TIN: Terapan Informatika Nusantara
Vol 2, No 7, Desember 2021, Hal 407-413
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Penerapan Algoritma Arithmatic Coding Pada Aplikasi Kamus Teknologi Informasi Berbasis Android

Riris Ariska

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: riskaa2909@gmail.com

Abstrak−Perkembangan teknologi semakin hari semakin cepat dan luas. Saat ini banyak bermunculan berbagai jenis operating system (os) baik untuk komputer maupun aplikasi handset. Kamus pada telepon seluler lebih praktis, karena pada saat ini pengguna mmbutuhkan banyak informasi dengan cepat dimana saja dan tanpa adanya batasan waktu. Kompresi file merupakan salah satu aspek penting dalam pengembangan teknologi informasi. Kecepatan pengiriman sangat dibutuhkan dan sangat bergantung kepada ukuran dari informasi tersebut, salah satunya adalah dengan melakukan pemampatan data. Jika proses kompresi berjalan dengan efektif, maka hasil file yang diperoleh dapat lebih kecil dari file aslinya. Efektif tidaknya sistem kompresi data tergantung dari pemodelan dan pengkodean yang digunakan. Berdasarkan masalah diatas peneliti membuat judul yaitu "Penerapan Algoritma Arithmetic Coding Pada Aplikasi Kamus Teknologi Berbasis Android". Yang mana diharapkan dengan adanya peenlitian ini dapat membantu perusahaan dalam menentukan keputusan yang lebih sistematis.

Kata Kunci: Kamus; Kompresi; Arithmetic Coding; Android

Abstract−The development of technology is getting faster and more widespread. Currently, there are many different types of operating systems (OS) for both computers and handset applications. Dictionaries on cell phones are more practical, because at this time users need a lot of information quickly anywhere and without time limits. File compression is one of the important aspects in the development of information technology. The speed of delivery is very much needed and depends on the size of the information, one of which is by compressing the data. If the compression process runs effectively, the resulting file can be smaller than the original file. The effectiveness of the data compression system depends on the modeling and coding used. Based on the problem above, the researcher made a title, namely "Application of the Arithmetic Coding Algorithm in Android-Based Technology Dictionary Applications". Which is expected with this research can help companies in determining more systematic decisions.

Keywords: Dictionary; Compression; Arithmetic Coding; Android

1. PENDAHULUAN

Kamus merupakan sebuah media yang dapat diartikan sebagai buku yang berisikan tentang arti suatu kata dari bahasa atau istilah asing. Misalkan untuk istilah teknologi informasi, dalam hal ini sanagt jelas bahwa kamus tersebut berisikan tentang arti dan kata dari istilah teknologi infromasi. Kamus teknologi informasi merupakan salah satu kamus yang diperlukan oleh masyarakat terutama para pengguna komputer. Kamus memiliki kelebihan dan kekurangan, kelebihan dalam hal jumlah kosa kata dan kekurangan dalam pencarian arti kata yang memakan waktu lama.

Seiring dengan perkembangan teknologi, Android bisa menjadi sebuah alternatif untuk pembuatan kamus teknologi informasi yang menarik. Android telah menyediakan banyak *tools Application Programming Interface* (API) untuk pengembangan aplikasi. Namun terkadang kapasitas media penyimpanan yang dimiliki tidak sebanding dengan data atau *file* yang akan disimpan. Oleh sebab itu adanya sebuah teknik kompresi data dari bidang ilmu komputer yang dapat mereduksi atau memperkecil ukuran data.

Kompresi data merupakan salah satu metode untuk memperkecil ruang penyimpanan data pada suatu media penyimpanan. Kompresi data bertujuan untuk mengurangi jumlah bit yang digunakan dalam penyimpanan ataupun pengiriman data. Selain itu kompresi data juga dapat membantu memperkecil yang ditransmisikan didalam suatu media jaringan, seperti internet sehingga memperkecil waktu transfer data. Penggunaan metode algoritma juga sangat dibutuhkan. Salah satu algoritma tersebut adalah algoritma *Arithmetic coding*. Metode *Arithmetic coding* merupakan salah satu metode kompresi lossless yang memakai teknik statistical modeling dengan cara mengkodekan suatu barisan karakter/pesan menjadi suatu bilangan tunggal.

