



# Implementasi Metode Taboo Code Untuk Kompresi File Video

Saiful Simanjuntak

Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Budi Darma,  
Jalan. Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia  
Email : saifulsimanjuntak11@gmail.com

**Abstrak-**Video adalah teknologi untuk menangkap, merekam, memproses, dan menataulang gambar bergerak yang biasanya menggunakan sinyal elektronik, atau media digital. Misalnya, *file* video yang ukurannya besar mengakibatkan proses pengiriman semakin lama, serta menggunakan ruang memori yang besar dalam penyimpanannya. Permasalahan yang lahir karena besarnya ukuran *file* antara lain adalah borosnya media penyimpanan yang digunakan, serta memakan waktu lebih banyak dalam hal pertukaran data, transfer data melalui jaringan internet menjadi masalah besar di Indonesia karena kecepatan download dan upload yang sangat rendah. Untuk mengatasi masalah tersebut maka diperlukan teknik kompresi data. Konsep kompresi lahir sebagai solusi nyata dari permasalahan mengenai besarnya ruang yang dibutuhkan untuk menyimpan bermacam-macam *file* digital, tidak terkecuali dengan *file* video. Teknik kompresi merupakan teknik yang memadatkan isi *file* sehingga ukurannya menjadi lebih kecil dengan tetap menjaga kualitas representasi dari *file* tersebut.

**Kata Kunci:** Kompresi; File Video; Algoritma Taboo Code

**Abstrac-**Video is a technology for capturing, recording, processing, and rearranging images usually using electronic signals, or digital media. For example, a video file that is large in size causes the sending process to take longer, and uses a large amount of memory space in its storage. Problems that arise due to the large file size include the wasteful storage media used, as well as more time in terms of data exchange, data transfer via the internet is a big problem in Indonesia because download and upload speeds are very low. To overcome this problem, data compression techniques are needed. The concept of compression was born as a real solution to the problem of the amount of space needed to store various digital files, including video files. Compression technique is a technique that compresses files so that their size becomes smaller while maintaining the quality of the representation of the file.

**Keywords:** Compression; Video Files; Taboo Code Algorithm

## 1. PENDAHULUAN

Perkembangan teknologi yang begitu cepat seperti sekarang ini sangat bermanfaat guna bertukar informasi secara global baik itu dalam bentuk teks, gambar, audio ataupun video. Sehingga secara tidak langsung menimbulkan masalah yaitu membuat kebutuhan akan penyimpanan semakin meningkat. Dan bila mengirimkan data melalui media transmisi, maka akan memerlukan waktu yang semakin lama untuk proses pengiriman data tersebut. Untuk itu diperlukan suatu teknik yang dapat mereduksi besarnya ukuran suatu *file* yang disebut kompresi data. Kompresi data adalah 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 lebih kecil[1]. Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, dan mempercepat pengiriman data. Pada penelitian sebelumnya mengenai kompresi *file* video yang berjudul “Kompresi File Video MP4 Dengan Menggunakan Metode Discrete Cosine Transform” oleh Rizky Syahputra pada tahun 2016. Pada penelitian tersebut memiliki kekurangan yaitu metode discrete cosine transform tidak tahan terhadap perubahan suatu objek dikarenakan pesan mudah dihapus karena lokasi penyisipan data dan pembuatan data dengan metode DCT diketahui [2]. Penelitian lainnya yang berjudul “Perancangan Dan Implementasi Algoritma *Arithmetic Coding* Untuk Aplikasi Kompresi Data Video dan Audio” oleh Hengki Tomando Sihotang tahun 2018. Pada penelitian tersebut dinyatakan bahwa ukuran *file* sebelum kompresi sama dengan ukuran *file* sesudah didekompresi yang berarti bahwa tidak ada data yang hilang selama proses kompresi dan waktu yang diperlukan untuk kompresi *file* MP4. Algoritma *Arithmetic Coding* tidak optimal dalam melakukan kompresi audio dan video dapat dilihat pada rasio kompresi yang rendah dan grafik yang tidak stabil[3]. Pada penelitian ini algoritma yang digunakan adalah algoritma *Taboo Code*. Algoritma *Taboo Code* memiliki pendekatan terhadap panjangnya suatu *variable*. Penggagas Algoritma *Taboo Code* adalah Steven Pigeon. Prinsip dari Algoritma *taboo code* untuk memilih bilangan positif  $n$  dan membalikkan pola  $n$  untuk mengindikasikan akhir dari kode tersebut[3].

