikasi psikologi.

Pada tahun 2017 Muhammad Rio Irlansyah, dkk meneliti tentang kompresi file teks dengan menerapkan metode *Deflate* dan algoritma *Goldbach Code*. Hasil penelitian yang dilakukan dengan cara mengkompresi tiap karakter yang terdapat pada file teks berdasarkan cara kerja tiap algoritma dan menghasilkan file dengan ukuran lebih kecil dari sebelum dilakukan proses kompresi[1]. Kemudian penelitian yang diteliti oleh Nadhira Ayuningtyas tahun 2007 telah berhasil membuktikan bahwa kompresi file teks dalam layanan SMS dengan menerapkan algoritma Huffman dapat melakukan kompresi pesan hingga tiga kali lebih kecil dengan pembuatan pohon Huffman untuk karakter yang terdapat dalam pesan[2].

Ada banyak algoritma yang bisa digunakan didalam kompresi *file*, diantaranya yaitu *Huffman*, *Lempel Ziv Welch* (LZW), *Dynamic Markov Compression* (DMC), *Arithmetic coding*, *Run Length Encoding* (RLE) dan algoritma lainnya. Dari sekian banyak algoritma yang bisa digunakan dalam kompresi *file*, penulis memilih algoritma *Inverted Elias Delta* untuk diangkat sebagai topik penelitian guna untuk mengetahui bagaimana kinerja kompresi apabila diterapkan kedalam sebuah aplikasi penyimpanan *file*. Dengan menerapkan algoritma *invertedelias delta* untuk mengkompresi *konten* pada aplikasi psikologi sehingga memori yang digunakan untuk menyimpan *database* aplikasi psikologi ini menjadi lebih kecil dan dapat berjalan secara optimal pada *smartphone* (*handphone*).

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi data dalam bidang ilmu komputer, ilmu pengetahuan dan seni adalah sebuah penyajian informasi ke dalam bentuk yang lebih sederhana. Kompresi data atau *source coding* dalam ilmu komputer dan teori informasi adalah proses meng-*encode* informasi dengan menggunakan lebih sedikit bit dari suatu sumber yang belum di-*encode* melalui penggunaan skema pengkodean yang spesifik. Kompresi data dapat diartikan juga sebagai proses yang dapat mengubah sebuah aliran data masukan (sumber atau data asli) ke dalam aliran data yang lain (keluaran atau data yang dimampatkan) yang memiliki ukuran yang lebih kecil. Kompresi data sudah ada sejak puluhan tahun yang lalu. Kompresi data menjadi salah satu cabang teori informasi karena kompresi data berkecimpung dengan masalah *redundancy* dalam suatu informasi[4].

2.2 Algoritma Inverted Elias Delta

Inverted Elias Delta adalah kebalikan dari *Elias Delta Code*. *Elias Delta Code* adalah sebuah algoritma kompresi yang dibuat oleh Peter Elias menggunakan kode yang telah di buat sebelumnya, yaitu *Elias Gamma Code*[6].

Dari pengertian algoritma *Inverted Elias Delta* di atas, dapat penulis simpulkan bahwa *Inverted Elias Delta* adalah algoritma yang dikembangkan dari algoritma sebelumnya yaitu algoritma *Elias Delta Code* yang diciptakan oleh Peter Elias yang dimana perhitungannya yaitu, kebalikan dari *Elias Delta Code*. Pada *Elias Gamma Code*, ada tambahan kode dalam *unary* (α). Dalam kode berikutnya δ (delta), ditambahkan pada panjang kode dalam biner (β). Maka *Elias Delta Code*, yang juga untuk bilangan bulat positif, sedikit lebih kompleks. Untuk membangun kode *Inverted Elias Delta*, lakukan pengkodean *Elias Delta Code* terlebih dahulu, kemudian lakukan pembalikan kode dari hasil pengkodean *Elias Delta Code* tersebut. Misalnya, 0 dibalikkan jadi 1 atau 1 dibalikkan jadi 0 sehingga kode *Inverted Elias Delta* dapat dibuat dari hasil pembalikan kode *Elias Delta Code* tersebut.