Pada umumnya, algoritma tersebut melakukan penggantian satu atau lebih simbol input dengan kode tertentu. *Arithmetic coding* menggantikan satu deretan simbol input dengan sebuah bilangan *floating point.* Semakin panjang dan semakin kompleks pesan yang dikodekan, semakin banyak bit yang diperlukan untuk keperluan tersebut.Aplikasi teknologi informasi dengan berbasis Android ini dibuat karena kebutuhan akan informasi sangat penting dalam waktu yang sangat lama pula.

Pada penelitian terdahulu yang dilakukan oleh Petrus Santoso, 2001 yang berjudul Studi Kompresi Data dengan metode *Arithmetic Coding* menyimpulkan bahwa Algoritma *Arithmetic Coding* ini cukup baik dipakai untuk kompresi data dan akan lebih optimal dari *Huffman Coding* jika ada data/simbol yang mempunyai probabilitas besar (sebagai konsekuensi, jumlah banyak) [1].

Pada penelitian yang lain yang dilakukan oleh Hengki Tamando Sihotang, 2018 yang berjudul Perancangan dan Implementasi Algoritma Arithmetic Coding Untuk Aplikasi Kompresi Data Video dan Audio menyimpulkan bahwa proses kompresi menggunakan *Aritmetic Coding* dari aplikasi yang dibuat bersifat statis. Karena walaupun digunakan dan diujikan file *audio video* yang sama berulang kali, tetap memiliki hasil keluar dengan ukuran yang sama. Kecepatan proses tidak hanya bergantung dari jumlah *audio video* yang di proses, tetapi juga bergantung dari ukuran file yang akan dikompresi [2].

TIN: Terapan Informatika Nusantara
Vol 2, No 7, Desember 2021, Hal 407-413
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi data merupakan penerapan Teori Informasi yang merupakan cabang ilmu matematika. Dalam teori informasi digunakan istilah "*entropy*" yang diambil dari istilah ilmu Thermodinamika. Makna dari entropy menunjukkan ketidakteraturan suatu informasi. Semakin tinggi nilai entropy sebuah informasi (*message*) berarti semakin banyak simbul yang mucul dalam informasi tersebut.

Pada umumnya proses kompresi data meliputi pembacaan simbul. Jika proses kompresi berjalan dengan efektif, maka hasil file yang diperoleh dapat lebih kecil dari file aslinya. Efektif tidaknya sistem kompresi data tergantung dari pemodelan dan pengkodean yang digunakan. Model dan kode yang digunakan dalam kompresi data dibentuk berdasarkan nilai probabilitas tiap-tiap simbul [3].

2.2 Algoritma *Arihtmetic Coding (AAC)*

Arithmetic Coding adalah suatu bagian dari *Entropy Encoding* yang mengkonversi suatu data kedalam bentuk data yang lain dengan lebih sering menggunakan sedikit bit dan jarang menggunakan lebih banyak bit karakter. *Arithmetic Coding* menggantikan satu deretan simbol input dengan sebuah bilangan *floating point*. Semakin panjang dan semakin kompleks pesan yang dikodekan, semakin banyak bit yang diperlukan untuk keperluan tersebut. Output dari *arithmetic coding* ini adalah satu angka yang lebih kecil dari 1 dan lebih besar atau sama dengan 0. Angka ini secara unik dapat di-decode sehingga menghasilkan deretan simbol yang dipakai untuk menghasilkan angka tersebut. Untuk menghasilkan angka output tersebut, tiap simbol yang akan di-encode diberi satu set nilai probabilitas. Teknik pengkodean ini memisahkan pesan masukan ke dalam simbol dan menukar masing-masing simbol dengan suatu *floating point*. Dimana *arithmetic coding* mengkodekan seluruh pesan ke dalam suatu bilangan pecahan n dimana (0.0 = n < 1.0). Kemudian digunakan algoritma sebagai berikut [3] :

1. *Set Low = 0.0*
2. *Set High = 1.0*
3. *While (simbol input masih ada) do*
4. *Ambil simbol input*
5. *CR = high – low*
6. *High = low + CR*high_range (simbol)*
7. *Low = low + CR*low_range (simbol)*
8. *End While*
9. *Cetak Low*

Di sini '*Low*' adalah output dari proses *Arithmetic Coding*.

