

## Implementasi Algoritma Tunstall Codes Untuk Kompresi File Vidio Pada Aplikasi Chatting Berbasis Android

Muhammad Tifri

Program Studi Teknik Informatika, Fakultas Ilmu Komputer Dan Teknologi Informasi, Universitas Budi Darma, Medan, Indonesia

Jl. Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia

Email: Muhammad.tifri@gmail.com

**Abstrak**— File video adalah sebuah teknologi untuk menangkap, memproses mentransmisikan dan menata ulang gambar yang bergerak saat merekam video maka akan membutuhkan tempat penyimpanan yang besar dan sehingga dalam proses pertukaran data tidak terjadi masalah. Untuk mengatasi masalah tersebut maka didapatkan solusi dengan melakukan kompresi. Algoritma *Tunstal code* merupakan teknik kompresi *lossless*, yang tidak menghilangkan informasi sebelumnya, dimana hasil dekompresi dari video yang terkompresi sama dengan *file* vidio aslinya. Dengan menggunakan algoritma *Tunstal code* menghasilkan kinerja *Compression Ratio* sebesar 50% Hasil tersebut membuktikan bahwa kompresi *file* vidio dengan menerapkan algoritma *Tunstal code* dapat memperkecil ukuran *file* video yang besar menjadi lebih kecil dari ukuran aslinya.

**Kata Kunci:** kompresi, file, video, Tunstal, code.

**Abstract**— Video files are a technology for capturing, processing, transmitting and rearranging moving images. When recording a video, it will require a large storage area and so that there are no problems in the data exchange process. To overcome this problem, a solution was found by performing compression. The Tunstal code algorithm is a lossless compression technique, which does not lose previous information, where the decompression results of the compressed video are the same as the original video file. Using the Tunstal code algorithm produces a Compression Ratio performance of 50%. These results prove that video file compression by applying the Tunstal code algorithm can reduce the size of large video files to smaller than their original size.

**Keywords:** compression, file, video, Tunstal, code

### 1. PENDAHULUAN

Seiring perkembangan zaman perkembangan teknologi informasi saat ini dapat memudahkan manusia untuk beraktifitas melihat berita dan berbagi informasi yang beredar di sosial media, Sehingga hal ini menimbulkan masalah yaitu kebutuhan akan penyimpanan semakin meningkat. salah satunya yang sangat sering digunakan untuk berbagi informasi tersebut adalah aplikasi chatting berbasis android. Aplikasi chatting merupakan aplikasi yang memudahkan untuk berkomunikasi dan bertukar informasi secara *Real-time*, salah satu informasi yang dapat dibagi adalah file vidio. Akan tetapi dalam berkomunikasi melalui aplikasi chatting berbasis android ini diberikan batasan-batasan tidak seleluasa ketika kita berkomunikasi secara langsung pada umumnya. Dimana aplikasi chatting memiliki jumlah bit yang telah ditetapkan, sehingga dalam pengiriman file pada aplikasi chatting tidak dapat sekaligus atau dalam sekali pengiriman melainkan dalam beberapa kali pengiriman apabila file yang dikirim terlalu besar.

Perpindahan data sangat mudah dilakukan pada saat ini, tetapi tempat penyimpanannya yang menjadi kendala sangat mendasar. Hal ini disebabkan oleh ukuran data yang ingin dipindahkan tidak sesuai dengan media penyimpanan yang tersedia. Ukuran file yang semakin besar menuntut para pemakai Komputer untuk melakukan berbagai macam cara agar dapat menyimpan sejumlah file yang berukuran besar dalam media penyimpanan yang terbatas. Hal inilah yang menyebabkan file harus dimampatkan agar ukurannya menjadi lebih kecil. Teknik pemampatan data ini disebut dengan teknik kompresi data. Vidio merupakan teknologi pengiriman sinyal elektronik dari suatu gambar yang bergerak. Selain itu vidio merupakan teknologi yang gunanya menangkap, merekam, memproses, mentransmisikan serta menataulang gambar bergerak. Yang mana biasanya menggunakan film seluloid, sinyal elektronik, atau media digital. Digital vidio sendiri merupakan jenis sistem vidio recording yang bekerja menggunakan sistem digital dibandingkan dengan analog dalam hal representasi vidionya.