## 2. METODOLOGI PENELITIAN

### 2.1 Kompresi

Kompresi merupakan proses untuk menghilangkan berbagai kerumitan yang tidak penting (redundansi) dari suatu informasi dengan cara memadatkan isi file sehingga ukurannya menjadi lebih kecil dengan memaksimalkan kesederhanaannya dan tetap menjaga kualitas penggambaran dari informasi tersebut[4].

### 2.2 Rasio Kompresi (*Compression Ratio*)

*Compression Ratio* adalah presentasi besar data yang telah dikompresi yang didapat dari hasil perbandingan antara ukuran data setelah dikompresi dengan ukuran sebelum dikompresi[5].

$$Cr = \frac{\text{Ukuran data sesudah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100\% \quad (1)$$

### 2.3 Video MP4

MP4 adalah salah satu format video yang mungkin sering dijumpai di internet. MP4 merupakan alternatif format video selain beberapa format video Digital seperti AVI, MKV, 3GP, RMVB dimana masing-masing memiliki sifat dan kualitas yang berlainan. Format MP4 biasanya digunakan untuk video dengan kualitas tinggi yang tidak semua PC mampu memutarinya dengan baik. Sebuah *file* video digital dalam format MP4 memiliki beberapa bagian, yaitu : a. Video b. *Audio* c. *Subtitle*, Semua bagian ini terpisah, namun menjadi satu bagian didalam format MP4. Bagian-bagian ini nantinya akan digabungkan menggunakan sebuah codec MP4 sehingga video Digital dalam format MP4 ini dapat dibaca dan dijalankan menggunakan perangkat lunak *multimedia player*. [6]

### 2.4 Algoritma taboo Code

Algoritma Taboo memiliki pendekatan terhadap panjangnya suatu variable. Penggagas Algoritma Taboo adalah Steven Pigeon. Prinsip dari Algoritma Taboo yaitu untuk memilih bilangan positif  $n$  dan membalikkan pola  $n$  untuk mengindikasikan akhir dari kode tersebut [7].

Terdapat dua tipe yang ada dalam Algoritma Taboo diantaranya:

1. Tipe pertama dari Algoritma Taboo yaitu block-based dan memiliki panjang berupa kelipatan dari  $n$ . Block based Algoritma Taboo dari integer adalah string dari  $n$ - bit blocks, dimana nilai  $n$  dipilih oleh user dan blok terakhir memiliki sedikit pola taboo yang tidak dapat muncul pada blok lainnya.  $n$ -bit blok memiliki nilai  $2^n$ , jika salah satu nilai dicadangkan, masing-masing sisa kode blok dapat memiliki salah satu yang ada pada pola bit  $2^n - 1$ .
2. Tipe kedua dari Algoritma Taboo yaitu panjang total kode ini tidak terbatas pada kelipatann. Tipe ini dikatakan unconstrained dan ditunjukkan bahwa tipe ini memiliki relasi terhadap nomor fibonaci ke- $n$ .

Rumus dari Algoritma Taboo codes yaitu:

$$g_n(k) = \sum_i^k (2^n - 1)^i = \frac{[(2^n - 1)^k - 1](2^n - 1)}{2^n - 2} \quad (2)$$

$g_n(k)$  : jumlah total kode dalam rentang  $k$

$n$  : jumlah blok bit

Tabel 1. Pola Taboo Codes

m	Code	m	Code	M	Code	m	Code
0	01 00	4	01 10 00	8	10 11 00	12	01 01 01 00
1	10 00	5	01 11 00	9	11 01 00	13	01 01 10 00
2	11 00	6	10 01 00	10	11 10 00	14	01 01 11 00
3	01 01 00	7	10 10 00	11	11 11 00	...	