Aturan untuk mengkodekan sebuah bilangan dengan menggunakan *Elias Delta Code*, sebagai berikut:

1. Tuliskan n dalam bentuk bilangan biner. Biner n diumpamakan dengan b .
2. Hitung jumlah bit pada b nya, kemudian tuliskan dalam bentuk bilangan biner. Jumlah bit pada b diumpamakan dengan M .
3. Tuliskan M dalam bentuk bilangan biner. Biner M diumpamakan dengan Mb .
4. hapus satu bit yang paling kiri dari biner n atau yang disebut dengan b , kemudian tuliskan b yang sudah dihapus satu bit dari sebelah kiri tersebut. b yang sudah dihapus satu bit dari sebelah kiri tersebut diumpamakan dengan L .
5. Kemudian lakukan perhitungan dengan menggunakan rumus di bawah ini[6]:

$$\Delta = (\text{length}(Mb) - 1) * "0" + Mb + L \quad (1)$$

Keterangan :

b = decimal to biner (n)

M = len (b)

Mb = decimal to biner (M)

Len (Mb) = jumlah digit (Mb)

L = b yang sudah dihapus satu digit nilai biner yang paling kiri

Misalnya untuk mengetahui kode *Inverted Elias Delta* dari $n=20$, maka dapat ditentukan dengan :

$b = 10100$

$M = 5$

$Mb = 101$

$L = 0100$

Elias Delta Code = $(\text{len}(Mb) - 1) * "0" + Mb + L$

$= (3 - 1) * "0" + "101" + "0100"$

$= 1 * "0" + "101" + "0100"$

$= 01010100$ *Inverted Elias Delta* = 10101011

Dari perhitungan diatas, dapat kita ketahui bahwa kode *Inverted Elias Delta* dari $n=20$ adalah "10101011".

3. HASIL DAN PEMBAHASAN

Berdasarkan penelitian ini, akan dilakukan analisa mengkompresi konten pada aplikasi psikologi dengan menggunakan algoritma *Inverted Elias delta*. Tetapi terlebih dahulu akan dianalisa tentang konten pada aplikasi psikologi. Kebutuhan manusia akan aplikasi untuk membantu dalam kegiatan belajar atau kegiatan sehari-hari serta lain sebagainya membuat manusia lupa bahwa aplikasi yang *download*, memiliki kapasitas yang berbeda-beda. Hal ini membuat ruang penyimpanan pada *smartphone* menjadi cepat penuh. Aplikasi Psikologi memiliki banyak teks yang disimpan kedalam *record database* sehingga semakin banyak teks yang disimpan, maka semakin besar pula kapasitas *database* yang diperlukan atau yang digunakan.

Algoritma *Inverted Elias Delta* sendiri merupakan algoritma kompresi yang akan merubah ukuran suatu teks menjadi ukuran lebih kecil dari ukuran aslinya. Model kompresi teks dengan menggunakan algoritma *Inverted Elias Delta* yaitu, dengan membangun biner *Inverted Elias Delta* dulu, kemudian mengganti biner teks yang akan dikompres menjadi biner dari *Inverted Elias Delta* yang telah dibangun dengan catatan data pada teks yang akan dikompres tersebut sudah diurutkan secara descending berdasarkan frekuensinya.

Dalam hal ini penerapan algoritma *Inverted Elias Delta* dilakukan terhadap *record database* pada aplikasi Psikologi yang memiliki keterbatasan memori sehingga menggunakan memori yang relatif besar. Kompresi *record database* pada aplikasi psikologi merupakan teknik sehingga memperkecil kapasitas penyimpanan aplikasi dan dapat digunakan pada *smartphone (handphone)* dengan memori yang kecil. Analisa yang dilakukan dari proses cara kerja algoritma *Inverted Elias Delta* mengarah pada file teks isi aplikasi psikologi kemudian proses kompresi menghasilkan file teks isi record terkompresi, tersimpan di database, selanjutnya file teks isi aplikasi psikologi di dekompresi kembali ke bentuk semula dan di buka melalui aplikasi psikologi yang di install di *smarthphone*.