Kompresi data adalah sebuah teknik pada ilmu komputer untuk mengecilkan ukuran data. Hal ini juga dapat disebut dengan memanipulkan data. Jadi, data yang ada dimampatkan menjadi lebih kecil dari ukuran sebenarnya dengan tujuan mempercepat pengiriman pertukaran serta menghemat ruang penyimpanan. Melalui kompresi data ini diharapkan mampu mengurangi jumlah bit pada file vidio yang besar tadi sehingga dapat memudahkan proses pengiriman dan penerimaan file vidio tersebut. Salah satu algoritma kompresi yang dapat digunakan adalah algoritma *Tunstall Code* yang merupakan suatu algoritma yang membutuhkan masukan kata bersama dengan distribusi probabilitas untuk setiap masukan dan dikonstruksi sebagai pohon probabilitas yang akan diasosiasikan dengan huruf dari abjad masukan, algoritma ini juga dapat mengkompresi suatu file yang berukuran besar hingga memperkecil ukuran file tersebut

Pada penelitian terdahulu yang dilakukan oleh Dani Iqbal Pada Tahun 2018 telah menyimpulkan bahwa dalam mengkompresi *file* MP4 memberikan hasil rasio kompresi hampir 15% sehingga dapat menghemat ruang dan mempercepat proses pengiriman *file* MP4 yang dihasilkan [1]. Penelitian yang dilakukan oleh Juli Fitri, Muhammad Safi'i Berdasarkan penerapan algoritma *tunstall code* dapat membuktikan bahwa suatu *file* teks yang memiliki ukuran yang cukup besar dapat dikompresi menjadi lebih kecil dari ukuran sebelumnya [2]. Dalam pembahasan ini proses yang

akan dilakukan adalah dengan mengompresi file vidio dengan algoritma *Tunstall Code* yang akan menghasilkan sebuah file vidio yang sudah terkompres.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Adapun tahapan penelitian yang akan digunakan dalam penelitian ini sebagai berikut:

#### a. Studi Pustaka(*Library Research*)

Pada tahap ini penulis mengumpulkan referensi yang diperlukan dalam penelitian, yang mana dilakukan untuk memperoleh data-data atau informasi yang dibutuhkan dalam pembuatan skripsi ini. Pada tahap ini dilakukan pengumpulan buku, jurnal, e-book, artikel, makalah, maupun situs internet yang membahas tentang algoritma *Tunstall Codes* untuk dipelajari lebih lanjut.

#### b. Analisis dan Perancangan Sistem(*Analysysis and System Development*)

Tahap ini merupakan kegiatan untuk menganalisa kebutuhan sistem yang akan dibuat dan menjadi dasar untuk perancangan sistem, seperti analisa algoritma, data yang akan dijadikan input dan *user interface design*.

#### c. Perancangan

Pada tahap ini dilakukan perancangan aplikasi seperti pembuatan *flowchart*, pembuatan UML (*Unified Modelling Language*) meliputi *usecase* diagram, *activity* diagram serta *sequence* diagram dan rancangan untuk desain antarmuka aplikasi

#### d. Implementasi(*Implementation*)

Tahap ini merupakan kegiatan untuk membuat perangkat lunak yang sesuai dengan analisis dan perancangan sistem yang telah ditentukan.

#### e. Pengujian(*Testing*)

Tahap ini merupakan kegiatan untuk melakukan pengujian sistem apakah hasil penelitian yang telah dilakukan sesuai dengan tujuan penelitian dan menjawab rumusan masalah yang telah ditentukan.

#### f. Dokumentasi

Dalam tahap dokumentasi, dilakukan penyusunan laporan dari hasil perancangan sistem dalam format penulisan penelitian.