3. HASIL DAN PEMBAHASAN

Analisa dan perancangan bertujuan menganalisa kebutuhan pengembangan aplikasi kompresi dengan algoritma *Arithmetic Coding***.** Input yang diproses dalam aplikasi yang dirancang berupa *record database* dengan karakter yang berbeda agar memperkecil ukuran data, dalam perancangan hanya memberikan informasi *record database*, bukanlah *file*. Desain dan implementasi ini meliputi desain data, deskripsi sistem, desain proses dan implementasi desain dan semua yang diperlukan dalam aplikasi perancangan kompresi.

Dalam sistem media pengkompresian *record database* yang akan diimplementasikan dalam aplikasi adalah menggunakan algoritma *Arithmetic Coding*. Dengan membaca tiap karakter yang dimasukkan dan *record database* yang di masukkan. Tiap karakter lalu diproses hingga berurutan sampai karakter berikutnya yang menghasilakan karakter baru dan bila ditemukan kesamaan karakter, Sehingga dalam proses yang dikembangakan hanya menampilkan proses kerja algoritma *Arithmetic Coding* sebagai media pengkompresian agar pengguna dapat mengetahui bagaimana proses dari sebuah algoritma *Arithmetic Coding.*

Permasalahan yang dibahas adalah membuat suatu pengkodean karakter kompresi dengan menggunakan algoritma *Arithmetic Coding,* Berikut ini contoh pengaplikasian kompresi dan dekompresi dengan menggunakan algoritma A*rithmetic Coding*. Berikut merupakan prosedur kompresi dan dekompresi *record database* kamus teknologi informasi, Pada kamus akan diproses kompresi sehingga kamus teknologi informasi terkompresi jika sudah terkompresi maka simpan kedalam *database* sehingga *record* kamus terkompresi menggunakan algoritma *Arthmetic Coding*, setelah *record kamus* terkompresi maka aplikasi kamus teknologi informasi yang menggunakan algoritma *arithmetic coding* akan membaca *record database* kamus, setelah *record database* kamus dibaca maka akan melakukan proses dekompresi setelah dekompresi dilakukan maka akan menampilkan isi kamus.

3.1 Penerapan Algoritma *Arihtmetic Coding (AAC)*

Berdasarkan analisa, aplikasi Kamus memiliki kapasitas yang besar untuk di instal pada *smartphone* yang memiliki ruang penyimpanan yang kecil. Dengan melakukan kompresi data, Data yang berukuran besar akan dikompresi menjadi ukuran yang kecil dan akan mengurangi alokasi penyimpanan. Dalam menganalisa *record database* harus dilakukan mengambil

TIN: Terapan Informatika Nusantara
Vol 2, No 7, Desember 2021, Hal 407-413
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

sample *record database* untuk mendapatkan nilai dari hasil *String* tersebut. Berikut adalah contoh Sample *record database* Kamus.

Keterangan :

Login : Login disebut juga logan atau sign in adalah sistem keamanan komputer, yakni berupa proses masuk bagi pengguna untuk mengakses sistem komputer. Login/Logout untuk mengakses penuh keseluruhan halaman suatu domain jaringan.

Berikut adalah langkah untuk mengkompresi dan dekompresi *record database.* Contoh samplenya "JUGA LOGAN" sebelum diketahui ukuran *string*, maka dilakukan tahap pembentukan karakter terlebih dahulu yaitu karakter-karakter yang terdapat pada isi *record database*. Setelah karakter diperoleh maka frekuensi untuk kemunculan tiap karakter akan dihitung. Berikut adalah hasil untuk mengetahui ukuran *string* sebagai berikut :

Data *String* = JUGA LOGAN
Karakter = ∑ = {J, U ,G, A, *Space*, L, O, G, A, N}
Frekuensi Karakter =

J = 1	L = 1
U = 1	O = 1
G = 2	N = 1
A = 2	
Space = 1	

Setelah karakter dan frekuensi kemunculan tiap karakter, maka selanjutnya akan mengetahui pengukuran hasil karakter sebelum dikompresi. Bila dikodekan menggunakan kode *Arithmetic Coding*, langkahnya sebagai berikut:

1. Buat daftar frekuensi kemunculan tiap-tiap simbol

Tabel 1. Pendataan Simbol *Arithmetic Coding*

No	Karakter	Frekuensi
1	J	1
2	U	1
3	G	2
4	A	2
5	Space	1
6	L	1
7	O	1
8	N	1

2. Dihitung probabilitas frekuensi kemunculan setiap simbol seperti tabel 3.4 sebagai berikut :

Tabel 2. Probabilitas Frekuensi Kemunculan Setiap Simbol

No	Karakter	Frekuensi	Probabiltas
1	J	1	1/10
2	U	1	1/10
3	G	2	2/10
4	A	2	2/10
5	Space	1	1/10
6	L	1	1/10
7	O	1	1/10
8	N	1	1/10
Jumlah		10	1,0

3. Selanjutnya dihitung jangkauan setiap simbol, dengan titik terendah (*low*) adalah 0,0 dan titik tertinggi (*high*) adalah 1,0, sehingga jumlah dan probabilitas seluruh simbol sama dengan titik tertinggi dari jangkauan tersebut.