Adapun tujuan dari analisa terhadap sistem yang akan dibangun yaitu untuk mengetahui dan merumuskan kebutuhan dari sistem serta membantu meminimalisir sumber daya yang berlebih.

3.1 Penerapan Algoritma *Inverted Elias Delta*

Aplikasi psikologi membutuhkan kapasitas penyimpanan yang besar apabila melakukan instalasi pada *smartphone(handphone)*. Dengan melakukan kompresi *record database*, aplikasi psikologi di instal pada *smartphone(handphone)* yang memerlukan kapasitas penyimpanan memori yang besar akan dikompresi sehingga mengurangi jumlah penyimpanan memori dan dapat melakukan instalasi di memori yang kecil. Dalam menganalisa isi konten harus dilakukan pengambilan *sample* yaitu isi konten dari aplikasi psikologi. Berikut adalah file teks isi *record*.

Tabel 1. File Teks Isi *Record* Aplikasi Psikologi

No	Istilah Psikologi	Isi Konten Psikologi
1	Pengertian psikologi	Menurut Crow & crow Mengemukakan pendapatnya tentang pengertian psikologi adalah tingkah laku manusia yaitu interaksi manusia dengan dunia sekitarnya, baik berupa manusia lain maupun bukan manusia, hewan iklim, kebudayaan dan sebagainya.
2	Cabang psikologi	Ruang lingkup psikologi Psikologi umum merupakan suatu keilmuan mengenai tingkah laku individu secara umum. Hal ini mencakup semua fase perkembangan psikologi manusia serta mencakup segala tingkatan usia dan jenis kelamin.
3	Istilah-istilah psikologi	Consious mind (alam sadar/pikiran sadar): Apa yang anda sadari pada saat tertentu, seperti penginderaan langsung, ingatan ingatan perpepsi, pemikiran, fantasi serta perasaan yang anda miliki.
4	Pengertian perilaku	Pengertian perilaku adalah tindakan atau aktivitas dari manusia itu sendiri yang mempunyai bentangan arti yang sangat luas antara lain: berjalan, berbicara, menangis, tertawa, bekerja, kuliah, menulis, membaca, dan sebagainya.
5	Bentuk perilaku	Pada dasarnya bentuk perilaku dapat diamati, melalui sikap dan tindakan, namun demikian tidak berarti bahwa bentuk perilaku itu hanya dapat dilihat dari sikap dan tindakannya saja, perilaku dapat pula bersifat potensial, yakni dalam bentuk pengetahuan, motivasi.
6	Perilaku sadar	Perilaku sadar merupakan perilaku yang melalui kerja otak dan pusat susunan.
7	Perilaku kognitif	Perilaku kognitif adalah perilaku yang mencakup kegiatan mental (otak). Segala upaya yang menyangkut aktivitas otak adalah termasuk dalam perilaku kognitif.
8	Lingkar motivasi	Pada umumnya motivasi mempunyai sifat siklus(melingkar), yaitu motivasi timbul, memicu perilaku tertuju kepada tujuan (goal), dan akhirnya setelah tujuan tercapai motivasi itu berhenti.
9	Fustasi dan konflik	Dalam rangka individu mencapai tujuan kadang-kadang atau justru sering individu menghadapi kendala, sehingga kemungkinan tujuan tersebut tidak dapat tercapai.

No	Istilah Psikologi	Isi Konten Psikologi
10	Pengertian kepribadian	Keprbadian atau personality merupakan sifat dan tingkah laku yang membedakannya dengan orang lain. Kepribadian seseorang dibentuk dan terbentuk oleh factor internal dan eksternal.