## 3. HASIL DAN PEMBAHASAN

Salah satu permasalahan yang hadir di era milenial ini adalah semua kebutuhan yang dilakukan banyak orang sangat bergantung pada teknologi dan dunia digital, dunia teknologi saat ini sudah sangat *modern* sehingga mengakibatkan *file* berupa *text* juga membutuhkan ruang penyimpanan yang besar sehingga membuat pemborosan di ruang penyimpanan, dan koneksi internet di Indonesia juga sangat rendah sehingga dibutuhkan proses *transfer* data secara cepat baik secara *online* maupun *offline*. Pada penelitian ini akan dilakukan analisa dan perancangan sistem pengkompresian *file* video menggunakan algoritma *Taboo Codes*.

Algoritma *Taboo Codes* merupakan algoritma kompresi yang bersifat *lossless*, algoritma *Taboo Codes* mengkompresi *file* dengan frekuensi karakter pada objek yang akan dilakukan kompresi. Algoritma *Taboo Codes* bekerja dengan cara mengurutkan karakter yang sering muncul dan menggantikan bilangan biner dari setiap karakter menjadi nilai kebenaran berdasarkan tabel kebenaran algoritma *Taboo Codes*.

Permasalahan yang diangkat dari penelitian ini adalah untuk mendapatkan algoritma kompresi yang lebih baik yang akan digunakan untuk memperkecil. ukuran *file* video berformat \*MP4 dengan menggunakan algoritma *Taboo Codes*. Pada penelitian ini prosedur kerjanya adalah dengan cara mengkompresi *file* video dan *file* video yang sudah dikompresi kemudian didekompresi lagi untuk mengembalikan *file* video ke *format* semula.

Adapun proses mengenai sistem dan cara kerja kompresi *file* video pada penelitian ini. Pada gambar dibawah dijelaskan contoh *file* video dikompresi menggunakan algoritma *Taboo Codes* dan menghasilkan *output* hasil kompresi berupa karakter/symbol-simbol unik, kemudian hasil dari kompresi *file* video yang berupa simbol unik tersebut selanjutnya dikompresi kembali menggunakan algoritma *Taboo Codes* untuk mengembalikan *file* asli berupa karakter tulisan.



**Gambar 1.** *General Architecture System*

Penelitian ini membahas 2 proses utama yaitu proses kompresi dan proses dekompresi, dalam penelitian ini penulis akan mengkompresi *file* video berformat \*.doc menggunakan algoritma *Taboo Codes*, algoritma ini merupakan algoritma yang bersifat *lossless* sehingga *file* yang dikompresi tidak hilang banyak dan ketika didekompresi *file* akan kembali kepada ukuran semula sehingga tidak mempengaruhi isi *file* yang dikompresi dan tidak menghilangkan video.

Algoritma *Taboo Codes* memiliki pendekatan terhadap panjangnya suatu variabel. Penggagas Algoritma *Taboo Codes* adalah Steven Pigeon. Prinsip dari Algoritma *Taboo Codes* yaitu untuk memilih bilangan positif  $n$  dan membalikkan pola  $n$  untuk mengindikasikan akhir dari kode tersebut.

Terdapat dua tipe yang ada dalam Algoritma *Taboo Codes* diantaranya:

1. Tipe pertama dari Algoritma *Taboo Codes* yaitu *block-based* dan memiliki Panjang berupa kelipatan dari  $n$ . *Block based* Algoritma *Taboo Codes* dari *integer* adalah *string* dari  $n$ - bit *blocks*, dimana nilai  $n$  dipilih oleh *user* dan blok terakhir memiliki sedikit pola *taboo* yang tidak dapat muncul pada blok lainnya.  $n$ -bit blok memiliki nilai  $2^n$ , jika salah satu nilai dicadangkan, masing-masing sisa kode blok dapat memiliki salah satu yang ada pada pola bit  $2^n - 1$ .
2. Tipe kedua dari Algoritma *Taboo Codes* yaitu panjang total kode ini tidak terbatas pada kelipatan. Tipe ini dikatakan *unconstrained* dan ditunjukkan bahwa tipe ini memiliki relasi terhadap nomor *fibonacci* ke- $n$ .