### 2.2 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]. Kompresi data adalah proses mengkodekan informasi menggunakan bit atau information-bearing unit yang lain yang lebih rendah daripada representasi data yang tidak terkodekan dengan suatu sistem enkoding tertentu. Contoh kompresi sederhana yang biasa kita lakukan misalnya adalah menyingkat kata-kata yang sering digunakan tapi sudah memiliki konvensi umum. Misalnya: kata “yang” dikompres menjadi kata “yg”.

Pengiriman data hasil kompresi dapat dilakukan jika pihak pengirim/yang melakukan kompresi dan pihak penerima memiliki aturan yang sama dalam hal kompresi data. Pihak pengirim harus menggunakan algoritma kompresi data yang sudah baku dan pihak penerima juga menggunakan teknik dekompresi data yang sama dengan pengirim sehingga data yang diterima dapat dibaca/di-dekripsi kembali dengan benar. Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan bandwidth. Teknik kompresi bisa dilakukan terhadap data teks/biner, gambar (JPEG, PNG, TIFF), audio (MP3, AAC, RMA, WMA), dan video (MPEG, H261, H263). [5]

### 2.3 Dekompreksi

Dekompreksi adalah sistem yang digunakan untuk mengembalikan sebuah file yang telah mengalami pemampatan (*compression*) pada bentuk dan ukuran file aslinya sehingga isi dari file tersebut dapat dilihat kembali [10]. Data yang telah dikompresi, harus dikembalikan seperti semula untuk kemudian dibaca kembali, proses pengembalian ini disebut sebagai proses dekompreksi. Sama seperti kompresi, dekompreksi dibagi menjadi dua, yaitu *lossy* dimana file yang dihasilkan, tidak sama persis seperti sebelum file dikompres, dan *lossless* dimana file yang dihasilkan akan sama persis seperti file sebelum dikompres [11]. Dari pengertian dekompreksi diatas, dapat penulis simpulkan bahwa dekompreksi adalah sebuah proses pengembalian data yang telah dikompres ke data semula untuk kemudian dibaca kembali dengan cara yang berbeda seperti pada waktu proses kompres dilakukan dengan memperhatikan waktu dan ukuran yang sama saat kompresi dilakukan.

### 2.4 Tunstall Code

*Tunstall code* merupakan salah satu algoritma yang termasuk dalam metode *lossless* data. *Tunstall coding* ditemukan oleh Brian Parker dalam tesisnya pada tahun 1967 ketika berada di Institute Teknologi Georgia. Langkah pertama dalam algoritma ini adalah membuat tabel simbol, frekuensi dan kolom *probability*. Setelah itu membuat simbol yang pendek ke dalam kemungkinan yang banyak. Kemudian melakukan iterasi untuk mengetahui berapa

banyak iterasi yang dilakukan untuk dimasukkan ke dalam formula  $N + k(N-1) \leq 2^n$ . Untuk melakukan iterasi, simbol yang pendek menjadi probabilitas yang banyak. Kemudian menghapus simbol dengan probabilitas tertinggi, setelah itu masukan simbol dengan simbol yang terdapat di dalam tabel.

Ilustrasi dari algoritma ini adalah sebuah kalimat yang terdiri dari dua simbol A dan B dimana A lebih sering dijumpai. Berikan sebuah *string* yang unik dari kalimat ini, diberikan sub-*string* dalam form AA, AAA, AB, AAB dan B, tetapi *string* yang jarang dalam bentuk BB. Masukkan ukuran kode yang sudah dimodifikasi ke dalam 5 bentuk *string* yang sudah dibuat. AA = 000, AAA = 001, AB = 010, ABA = 011 dan B = 100. *Tunstall code* memberikan sebuah alphabet untuk N simbol, dimulai dengan sebuah tabel kode yang terdiri dari simbol-simbol. Kemudian akan diiterasi sepanjang ukuran tabel kode yang lebih sedikit dari jumlah angka di dalam kode. Tiap iterasi mengikuti langkah-langkah [9] sebagai berikut:

- Pilih simbol dengan probabilitas terbanyak di dalam tabel, disimbolkan dengan S.
- Ganti S dan masukkan N sub-*string* S<sub>x</sub> dimana x malampaui semua simbol N. langkah ini meningkatkan ukuran simbol tabel N-1. Kemudian setelah iterasi k, ukuran tabel akan menjadi elemen N + k (N-1).
- Jika  $N + k(N-1) \leq 2^n$ , dilakukan dalam iterasi yang lain.