Berikut adalah langkah untuk mengkompresi dan mendekompresi *record database* pada aplikasi psikologi. Analisa proses kompresi *record database* pada aplikasi psikologi. Langkah pertama yang dilakukan adalah dengan memasukkan file teks isi *record* yang diambil sebagai contoh dari tabel ilmu psikologi “**PERILAKU SADAR MERUPAKAN PERILAKU YANG MELALUI KERJA OTAK DAN PUSAT SUSUNAN SARAF**”. Harus di ketahui terlebih dahulu bawah *string* “PERILAKU SADAR MERUPAKAN PERILAKU YANG MELALUI KERJA OTAK DAN PUSAT SUSUNAN SARAF” terdiri dari 80 karakter, dimana 1 karakter tersebut sama dengan (=) 1 byte dan 1 byte sama dengan (=) 8 bit, maka total *string* sebelum dikompresi adalah 80 byte atau sama dengan (=) 648 bit. Berikut adalah contoh teks yang akan dikompresi dan dekompresi.

Dari input *string* untuk mengetahui ukuran dari *string* “PERILAKU SADAR MERUPAKAN PERILAKU YANG MELALUI KERJA OTAK DAN PUSAT SUSUNAN SARAF”, dapat dilihat pada tabel dibawah ini:

Tabel 2. FrekuensiData

Karakter	Frekuensi
P	4
E	5
R	6
I	3
L	4
A	15
K	5
U	7
SP	11
S	5
D	2
M	2
N	5
Y	1
G	1
J	1
O	1
T	2
F	1
Total	80

Berdasarkan pada tabel di atas, didapatkan beberapa karakter yang sama. Sebelum proses kompresi *record database*, langkah awal adalah membaca karakter Psikologi kemudian membuat tabel nilai karakter yang diurutkan dari nilai frekuensi terbesar (karakter yang sama) ke terkecil. Urutan karakter dapat dilihat pada tabel di bawah ini

Tabel 3. Ukuran String Sebelum Dikompres

Char	ASCII Code	ASCII Biner	Bit	Frekuensi	Bit x Frekuensi
P	80	1010000	8	4	32
E	69	1000101	8	5	40
R	82	1010010	8	6	48
I	73	1001001	8	3	24
L	76	1001100	8	4	32
A	65	1000001	8	15	120
K	75	1001011	8	5	40
U	85	1010101	8	7	56
SP	32	100000	8	11	88
S	83	1010011	8	5	40
D	68	1000100	8	2	16
M	77	1001101	8	2	16

Berdasarkan tabel 3, satu karakter bernilai 8 bit bilangan biner. Sehingga 80 karakter mempunyai nilai biner sebanyak 648 bit. Sebelum melakukan proses kompresi karakter tersebut diurutkan terlebih dahulu secara descending. Pengurutan secara descending dapat di lihat pada tabel 4.

Tabel 4. Ukuran String Sebelum Dikompresi

Char	ASCII Code	ASCII Biner	Bit	Frekuensi	Bit x Frekuensi
A	65	1000001	8	15	120
SP	32	100000	8	11	88
U	85	1010101	8	7	56
R	82	1010010	8	6	48
E	69	1000101	8	5	40
K	75	1001011	8	5	40
S	83	1010011	8	5	40
N	78	1001110	8	5	40
L	76	1001100	8	4	32
P	80	1010000	8	4	32
I	73	1001001	8	3	24
D	68	1000100	8	2	16
M	77	1001101	8	2	16
T	84	1010100	8	2	16
F	70	1000110	8	1	8
J	74	1001010	8	1	8
O	79	1001111	8	1	8
Y	89	1011001	8	1	8
G	71	1000111	8	1	8
JUMLAH				80	648

Untuk membangun kode Inverted Elias Delta selalu diumpamakan dengan $n=1, 2, 3$. Untuk $n \geq 1$, gunakan rumus di bawah ini.