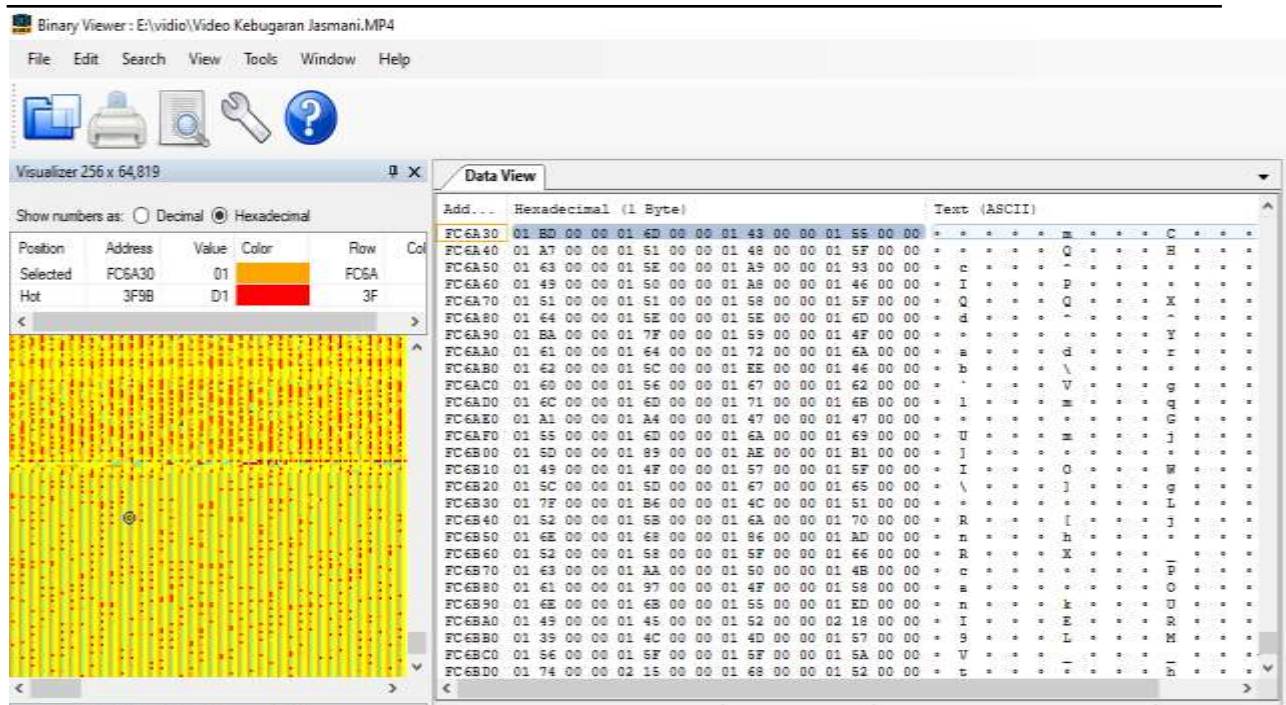
### 3.1 Prose Kompresi

Sebelum *file* dikompresi, terlebih dahulu dilakukan pembacaan biner yang terdapat pada *file* video untuk mendapatkan data berupa data biner. Membaca biner yang terdapat pada *file* video menggunakan aplikasi *Binary Viewer* untuk mencari nilai biner pada *file* video. Tabel 2 adalah contoh *file* video yang akan dikompresi dan didekompresi.