Sebuah properti penting dalam *Tunstall code* adalah keandalannya. Jika satu *bit* menjadi *corrupt*, hanya satu kode yang menjadi buruk. Tidak seperti *variable length codes* lainnya simbol dalam algoritma *tunstall* merupakan angka campuran [12].

## 2.5 File Vedio

Video adalah teknologi pemrosesan sinyal elektronik mewakilkan gambar bergerak. Video merupakan sekumpulan gambar bergerak yang diperoleh dari hasil rekaman kamera atau hasil animasi komputer. Pada mulanya informasi video ini disimpan secara analog, sebagai perubahan bentuk gelombang secara kontinyu yang mewakili adanya perubahan warna dan kecerahan (*brightness*) dari gambar yang direkam. Di sisi lain, komputer digital hanya dapat menyimpan dan mengolah data yang bersifat biner. Untuk itu di kalangan industri komputer didefinisikan warna dalam besaran 24-bit yang dapat digunakan untuk menyimpan sekitar 16,7 juta kemungkinan warna yang berbeda. Dengan demikian data video dapat disimpan secara digital sebagai titik-titik yang masing-masing memiliki warna tertentu dan titik-titik tersebut jika disusun sebagai satu kesatuan akan membentuk suatu gambar secara utuh [13].

## 3. HASIL DAN PEMBAHASAN

*File* vido yang berekstensi MP4 mempunyai ukuran yang cukup besar, semakin besar ukuruna yang ada dalam *file*, maka semakin besar pula tempat penyimpanannya, dan proses transmisi yang dibutuhkan juga semakin besar. Dalam melakukan proses kompresi *file* vido sebelumnya harus dilakukan analisa terhadap *file* vido.

### 3.1 Implementasi Algoritma Stunstal Code

Dengan melakukan kompresi *file* MP4, data yang berukuran besar akan dikompresi menjadi ukuran yang kecil dan akan mengurangi alokasi penyimpanan. Dalam menganalisa *file* MP4 harus dilakukan mengambilan *sample file* untuk mendapatkan nilai dari data pada sebuah *file* MP4 yang berupa nilai *hexadecimal*. Proses kompresi *file* vido dengan algortima *Tunstall code* yang akan dilakukan adalah dengan menggunakan rumus sebagai berikut:

$$N + K(N-1) \leq 2^n \quad (1)$$

Dimana n = Panjang *bit* yang diinginkan, N = Jumlah simbol dan k = maks.

#### 3.1.1 Proses Kompresi

Langkah-langkah proses kompresi algoritma *Tunstall code* dapat dilihat dari contoh dengan menggunakan hasil sebagai berikut:

- Analisa proses kompresi *file* video dengan menggunakan *Tunstall Code*

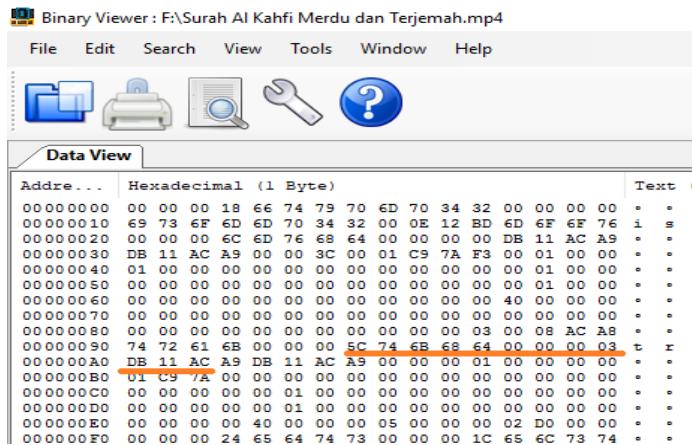
- Memasukkan *File*

*File* MP4 yang digunakan adalah *file* MP4 yang didownload dari internet. Berikut adalah informasi objek *file* video yang akan diambil sampelnya sebelum dilakukan kompresi:

**Tabel 1.** Informasi *File* Video *Sample*

Keterangan	
Format	.MP4
Nama <i>File</i>	Surah Al Kahfi Merdu Dan Terjemah
Ukuran	147 MB
Durasi	32.31 Menit

Dari *sample* MP4 di dapat nilai *hexadesimal* menggunakan bantuan *software binary viewer* seperti pada gambar di bawah ini:



Gambar 1. Nilai Hexadecimal File Vidio Sample

Berdasarkan pada gambar 2 diatas maka didapatkan nilai *hexadecimal file* video sample. Untuk keperluan hitungan manual, maka hanya akan diambil *sample* nilai sebanyak 12 karakter nilai *hexadecimal file* video *sample*. Nilai *hexadecimal* diambil dari sisi kiri sampai bilangan ke 12.

2. Melakukan Pembacaan Isi File

Adapun bilangan *hexadecimal* dari *file* video tersebut adalah 5C, 74, 6B, 68, 64, 00, 00, 00, 03, DB, 11, AC = 96 bit. Nilai data ini dimasukan kedalam tabel untuk dilakukan pembacaan frekuensi. Pembacaan frekuensi dilakukan dengan menghitung jumlah nilai yang sama di setiap nilai data yang muncul. Adapun pembacaan frekuensi dapat dilihat pada tabel di bawah ini:

Tabel 2. Simbol dan Frekuensi

Simbol	Frekuensi
5C	1
74	1
6B	1
68	1
64	1
00	3
03	1
DB	1
11	1
AC	1

Jumlah karakter berjumlah 12 sehingga dibutuhkan untuk *tunstall code* adalah 10 bit.

- b. Menghitung jumlah iterasi yang akan dilakukan dengan menggunakan rumus algoritma *tunstall code*.

$$N + k(N-1) \leq 2^n$$

$$12+k(12-1) \leq 2^{10}$$

$$12+ 11 k \leq 1024$$

$$11k \leq 1024-12= 1012$$

$$k \leq 1012/11= 92$$

Berdasarkan perhitungan diatas, iterasi yang akan dilakukan maksimal 92 kali iterasi, Dalam penelitian ini iterasi yang akan dilakukan hanya sebanyak 10 kali iterasi, karena dalam 10 kali iterasi hasil sudah didapatkan.

- c. Membuat tabel simbol dan probabilitas untuk melakukan iterasi, untuk mencari probabilitas adalah dengan membagi frekuensi kemunculan karakter dengan jumlah karakter.

$$5C = 1/10 = 0.1$$

$$74 = 1/10 = 0.1$$

$$6B = 1/10 = 0.1$$

$$68 = 1/10 = 0.1$$

$$64 = 1/10 = 0.1$$

$$00 = 3/10 = 0.3$$

$$03 = 1/10 = 0.1$$

$$DB = 1/10 = 0.1$$

$$11 = 3/10 = 0.3$$

$$AC = 1/10 = 0.1$$

Tabel 3. Tabel Simbol Dan Probabilitas

Simbol	Probabilitas
5C	0,1
74	0,1
6B	0,1
68	0,1
64	0,1

Simbol	Probabilitas
00	0,3
03	0,1
DB	0,1
11	0,1
AC	0,1

- d. Langkah selanjutnya dengan mengalikan karakter dengan probabilitas tertinggi dan menggabungkannya dengan karakter lainnya.  
e. Kompresi dilakukan dengan mengubah Simbol menjadi kode di dalam tabel *tunstall code* sehingga didapatkan hasil  $00001100001000100011100000000010100010001011010 = 48 \text{ bit}$