$$\Delta = (\text{length}(Mb) - 1) * "0" + Mb + L \quad (1)$$

$$n = 1, \quad b = 1, \quad M = 1, \quad Mb = 1, \quad L = -$$

$$\begin{aligned} \text{Elias Delta Code} &= (\text{len}(Mb) - 1) * "0" + Mb + L \\ &= (1 - 1) * "0" + "1" \\ &= "0" + "1" \\ &= 01 \end{aligned}$$

$$\text{Inverted Elias Delta} = 10$$

$$n = 2, \quad b = 10, \quad M = 2, \quad Mb = 10, \quad L = 0$$

$$\begin{aligned} \text{Elias Delta Code} &= (\text{len}(Mb) - 1) * "0" + Mb + L \\ &= (2 - 1) * "0" + "10" + "0" \\ &= "0" + "10" + "0" \\ &= 0100 \end{aligned}$$

$$\text{Inverted Elias Delta} = 1011$$

$$n = 3, \quad b = 11, \quad M = 2, \quad Mb = 10, \quad L = 1$$

$$\begin{aligned} \text{Elias Delta Code} &= (\text{len}(Mb) - 1) * "0" + Mb + L \\ &= (2 - 1) * "0" + "10" + "1" \\ &= "0" + "10" + "1" \\ &= 0101 \end{aligned}$$

$$\text{Inverted Elias Delta} = 1010$$

Tabel 5. Kode Elias Delta Code dan Inverted Elias Delta

N	Elias Delta Code	Inverted Elias Delta
1	01	10
2	0100	1011
3	0101	1010
4	01100	10011
5	01101	10010
6	01110	10001
7	01111	10000
8	0100000	1011111
9	0100001	1011110
10	0100010	1011101
11	0100011	1011100
12	0100100	1011011

N	Elias Delta Code	Inverted Elias Delta
13	0100101	1011010
14	0100110	1011001
15	0100111	1011000
16	01010000	10101111
17	01010001	10101110
18	01010010	10101101
19	01010011	10101100
20	01010100	10101011

Setelah dilakukan pengurutan secara descending, selanjutnya masuk ketahap proses kompresi, dapat dilihat pada tabel 6. dibawah ini.

Tabel 6. Ukuran String Setelah Dikompres Menggunakan *Inverted Elias Delta*

n	Char	Frekuensi	Inverted Elias Delta	Bit	Bit x Frekuensi
1	A	15	10	2	30
2	SP	11	1011	4	44
3	U	7	1010	4	28
4	R	6	10011	5	30
5	E	5	10010	5	25
6	K	5	10001	5	25
7	S	5	10000	5	25
8	N	5	1011111	7	35
9	L	4	1011110	7	28
10	P	4	1011101	7	28
11	I	3	1011100	7	21
12	D	2	1011011	7	14
13	M	2	1011010	7	14
14	T	2	1011001	7	14
15	F	1	1011000	7	7
17	O	1	10101110	8	8
18	Y	1	10101101	8	8
19	G	1	10101100	8	8
			JUMLAH		400

Dari Tabel 6.. dapat di bentuk string bit dari string sebelum dikompresi yaitu, “PERILAKU SADAR MERUPAKAN PERILAKU YANG MELALUI KERJA OTAK DAN PUSAT SUSUNAN SARAF”, menjadi string bit :

“101110110010100111011100101111010100011010101110000101011011110100111011101101010011100111
01010111011010000101011111011101110110010100111011100101111010100011010101110101101101011
1111010110010111011010100101011110101011110101011100101110001100101001110101111101011101
0111010110011010001101110110111101111101110111011010100001010110011011100001010100001010
10111111010111111011100001010011101011000”

Sebelum di dapatkan hasil akhir kompresi, dilakukan penambahan string bit itu sendiri yaitu *padding bit* dan *flag bit*. Penambahan bit (*padding*) dilakukan jika hasil kompresi dibagi 8 memiliki sisa. Sedangkan hasil kompresi tidak memiliki sisa atau nol maka tidak perlu adanya penambahan bit (*padding*). Sedangkan *flag bit* adalah nilai biner yang didapat dari nilai hasil *padding*. *Flag bit* memiliki jumlah nilai 8 bit biner.