**Tabel 2.** Sampel *file* video yang akan di kompresi

Nama File	Video Kebugaran Jasmani
Size	150,8 MB
Type	MP4
Durasi	1:30:8 Menit

Berdasarkan table 2 terdapat nilai *biner viewer*, adapun nilai *hexadecimal* dari contoh *file* video pada sampel E:/ Video Kebugaran Jasmani tersebut adalah :



**Gambar 2.** Kode *Hexadecimal* file video dari hasil *Binary Viewer*

**Tabel 3.** *String* karakter yang belum dikompresi

M	Karakter	Frekuensi	ASCII (binary)	Bit	Frek×Bit
1	01	4	00000001	8	32
2	BD	1	10111101	8	8
3	00	4	00000000	8	32
4	6D	1	01101101	8	8
5	43	1	01100011	8	8
6	55	1	01010101	8	8
Total Bit					96

Berdasarkan tabel 3, satu karakter bernilai delapan bit bilangan biner sehingga 12 karakter pada *string* mempunyai nilai biner sebanyak 96 bit. Algoritma *Taboo Codes* bekerja dengan memulai *sorting* jumlah frekuensi kemunculan setiap karakter yang akan dikompresi. Adapun *string* bit sebelum dikompresi adalah sebagai berikut: 0000000110111101000000000110110101 10001101010101

Setelah menentukan kemunculan setiap karakter dan mencari nilai bilangan biner dari setiap karakter, langkah selanjutnya adalah mengurutkan setiap karakter berdasarkan frekuensi kemunculan yang paling banyak pada setiap karakter. Setelah diurutkan berdasarkan frekuensi terbanyak, bilangan biner sebelum dikompresi pada setiap karakter diganti dengan kode kebenaran dari algoritma *taboo codes* yang diurutkan berdasarkan frekuensi kemunculan.

**Tabel 4.** Data Setelah Dikompresi Menggunakan Algoritma *Taboo Codes*

N	Char	Biner	Bit	Frek	Bit × Frek
1	01	0100	4	4	16
2	00	1000	4	4	16
3	BD	1100	4	1	4
4	6D	010100	6	1	6
5	43	011000	6	1	6
6	55	011100	6	1	6
Total Bit					54

Setelah kode berhasil di *encode* berdasarkan kode kebenaran algoritma *Taboo Codes*, tahap selanjutnya adalah menyusun kembali kode-kode yang telah dihasilkan dari proses kompresi sesuai dengan posisi karakter pada *string* seperti pada tabel 4.

**Tabel 5.** *String* bit hasil kompresi dengan algoritma *Taboo Codes*



01	BD	00	01	6D
0100	1100	1000	0100	010100
01	43	00	01	55
1000	011000	1000	0100	011100

Setelah setiap karakter disusun kembali menggunakan kode kebenaran algoritma *taboo codes*, tahap selanjutnya menyusun kembali hasil kompresi setiap karakter, sehingga diperoleh *string bit* sebagai berikut: 0100 1100 1000 0100 0101001000 011000 1000 0100 011100.

Dalam hal penyimpanan data digunakan aturan 8 bit, sehingga untuk menyesuaikan data dengan aturan penyimpanan *string bit* sebanyak kekurangan dari kelipatan 8. Setelah itu, untuk mengetahui banyaknya *padding* dan *flagging* yang akan ditambahkan pada *string bit* kompresi, maka total dari keseluruhan kompresi *string bit* akan dibagi 8. Jika sisa bagi total seluruh *string bit* terhadap 8 adalah 0 maka akan ditambahkan *string bit* 00000001, sedangkan jika sisa bagi panjang *string bit* terhadap 8 tidak habis dibagi, maka akan ditambahkan *padding* dan *flagging*.