Tabel 3. Jangkauan Setiap Simbol

No	Karakter	Frekuensi	Probabilitas	Jangkauan	
				Low_Range	*High_Range*
1	J	1	0,1	0,9	1,0
2	U	1	0,1	0,8	0,9
3	G	2	0,2	0,6	0,8
4	A	2	0,2	0,4	0,6
5	Space	1	0,1	0,3	0,4
6	L	1	0,1	0,2	0,3
7	O	1	0,1	0,1	0,2
8	N	1	0,1	0,0	0,1

TIN: Terapan Informatika Nusantara
Vol 2, No 7, Desember 2021, Hal 407-413
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

No	Karakter	Frekuensi	Probabilitas	Jangkauan	
				Low_Range	*High_Range*
	Jumlah	10	1,0		

4. Diinisialisasi nilai titik terendah (*low*) sebagai 0 dan titik tertinggi (*high*) sebagai 1,0. Dengan menggunakan gabungan karakter sebanyak 2 karakter untuk setiap pemrosesan. Kemudian dilakukan perhitungan low dan high sesuai setiap simbol dengan rumus sebagai berikut:
   Set low = 0.0
   Set high = 1.0
   While (simbol *input* masih ada) *do*
   Ambil simbol input
   CodeRange = *high* – *low*
   High = *low* + *CodeRange* **high_range* (*symbol*)
   Low = *low* + *CodeRange* **low_range* (*symbol*)

Tabel 4. Kompresi *Arithmetic Coding*

Karakter		Low / High	
JU	Low	0,0 + (1.0 – 0,0) * 0,9	0,9
	High	0,0 + (1.0 – 0,0) * 1,0	1,0
GA	Low	0,9 + (1.0 – 0,9) * 0,8	0,98
	High	0,9 + (1.0 – 0,9) * 0,9	0,99
	Low	0,0 + (1.0 – 0,0) * 0,6	0,6
	High	0,0 + (1.0 – 0,0) * 0,8	0,8
	Low	0,6 + (0.8 – 0,6) * 0,4	0,68
	High	0,6 + (0.8 – 0,6) * 0,6	0,72
SpL	Low	0,0 + (1.0 – 0,0) * 0,3	0,3
	High	0,0 + (1.0 – 0,0) * 0,4	0,4
	Low	0,3 + (0,4 – 0,3) * 0,2	0,32
	High	0,3 + (0,4 – 0,3) * 0,3	0,33
ON	Low	0,0 + (1.0 – 0,0) * 0,1	0,1
	High	0,0 + (1.0 – 0,0) * 0,2	0,2
	Low	0,1 + (0,2 – 0,1) * 0,0	0,1
	High	0,1 + (0,2 – 0,1) * 0,1	0,11

5. Representasi dari simbol “JUGASpLON” yang telah dimampatkan diambil dari nilai biner yang berada diantara (0,98 – 0,99), (0,68 – 0,72), (0,32 – 0,33), (0,1 – 0,11). Nilai biner didapatkan dengan cara berikut

1. (0,98 - 0,99)

1	1,0		1,0		1,0		1,0		1,0		1,0
x	0,5	1	0,75	1	0,875	1	0,9375	1	0,96875	1	0,98438
0	0,0	0	0,5	0	0,75	0	0,875	0	0,9375	0	0,96875

X>=0,98	False	False	False	False	False	True
X<=0,99	True	True	True	True	True	True
	1	1	1	1	1	1

2. (0,68 – 0,72)

1	1,0		1,0		0,75		0,75		0,75		
		1		1		1		1			
x	0,5		0,75		0,625		0,6875		0,71875		
		0		0		0		0			
0	0,0		0,5		0,5		0,625		0,625		

X>=0,68	False	True	False	False	True
X<=0,72	True	False	True	True	True
	1	0	1	1	0

3. (0,32 – 0,33)