Berdasarkan hasil kompresi nilai *biner* pada teks menggunakan algoritma *Inverted Elias Delta* di dapat nilai string bit sebanyak 400 bit dimana jika bilangan tersebut dibagi 8 Jumlah bit hasil kompresi tidak memiliki sisa. Maka tidak perlu ditambah lagi *padding* dan *flag bit*, karena bit hasil kompresi habis dibagi 8. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti gambar dibawah ini.

Tabel 7. Pengelompokan Bit

10111011	01001010
00101001	11110101
11011100	01111010
10111101	10101110
01000110	01011100
10101110	01100101
00010101	00111010
10111010	11111010

01110111	11101011
01101010	10101100
01110011	11010001
10101011	10111011
10110100	01110101
00101011	11111011
11110111	10111011
01110110	01010000
01010011	10101100
10111001	11011100
01111010	00101010
10001101	00010101
01011101	01111110
01101101	10111111
01111110	01110000
10110010	10100111
11101101	01011000

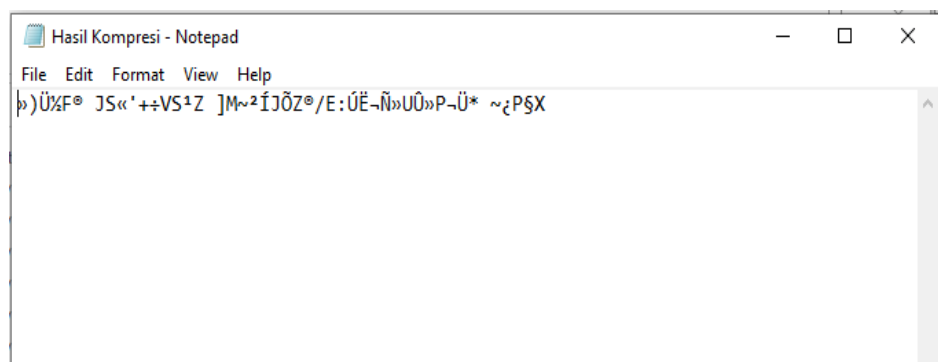
Berdasarkan pada pembagian kelompok nilai biner, didapatkan 16 kelompok nilai biner baru yang sudah terkompresi. Setelah pembagian dilakukan, maka nilai yang sudah dibagi dirubah kedalam suatu karakter dengan terlebih dahulu mencari nilai desimal dari *string bit* tersebut menggunakan kode ASCII untuk mengetahui nilai yang sudah terkompresi. Adapun simbol yang sudah terkompresi dapat dilihat pada tabel di bawah ini:

Tabel 8. Hasil Kompresi yang Sudah Menjadi karakter

Biner	Nilai Desimal Terkompresi	Karakter
10111011	187	»
00101001	41	)
11011100	220	Ü
10111101	189	½
01000110	70	F
10101110	174	®
00010101	21	
01101010	106	J
01110011	115	S
10101011	171	«
10110100	180	'
00101011	43	+
11110111	247	÷
01110110	118	V
01010011	83	S
10111001	185	¹
01111010	122	Z
10001101	141	
01011101	93	]
01101101	109	M
01111110	126	~
10110010	178	²
11101101	237	Í
01001010	74	J
11110101	245	Õ
01111010	122	Z
10101110	174	®
01011100	92	\
01100101	101	E
00111010	58	:
11111010	250	Ú
11101011	235	Ë
10101100	172	¬
11010001	209	Ñ
10111011	187	»
01110101	117	U
11111011	251	Û

Biner	Nilai Desimal Terkompresi	Karakter
10111011	187	»
01010000	80	P
10101100	172	¬
11011100	220	Ü
00101010	42	*
00010101	21	
01111110	126	~
10111111	191	¿
01110000	112	P
10100111	167	§
01011000	88	X

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



Gambar 1. Karakter Hasil Kompresi

Dari hasil kompresi dengan algoritma Inverted Elias Delta diatas dapat dihitung kinerja dari metode tersebut menggunakan *Compression Ratio* (CR) yaitu:

Compression Ratio (CR)

$$C_R = \frac{UkuranDataSetelahDikompresi}{UkuranDataSebelumDikompresi} \times 100\%$$

$$C_R = \frac{400}{648} \times 100\%$$

$$C_R = 62\%$$

Dari hasil perhitungan *compression ratio* dapat disimpulkan bahwa setelah dikompres ukuran file adalah 62% dari data sebelum di kompres. 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. Bit Keseluruhan Hasil Kompresi

Karkter	Nilai Desimal Terkompresi	Biner
»	187	10111011
)	41	00101001
Ü	220	11011100
½	189	10111101
F	70	01000110
®	174	10101110
	21	00010101
°	186	10111010
W	119	01110111
J	106	01101010
S	115	01110011
«	171	10101011
'	180	10110100
+	43	00101011
÷	247	11110111
V	118	01110110

Karakter	Nilai Desimal Terkompresi	Biner
S	83	01010011
ı	185	10111001
Z	122	01111010
	141	10001101
]	93	01011101
M	109	01101101
~	126	01111110
2	178	10110010
Í	237	11101101
J	74	01001010
Õ	245	11110101
Z	122	01111010
®	174	10101110
\	92	01011100
e	101	01100101
:	58	00111010
ú	250	11111010
ë	235	11101011
¬	172	10101100
Ñ	209	11010001
»	187	10111011
u	117	01110101
û	251	11111011
»	187	10111011
P	80	01010000
¬	172	10101100
Ü	220	11011100
*	42	00101010
	21	00010101
~	126	01111110
ç	191	10111111
p	112	01110000
§	167	10100111
X	88	01011000

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi

“1111011101110110010100111011100101111010100011010101110101101101011111101011001011101101010010101111010101111010101110010111000110010100111010111110101110101110101100110100011011101101110101111101110111011101110111011010100001010110011011100001010100001010101111101011111011100001010011101011000”

Proses dekompresi selanjutnya adalah membaca nilai *padding* dan *flagbit* dari keseluruhan nilai bit. Kebetulan pada kasus ini keseluruhan nilai string bit tidak memiliki sisa bagi. Maka tidak ditemukan nilai *padding* dan *flag*, maka dilanjutkan dengan proses pembacaan nilai srtingbit dari bit pertama atau indeks ke-0 sampai dengan bit terakhir atau indeks ke-*n* dari biner yang telah digabungkan. Jika ditemukan bit yang sesuai dengan tabel 3.4 di atas maka ubah string yang sesuai.

Setelah nilai n ditentukan maka dilakukan pembacaan pada indeks ke-0="1" string 1 tidak ditemukan kesesuaian pada tabel, maka pembacaan akan beralih pada indeks selanjutnya yaitu, menambahkan indeks ke-0="1" dengan indeks ke-1="0", string menjadi "10", string tersebut juga tidak ditemukan kesesuaian pada tabel, kemudian ditambahkan lagi pada indeks ke-2="1", string menjadi "101", string tersebut juga tidak ditemukan kesesuaian pada tabel, berikutnya ditambahkan lagi indeks ke-3="1" string menjadi "1011" sehingga didapat kesesuaian karakter pertama yaitu karakter P, maka ditulis P ke dalam string yang baru. Kemudian pembacaan akan diulangi untuk mendapatkan 80 karakter selanjutnya.

Indeks ke-4="1" tidak ditemukan kesesuaian pada Tabel maka ditambahkan dengan indeks ke-5="0", string menjadi "10" string tersebut juga tidak ditemukan kesesuaian pada Tabel, kemudian ditambahkan lagi dengan indeks berikutnya sampai di indeks ke-8="0", string menjadi "10010" sehingga ditemukan kesesuaian dengan karakter kedua yaitu, karakter E maka ditambahkan karakter E ke dalam string yang telah dibuat tadi sehingga menjadi "PE" dan begitu seterusnya sampai indeks ke-n sehingga dihasilkan String semula yaitu "PERILAKU SADAR MERUPAKAN PERILAKU YANG MELALUI KERJA OTAK DAN PUSAT SUSUNAN SARAF"