**Tabel 6.** Penambahan *padding* dan *flagging*

Padding	Flagging
$7 - n + "1"$	$9 - n$
$7 - 6 + "1" = 01$	$9 - 6 = 3$
	00000011

Sehingga diperoleh *string bit* sebagai berikut: 0100 1100 1000 0100 010100 1000 011000 1000 0100 01110001 00000011. Dimana total dari keseluruhan bit yang telah ditambahkan *padding* dan *flagging*, panjang *string bit* awal adalah 54 setelah ditambahkan *padding* dan *flagging* panjang total keseluruhan *string bit* menjadi 56 bit. Selanjutnya total seluruh *string bit* akan dibagi menjadi 8 bit dalam satu kelompok *string bit*, dan *string bit* yang sudah dikelompokkan akan diubah ke dalam bilangan desimal, bilangan desimal yang diperoleh dari bilangan kelompok biner selanjutnya akan diubah menjadi berbagai karakter unik yang terdapat pada tabel ASCII.

**Tabel 7.** Pengelompokan Bit

No	String bit hasil kompresi	Bilangan Desimal
1	01001100	76
2	10000100	132
3	01010010	82
4	00011000	24
5	10000100	130
6	01110001	113
7	00000011	3

Berdasarkan hasil kompresi *file* video diatas menggunakan algoritma *Taboo Codes*, maka didapatkan hasil kompresi *string bit* yang lebih sedikit yaitu sebanyak 56 *string bit* dari *string bit* awal sebanyak 96 *string bit*, sebagai berikut :

01001100100001000101001000011000100001000111000100000011.

### 3.2 Parameter Analisis Kompresi

Dari hasil kompresi diatas setiap kelompok *string bit* menghasilkan sebuah karakter unik, karakter unik tersebut akan disimpan kedalam notepad dengan ekstensi \*.MP4. Dalam menganalisis algoritma kompresi parameter yang akan dipakai penulis sebagai tolak ukur untuk menentukan kinerja algoritma *Taboo Codes* adalah *Compression Ratio* dan *Space Saving*.

1. *Compression Ratio*

$$CR = \frac{\text{ukuran data sesudah dikompresi}}{\text{ukuran data sebelum dikompresi}} \times 100\%$$

$$CR = \frac{56 \text{ bit}}{96 \text{ bit}} \times 100\%$$

$$CR = 58,33 \%$$

2. *Space Saving*

$$SS = 1 - \frac{\text{ukuran file kompresi}}{\text{ukuran file asli}} \times 100\%$$

$$SS = 1 - \frac{56}{96} \times 100\%$$



$$SS = (1 - 0.8533) \times 100\%$$

$$SS = 0.1467 \times 100\%$$

$$SS = 0.1467\%$$

### 3.3 Proses Dekompresi

Setelah *file* berhasil dikompresi maka akan menghasilkan *file* baru yang berisi informasi serta *string bit* dari hasil proses kompresi. Hasil kompresi dengan mengembalikan *binary* menjadi *string bit* semula dengan menghilangkan biner *padding* dan *flagging*. 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  $n = 6$ , hilangkan  $7 + n$  atau  $7 + 3 = 10$ . Penjelasan diatas menunjukkan bahwa bit akhir harus dihilangkan. Hasil pengembalian *binary* menjadi *string bit* semula dapat dilihat sebagai berikut ini: 0100 1100 1000 0100 010100 1000 011000 1000 0100 011100.

Setelah proses penghilangan *padding* dan *flagging*, langkah selanjutnya semua bilangan biner hasil kompresi akan diubah langsung ke bilangan biner sebelum didekompresi, pembacaan *string bit* dilakukandari indeks terkecil sampai indeks terakhir dengan terus menambahkan nilai pada indeks sebelumnya yang tidak mewakili karakter kode kebenaran algoritma *Taboo Codes*. Ketika setiap pembaca *string bit* sudah menemukan kode kebenaran algoritma *Taboo Codes* maka nilai biner tersebut sudah mewakili nilai kebenaran algoritma *Taboo Codes*.