Ukuran *file* awal sebelum dikompresi adalah 96 bit, sehingga rasio kompresinya adalah

$$RC = (\text{Ukuran File Asli}) / (\text{Ukuran File Terkompresi})$$

$$RC = 96/48 = 2$$

Jika dinyatakan dalam bentuk persen maka dituliskan dalam rumus sebagai berikut:

$$SS = (1 - (\text{Ukuran File Terkompresi}) / (\text{Ukuran File Asli})) \times 100\%$$

$$SS = (1 - (48/96)) \times 100\%$$

$$SS = (1 - 0,5) \times 100\%$$

$$SS = 0,5 \times 100\%$$

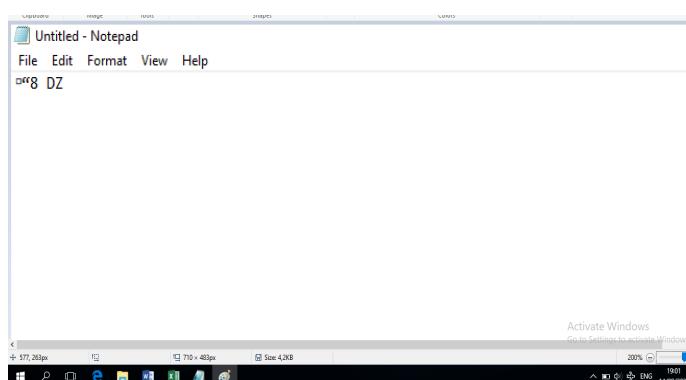
$$SS = 50\%$$

Dari perhitungan di atas, dapat disimpulkan bahwa dengan menggunakan algoritma *tunstall code* karakter di atas dapat di kompresi sebanyak 50 %.

Tabel 4. Hasil Karakter File Vidio Terkompresi

Biner	Dec	Char
00001100	12	□
00100010	34	“
00111000	56	8
00000001	1	█
01000100	68	D
01011010	90	Z

Setelah nilai *biner* diketahui, maka mengubah nilai *biner* ke decimal dan mengubahnya lagi kedalam suatu karakter. Karakter hasil dari proses kompresi yang dihasilkan tersimpan dalam suatu *file* dengan ekstensi “.tsl”, dan jika *file* tersebut dibuka dengan aplikasi *notepad*, maka akan tampil karakter seperti pada Gambar 3.2



Gambar 2. Hasil Karakter Kompresi

### 3.1.2 Proses Dekompreksi

Proses dekompreksi *tunstall code* diawali dengan meng-input *file tunstall code* yang akan didekompreksi, kemudian kode-kode untuk setiap simbol dari *file tunstall code* akan dibaca. Kode-kode tersebut akan dicocokkan *bit per bit* dengan kode pada simbol-simbol dari *file* asli. Berdasarkan tabel 3.18 iterasi kesepuluh nilai yang sudah menjadi kode akan dibaca oleh sistem, kemudian mengantinya dengan nilai awal nya sebagai berikut:

Kode 00001100 akan diganti dengan nilai simbol ‘5C74’,

Kode 00100010 akan digantikan dengan nilai simbol ‘6B68’,

Kode 00111000 akan digantikan dengan nilai simbol ‘6400’,

Kode 00000001 akan digantikan dengan nilai simbol ‘0000’,

Kode 01000100 akan digantikan dengan nilai simbol ‘03DB’,

Kode 01011010 akan digantikan dengan nilai simbol ‘11AC’,

Hingga menghasilkan *file* asli yaitu *file* teks “ 5C746B6864000003DB11AC”