1	1,0	0,5	0,5	0,375	0,375	0,343750

TIN: Terapan Informatika Nusantara
Vol 2, No 7, Desember 2021, Hal 407-413
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

		1		1		1		1		1		
x	0,5		0,25		0,375		0,3125		0,343750		0,32813	
		0		0		0		0		0		
0	0,0		0		0,25		0,25		0,3125		0,3125	

X>=0,32	True	False	True	False	True	True
X<=0,33	False	True	False	True	False	True
	0	1	0	1	0	0

4. (0,1 – 0,11)

1	1,0		0,5		0,25		0,125		0,125		0,125	
		1		1		1		1		1		
x	0,5		0,25		0,125		0,0625		0,09375		0,10938	
		0		0		0		0		0		
0	0,0		0		0		0		0,0625		0,09375	

X>=0,1	False	False	False	False	False	True
X<=0,11	True	True	True	True	True	True
	0	0	0	1	1	0

Berdasarkan perhitungan yang telah dilakukan, maka dapat disimpulkan dalam bentuk tabel 3.7 dibawah ini :

Tabel 5. Karakter dan *Codeword*

Karakter	Codeword
JU	111111
GA	10110
SpL	010100
ON	000110

Berdasarkan pada tabel di atas dapat dibentuk nilai bit baru hasil kompresi dari susunan nilai hexadecimal sampel awal sebelum kompresi yaitu, JU, GA SpL, ON, menjadi nilai *bit* biner "111111101100101000001110". Kemudian sebelum di dapatkan hasil keseluruhan akhir kompresi dilakukan penambahan *stringbit* 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 sebnyak 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 23 tidak habis dibagi delapan dan sisanya 7 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 – 7 + "1"=**1***
Bit Akhir 9 – n
*Bit Akhir = 9 – 7 = 2 = **00000010***

11111110110010100000110**100000010**. Total panjang bit keseluruhan setelah ada penambahan bit adalah 23+1+8=32. Selanjutnya lakukan pemisahan bit menjadi beberapa kelompok. Setiap kelompok terdiri dari 8 bit seperti gambar di bawah ini.

Tabel 6. Pembagian Bit

11111110	11001010	00001101	00000010

Berdasarkan pada pembagian kelompok nilai biner, didapatkan 4 kelompok nilai biner baru yang sudah terkompresi beserta nilai biner penambahan bit.Setelah pembagian dilakukan, maka nilaiyang sudah dibagi dirubah kedalam suatu karakter dengan terlebih dahulumencari 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:

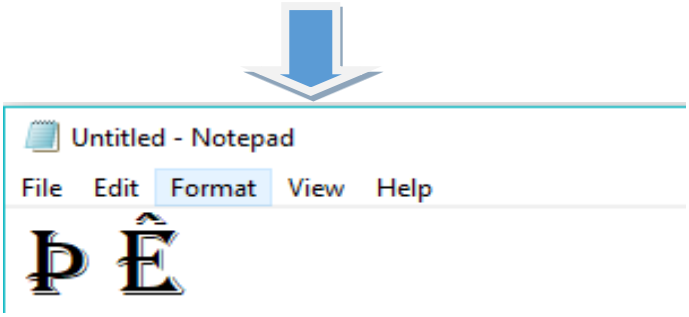
Tabel 7. Hasil Biner, Desimal dan Simbol

Biner	Nilai Desimal Terkompresi	Simbol
11111110	254	þ
11001010	202	Ê
00001101	13	
00000010	2	

TIN: Terapan Informatika Nusantara
Vol 2, No 7, Desember 2021, Hal 407-413
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Setelah diperoleh *string bit* maka, lakukan penggelompokkan biner-biner hasil kompresi dengan jumlah bit perkelompok adalah 4 kemudian konversi menjadi karakter.

11111110 11001010 00001101 00000010



Berdasarkan hasil kompresi dengan arithmetic coding diatas dapat dihitung kinerja kompresinya yaitu :

1. *Ratio of compression* (Rc)

$$Rc = \frac{Ukuran\ Data\ Sebelum\ Dikompresi}{Ukuran\ Data\ Setelah\ \ Dikompresi}$$

$$Rc = \frac{80}{32}$$

Rc= 2,5 % bit

2. *Compression Ratio* (Cr)

$$Cr = \frac{Ukuran\ Data\ Setelah\ Dikompresi}{Ukuran\ Data\ Sebelum\ \ Dikompresi}\ x\ 100\%$$

$$Cr = \frac{32}{80}\ x100\%$$

Cr = 40%

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 8. Hasil Simbol, Desimal dan Codeword

Simbol	Nilai Desimal	Nilai Biner
þ	254	11111110
Ê	201	11001010
	13	00000001
	2	00001000

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner dan digabungkan menjadi "11111110110010100000110**100000010**".