4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan dari pembahasan sebelumnya. Adapun kesimpulan tersebut Berdasarkan prosedur dari kompresi konten pada aplikasi psikologi dengan menggunakan algoritma inverted *elias delta* dapat ditarik kesimpulan bahwa prosedur tersebut telah berhasil untuk melakukan proses kompresi *record database* pada aplikasi psikologi. Berdasarkan hasil penerapan algoritma Inverted Elias Delta, ukuran data sebelum dikompresi adalah 648 bit dan setelah di kompresi adalah 400bit. Dari hasil perhitungan *compression ratio* dapat disimpulkan bahwa setelah dikompres ukuran file adalah 62% dari data sebelum dikompres. Pada aplikasi psikologi dirancang dengan menggunakan Eclipse Juno dengan menerapkan algoritma *inverted elias delta* sehingga diharapkan memudahkan penulis dalam mengkompresi *record database* pada aplikasi psikologi.

REFERENCES

- [1] M. R. Irliansyah, S. D. Nasution, and K. Ulfa, "PENERAPAN METODE DEFLATE DAN ALGORITMA GOLDBACH CODES," vol. I, pp. 186–189, 2017.
- [2] N. Ayuningtyas, "Implementasi kode Huffman dalam aplikasi kompresi teks pada layanan SMS," *Jur. Tek. Inform. Inst. Teknol. Bandung. Bandung*, no. 13506048, 2007.
- [3] I. K. Gunarta, "Implementasi Pembelajaran Yoga Dalam Meningkatkan Konsentrasi Belajar Di Sekolah Dasar Negeri 1 Sumerta," *J. Penjaminan Mutu*, vol. 3, no. 2, p. 180, 2017, doi: 10.25078/jpm.v3i2.198.
- [4] A. A. Guci, M. Syahrizal, and P. B. Simangunsong, "Analisa dan Implementasi Kompresi Citra CT Scan Menggunakan Metode Delta Modulation," *Pelita Inform. Budi Darma*, vol. 17, no. April, pp. 114–116, 2018.
- [5] D. Salomon, *Handbook of Data Compression*. London: Springer-Verlag London Limited, 2010.
- [6] D. Salomon, *Data compression*, vol. 3rd Editio. 2004.
- [7] L. Marlina, A. Putera, U. Siahaan, H. Kurniawan, and I. Sulistianingsih, "Data Compression Using Elias Delta Code," pp. 210–217, 2017, doi: 10.23883/IJRTER.2017.3406.TEGS6.
- [8] J. T. Volume *et al.*, "KOMPRESI TEXT GZIP UNTUK OPTIMASI KONTEN WEB MENGGUNAKAN ARSITEKTUR WAP PROXY," vol. 3, pp. 1–7, 2018.
- [9] Y. Sugiyani *et al.*, "PERANCANGAN APLIKASI EDUKATIF BERBASIS MULTIMEDIA," vol. 1, no. September, pp. 55–59, 2014.
- [10] D. Andayati, "Sistem Pakar Dalam Bidang Psikologi," *J. Tek. Inform. Inst. Saint Teknol. Yogyakarta*, no. November, pp. 286–293, 2012.
- [11] F. Aulia, "Aplikasi Psikologi Positif dalam Konteks Sekolah," *Semin. Psikol. dan Kemanus.*, pp. 120–124, 2015.
- [12] R. A.S and M. Shalahuddin, *Rekayasa Perangkat Lunak*. Bandung: Informatika Bandung, 2015.
- [13] M. S. Alfa Satyaputra and S. K. Eva Maulina Arintonang, *JAVA for Beginners with eclipse 4.2 Juno*. Jakarta, 2012.
- [14] V. O. L. N. O. Maret, "Perancangan Aplikasi Client Untuk Jaringan VOIP (VOICE OVER INTERNET PROTOCOL) Berbasis Android," vol. 11, no. 1, 2018.
- [15] Ihsan and D. P. Utomo, "Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
- [16] S. R. Saragih and D. P. Utomo, "Penerapan Algoritma Prefix Code Dalam Kompresi Data Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
- [17] Lamsah and D. P. Utomo, "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.