**Jurnal Sains Dan Teknologi Informasi**
Vol 1, No 3, Juni 2022, Hal 85-89
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v1i3.2182
Website https://ejurnal.seminar-id.com/index.php/jussi

# Implementasi Algoritma H.264 Untuk Kompresi File Pada Aplikasi Transfer File

**Lamsihar Grasella Sianturi**

Program Studi Tenik Informatika, Fakultas Ilmu Komputer Dan Teknologi Informasi, Universitas Budi Darma,
Jalan Sisingamangaraja No.338, Medan, Sumatera Utara, Indonesia
Email: gracellasianturi83@gmail.com

**Abstrak-**Transfer file yang memiliki ukuran relatif besar, mengakibatkan lamanya proses transfer yang dilakukan dan juga menghabiskan kuota internet yang besar dalam proses pengirimannya. Dimana semakin kecil ukuran file yang akan dikirim maka proses transfernya juga cepat, serta tidak menghabiskan banyak kuota internet. Untuk mengatasi masalah tersebut diperlukan adanya kompresi yang berfungsi untuk mengurangi ukuran file yang akan di transfer. Salah satu dari algoritma kompresi adalah H.264. Proses kompresi menggunakan algoritma H.264 dilakukan pada saat file akan dikirim, lalu pada saat file diterima oleh penerima maka file akan didekompresi.Dengan menggunakan algoritma H.264 file akan dikirim berhasil mengurangi ukuran file dengan ratio of compression (RC) 2.5%, compression ratio (CR) 40%, redundancy (RD) 60%, serta membantu proses transmisi pengiriman file yang berukuran besar akan menjadi lebih cepat dan menghemat kuota internet.

**Kata Kunci:** Transfer File; H.264; Kompresi;

**Abstrac-**Transfer files that have a relatively large size, result in a long transfer process and also consume a large internet quota in the sending process. Where the smaller the file size to be sent, the transfer process is also fast, and does not consume a lot of internet quota. To solve this problem, it is necessary to have compression which functions to reduce the size of the files to be transferred. One of the compression algorithms is H.264. The compression process using the H.264 algorithm is carried out when the file is sent, then when the file is received by the recipient, the file will be decompressed. Using the H.264 algorithm, the file will be sent successfully reducing the file size with a ratio of compression (RC) 2.5%, compression ratio (CR) 40%, redundancy (RD) 60%, and helping the transmission process of sending large files will be faster and save internet quota.

**Keywords:** File Transfer; H.264; Compression

## 1. PENDAHULUAN

*Transfer file* ini sering dilakukan akibat penumpukan data pada suatu ruang penyimpanan sehingga user harus melakukan transfer data. *Transfer file* saat ini seakan sudah menjadi kebutuhan, tidak hanya gambar, tetapi *audio, video, share it, super beam, we transfer* dan aplikasi *transfer file* lainnya. *Transfer file* merupakan suatu aplikasi yang berfungsi untuk mempermudah atau alat bantu supaya lebih efisien dalam proses *transfer file* melalui perangkat kesuatu jaringan yang menggunakan koneksi internet. *Transfer file* ini dapat terjadi melalui komputer yang berbentuk *main frame* dan sebuah computer dijaringan local atau *transfer file* dapat terjadi dari computer kita ke*server*.

Share It merupakan aplikasi yang menerapkan arsitektur UPnP (Universal Plug And Play) aplikasi share it menyediakan akses kepada pengunjung atau *user* untuk berbagi content dengan aplikasi lain yang sejenis, dan diakses oleh beberapa *computer* secara bersamaan, Seiring berkembangnya dunia digitalisasi dalam seluruh bidang maupun masyarakat *transfer file* sangat dibutuhkan dan memudahkan dalam proses pengiriman dan penerimaan suatu file dan mempersingkat waktu maupun penghematan biaya. Meskipun ukurannya sudah diperkecil dari ukuran sebelumnya namun tidak sedikitpun mengurangi kualitas dari video tersebut dan kompresi semacam ini bersifat *loseless. Loseless* merupakan kompresi data dimana hasil kompresi dan dekompresinya sama dan yang mengalami perubahan hanya ukuran data saja.

Untuk melakukan proses kompresi user memerlukan algoritma yang tepat digunakan untuk mereduksi ukuran dari file video ini. Salah satunya adalah dengan menggunakan algoritma *H.264* dimana algoritma *H.264* merupakan algoritma yang akan menghitung berdasarkan dari parameter *Ratio of Compression (RC), Compression Ratio (CR), Space Savings (SS)* dan waktu kompresi [1].

## 2. METODOLOGI PENELITIAN

### 2.1 Kompresi

Kompresi merupakan proses untuk mengabaikan berbagai kerumitan yang tidak penting (redundansi) dari suatu informasi dengan cara memampatkan isi file sehingga kapasitasnya menjadi lebih kecil dengan meningkatkan kesederhanaannya dan tetap menjaga kualitas deskripsi dari informasi tersebut [1].