## 4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan Berdasarkan prosedur kompresi dengan menggunakan algoritma *Tunstall Codes* telah berhasil melakukan proses kompresi *file* video hasil rekam layar berekstensi \*.MP4 sehingga proses kompresi dapat berjalan sesuai dengan teknik kompresi, Berdasarkan penerapan algoritma *Tunstall Code* telah membuktikan bahwa suatu *file* video yang memiliki ukuran besar dapat dikompres menjadi ukuran yang lebih kecil. Berdasarkan dari hasil pengujian terhadap sistem bahwa ukuran *file* video lebih kecil setelah dilakukan kompresi, Kompresi *file* video pada aplikasi chatting berbasis android dapat dirancang dan dibangun dengan menggunakan aplikasi *Eclipse Juno* dengan menerapkan algoritma *Tunstall Codes* sehingga diharapkan memudahkan penulis dalam mengkompresi ukuran *file* video dan adapun saran yang dapat diberikan pada penulis untuk pengembangan dan perbaikan lebih lanjut dalam melakukan penelitian yaitu Penggunaan algoritma *Tunstall Codes* dapat dikembangkan lagi untuk mengkompresi *file* video pada aplikasi lainnya. Sehingga memberi masukan yang lebih baik kemasa yang akan mendatang, Aplikasi yang dirancang untuk mengkompresi *file* video diharapkan dapat mengkompresi jenis *file* video yang berekstensi \*.3GP maupun yang lainnya dan Aplikasi yang digunakan kedepannya dapat berbasis Android sehingga lebih memberi perkembangan yang lebih baik.

## REFERENCES

- [1] D. Iqbal, “Implementasi Algoritma Levenstein Untuk Kompresi File Video,” Vol. 3, Pp. 266–273, 2019.*Kamus Besar Bahasa Indonesia (Kbbi)*.
- [2] A. M. Silitonga, S. D. Nasution, And P. Ramadhani, “Implementasi Algoritma Subexponential Code Untuk Kompresi File,” Vol. 3, Pp. 340–348, 2019.
- [3] A. Pendahuluan, “Kompresi Data,” Vol. 2, No. 1, 1940.
- [4] R. Syahputra, “Kompresi File Video Mp4 Dengan Menggunakan Metode,” Pp. 52–57, 2016.
- [5] A. A. Pramadi, S. D. Nasution, B. Purba, A. Event, And R. Code, “Penerapan Algoritma Even-Rodeh Pada Aplikasi Kompresi File,” Vol. 3, Pp. 73–84, 2019.
- [6] D. T. Sutoyo, S.Si, *Teori Pengolah Citra Digital*. Yogyakarta: Andi, 2009.
- [7] S. S. D. T. Sutoyo, “Teori Pengolah Citra Digital.”
- [8] Et Al Hidayat, W. Zaman, “Skema Kompresi Dan Dekompresi,” 2013.
- [9] R. N. Ibrahim, “Perbandingan Kompresi File Menggunakan Algoritma Run Length Dengan Two Level Hoshing,” Vol. 1, No. 2, Pp. 90–104, 2007.
- [10] E. Prayoga And K. M. Suryaningrum, “Implementasi Algoritma Huffman Dan Run Length Encoding Pada Aplikasi Kompresi Berbasis Web,” Vol. Iv, No. 2, Pp. 92–101, 2018.
- [11] Prof.David Salomon, “Handbook Of Data Comprassion,” 2010.
- [12] K. Umam And B. S. Negara, “Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Subtraction Dan Operasi Morfologi,” Vol. 2, No. 2, Pp. 31–40, 2016.
- [13] M. Ichwan, F. Hakiky, And J. T. Informatika, “Jurnal Informatika,” Vol. 2, No. 2, Pp. 13–21.
- [14] A. Nugroho, “Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode Usdp(Unfied Software Development Process).”
- [15] Rosa A.S And M.Shalahuddin, *Rekayasa Perangkat Lunak (Terstruktur Dan Berorientasi Objek)*. Bandung: Informatika Bandung, 2016.
- [16] S. K. Alfa Satyaputra, M.Sc, Eva Maulina Aritonang, *Java For Beginners With Eclipse 4.2 Juno*. Jakarta: Pt. Elex Media Komputindo, 2012.