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+2 = 9. Penjelasan diatas menujukan bahwa bit akhir harus dihilangkan. Hasil pengembalian *binary* menjadi *string bit* semula dapat dilihat sebagai berikut ini:

"11111110110010100000110".

Berdasarkan perhitungan dengan algoritma *arithmetic coding string bit* pada diatas berjumlah 23 bit seperti diawal sehingga dilakukan pembacaan *string bit* awal. Adapun tabel hasil perhitungan diatas adalah sebagai berikut:

Tabel 9. Pengecekan Bit

Indeks	Nilai	Keterangan
1	1	Tidak Ada
2	11	Tidak ada
3	111	Tidak ada
4	1111	Tidak ada
5	11111	Tidak ada
6	111111	Ada pada tabel
7	1	Tidak Ada
8	10	Tidak ada
9	101	Tidak ada
10	1011	Tidak ada

TIN: Terapan Informatika Nusantara
Vol 2, No 7, Desember 2021, Hal 407-413
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin

Indeks	Nilai	Keterangan
11	10110	Ada pada tabel
12	0	Tidak Ada
13	01	Tidak ada
14	010	Tidak ada
15	0101	Tidak ada
16	01010	Tidak ada
17	010100	Ada pada tabel
18	0	Tidak ada
19	00	Tidak ada
20	000	Tidak ada
21	0001	Tidak ada
22	00011	Tidak ada
23	000110	Ada pada tabel

Maka dari penjabaran diatas dapat dibentuk tabel *aritmethic coding* dan nilai *hexa*awal.

Tabel 10. Tabel *artihmetic Coding*dan*Hexa*

n	Arithmetic Coding	Karakter
1	111111	JU
2	10110	GA
3	010100	SpL
4	000110	ON

Berdasarkan hasil dekompresi di atas didapati karakter sehingga dihasilkan *string* semula seperti tabel berikut :

Tabel 11. Karakter dan Codeword Dekompresi

Karakter	Codeword
JU	111111
GA	10110
SpL	010100
ON	000110

4. KESIMPULAN

Berdasarkan analisa dari hasil bab-bab sebelumnya, maka dapat ditarik sebuah kesimpulan, dimana dari kesimpulan-kesimpulan tersebut dapat berguna bagi pembaca, sehingga penulisan skripsi ini dapat lebih bermanfaat bagi. Adapun kesimpulan-kesimpulan yang dapat ditarik adalah Tahapan pengkompresian kamus dalam mengirim sebuah file yaitu dengan memilih file kamus yang akan dikompresi, kemudian file tersebut akan dikompresi lalu file hasil kompresi tersebut dapat mengurangi kapasitas yang tinggi menjadi lebih rendah. Penerapan algoritma *Arithmetic Coding* terhadap file yang akan dipilih berhasil mengurangi ukuran file dengan rasio 40% sehingga dapat membanu proses transmisi menjadi lebih cepat dan menghemat kuota internet.

REFERENCES

[1] Petrus Santoso, “Studi Kompresi Data dengan metode *Arithmetic Coding”*, Teknik Elektro, vol 1, No. 1, pp 14-18, Maret 2001
[2] Hengki Tamando Sihotang, “Perancangan dan Implementasi Algoritma Arithmetic Coding Untuk Aplikasi Kompresi Data Video dan Audio”, Mantik Penusa, vol 2, No 1, Juni 2018
[3] Eko Darwiyanto, ST. MT, Gia Septiana, S.Si., M.Sc, Yudistira Yoga Aji, "Analisis Perbandingan Kompresi Dan Dekompresi Menggunakan Algoritma Shannon Fano 2 Gram Dan Lempe Ziv Welch Pada Terjemahan Hadits Shahih Muslim," vol 3, No. 3, p. 5197, Desember 2016
[4] Ed Rosa A.S dan M. Shalahuddin, Rekayasa Perangkat Lunak, Informatika Bandung, Ed. Bandung 2014