**Tabel 8.** Pengecekan *Bit* Dekompresi

Indeks	Nilai Biner	Keterangan	Nilai Heksadesimal
1	0	Tidak Ada	
2	01	Tidak Ada	
3	010	Tidak Ada	
4	0100	Ada	01
5	1	Tidak Ada	
6	11	Tidak Ada	
7	110	Tidak Ada	
8	1100	Ada	BD
9	1	Tidak Ada	
10	10	Tidak Ada	
11	100	Tidak Ada	
12	1000	Ada	00
13	0	Tidak Ada	
14	01	Tidak Ada	
15	010	Tidak Ada	
16	0100	Ada	01
17	0	Tidak Ada	
18	01	Tidak ada	
19	010	Tidak Ada	
20	0101	Tidak Ada	
21	01010	Tidak Ada	
22	010101	Ada	6D
23	1	Tidak Ada	
24	10	Tidak Ada	
25	100	Tidak Ada	
26	0100	Ada	01
27	0	Tidak Ada	
28	01	Tidakada	
29	011	Tidak Ada	
30	0110	Tidak Ada	
31	01100	Tidak Ada	
32	011000	Ada	43
33	1	Tidak Ada	



34	10	Tidak Ada	
35	100	Tidak Ada	
36	1000	Ada	01
37	0	Tidak Ada	
38	01	Tidak Ada	
39	010	Tidak Ada	
40	0100	Ada	01
41	0	Tidak Ada	
42	01	Tidak Ada	
43	011	Tidak Ada	
44	0111	Tidak Ada	
45	01110	Tidak Ada	
46	011100	Ada	55

#### 4. KESIMPULAN

Berdasarkan penelitian yang dilakukan pada kompresi *file* video menggunakan algoritma *taboo code* maka dapat diambil kesimpulan dimana hasil dari kompresi dari sampel data yang digunakan dengan parameter analisis kompresi *compression ratio* dan *space saving* dengan hasil *compression ratio* sebesar 58,33 % dan hasil *space saving* sebesar 0.1467% menunjukkan tingkat hasil kompresi yang sangat baik pada *file* video dan sangat baik apabila diterapkan

#### REFERENCES

- [1] U. S. Utara, U. S. Utara, and U. S. Utara, "Pengamanan File Teks Menggunakan Algoritma Skipjack dan Kompresi Algoritma Taboo," 2018.
- [2] R. Syahputra, "KOMPRESI FILE VIDEO MP4 DENGAN MENGGUNAKAN METODE," pp. 52–57, 2016.
- [3] H. T. Sihotang, "PERANCANGAN DAN IMPLEMENTASI ALGORITMA ARITHMETIC CODING UNTUK APLIKASI KOMPRESI DATA VIDEO DAN AUDIO," vol. 2, no. 1, pp. 58–64, 2018.
- [4] D. S. T. Mizwar, G. L. Ginting, Mesran, A. Fau, S. Aripin, "Implementasi Algoritma J-Bit Encoding pada kompresi File Teks," KOMIK (Konferensi Nas. Teknol. Inf dan komputer), vol. 1, pp. 232–236, 2017.
- [5] T. Chandra, "Aplikasi Kompresi File dengan Algoritma Elias Gamma," no. 18.
- [6] K. Y. and Y. Melita, "Aplikasi kompresi Citra digital menggunakan teknik Kompresi jpeg Dengan fungsi GUI pada Matlab," vol. 3, no. no.2, pp. 269–278, 2011.
- [7] D. Salomon and G. Motta, *Handbook Of Data Compression*, Fifth Edit. London: Springer, 2010.
- [8] R. Syahputra, "KOMPRESI FILE VIDEO MP4," pp. 57–59, 2016.
- [9] T. Sutabri, *Analisis Sistem Informasi*. 2012.
- [10] R. A. S.-M. Salahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*, 4th ed. Bandung: Informatika Bandung, 2016.
- [11] R. Priyanto, "Langsung Bisa Visual Basic NET 2008," 2010.
- [12] M. Sadeli, *7 Jam Belajar Interaktif Visual Basic.net 2008 Untuk Orang Awam*. Palembang: Maxikom, 2009.