### 2.2 Algoritma H.264

Algoritma H.264 merupakan suatu standar yang mampu menciptakan kapasitas video yang baik pada bit rate yang jauh lebih rendah dari kriteria sebelumnya (yaitu, setengah atau kurang bit rate MPEG-2, H.263, atau MPEG-4 Bagian 2), tanpa meningkatkan kesulitan desain sehingga tidak praktis atau terlalu sulit untuk dilakukan. Semakin besar Qstep, maka akan semakin kecil ukuran data yang dikompres[5]. Proses kuantisasi dapat dilihat seperti dibawah ini Langkah – Langkah Algoritma H.2.64, Sebagai Berikut[5] :

**Jurnal Sains Dan Teknologi Informasi**
Vol 1, No 3, Juni 2022, Hal 85-89
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v1i3.2182
Website https://ejurnal.seminar-id.com/index.php/jussi

a. Langkah pertama yang perlu dilakukan adalah melakukan normalisasi
b. Tahap selanjutnya yaitu dengan melakukan transformasi dan kuantitasi, dapat menggunakan rumus sebagai berikut:

$$Y = (C \times C_T) E \quad (1)$$

$$X = C_T (Y E) C \quad (2)$$

c. Membagi nilai-nilai pada matriks hasil transformasi dengan suatu nilai $Q_{step}$ yang menampilkan ukuran step dari *quantizer*, Semakin besar nilai $Q_{step}$, maka akan semakin kecil ukuran data yang dikompres. Proses kuantisasi dapat dilihat di Persamaan 3.

$$z = round\left(\frac{1}{Q_{Step}} Y.E\right) \quad (3)$$

d. Selanjutnya menentukan *Variabel Lenght Coding*, Setelah proses normalisasi, transformasi dan kuantisasi, *macroblock* umumnya menghasilkan nilai-nilai yang sebagian besar bernilai nol.
e. Koefisien *non-zero* tertinggi setelah dilakukan *zigzag scanning* sering bernilai ±1 atau biasa disebut *trailing ones*.
f. Koefisien *non-zero* yang bernilai lebih besar umumnya hanya terletak di bagian kiri setelah dilakukan *zigzag scanning* (pada bagian DC), kemudian nilai akan terus mengecil pada komponen AC
g. Melakukan proses VLC dengan melakukan inisialisasi kode berdasarkan jumlah koefisien *non-zero*
h. Tahapan pengkodean pengkodean *trailing ones*.
i. Melakukan pengkodean pengkodean sisa koefisien *non-zero*
j. Melakukan pengkodean pengkodean jumlah nilai nol sebelum koefisien terakhir
k. Melakukan pengkodean pengkodean *run-zero*.
l. Untuk mengodekan bilangan bertanda, maka digunakan format yang ditunjukkan di Persamaan seperti dibawah ini.

$$M = \log2 (num 1) \quad (4)$$

$$INFO = num 1 2^M \quad (5)$$

$$kode [M_{zeors}][1][INFO] \quad (6)$$

$$Kode = \begin{cases} [M_{Zeros}][1][INFO], & num > 0 \\ [M_{Zeros}][0][INFO], & num > 0 \end{cases} \quad (7)$$

### 2.3 Aplikasi Share It

Aplikasi berasal dari kata *application* yang artinya implementasi, pemakaian, dan penggunaan. Secara istilah aplikasi adalah program siap pakai yang didokumentasikan untuk melakukan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju [9]. Share it merupakan aplikasi yang digunakan untuk mengirim berbagai file seperti video, foto, lagu, rekaman, aplikasi, memo antara satu perangkat dan perangkat lainnya. Aplikasi ini banyak digunakan oleh masyarakat, karena berbagi dan mengirim file tidak membutuhkan waktu yang lama dan tidak dipungut biaya apapun. Namum banyak kalangan seperti mahasiswa tidak mengetahui bahwa dengan menggunakan aplikasi share it memungkinkan pengguna untuk melakukan pelanggaran hak cipta namum dalam peraturan hak cipta juga terdapat penggunaan yang wajar. Penelitian ini menggunakan metode H.264.

## 3. HASIL DAN PEMBAHASAN

Dalam melakukan kompresi pada file video dengan menggunakan H.264 maka langkah yang perlu dilakukan adalah harus mengetahui ukuran dari video yang akan di kompres. Video pada dasarnya adalah urutan dari banyak citra digital yang ditampilkan pada suatu *frame rate* tertentu untuk menciptakan ilusi animasi. Pada penelitian ini contoh video yang akan dikompres adalah video korea yang berekstensi.mp4. dapat dilihat pada gambar 1

**Jurnal Sains Dan Teknologi Informasi**
Vol 1, No 3, Juni 2022, Hal 85-89
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v1i3.2182
Website https://ejurnal.seminar-id.com/index.php/jussi

**Gambar 1.** video dengan ukuran

Untuk keperluan hitungan manual, maka hanya akan diambil *sample* nilai sebanyak 16 karakter nilai *hexadecimal file* video *sample*. Nilai *hexadecimal* diambil dari sisi kiri sampai bilangan ke 16.

**Tabel 1.** Nilai Hexadecimal Video 4 x 4 piksel

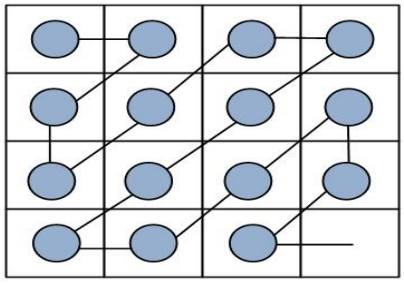
| | | | |
|---|---|---|---|
| 00 | 00 | 00 | 20 |
| 6F | 32 | 61 | 76 |
| 00 | 00 | 00 | 00 |
| 6E | FC | 17 | 53 |

Setelah mendapatkan nilai hexadesimal dari video yang akan di kompres, langkah yang perlu dilakukan terlebih dahulu adalah dengan melakukan konversi nilai dari ncilai hexadesimal video kedalam nilai desimal.

**Tabel 2.** Nilai Konversi Hexadesimal 4x4 piksel

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 32 |
| 111 | 50 | 97 | 118 |
| 0 | 0 | 0 | 0 |
| 110 | 252 | 23 | 83 |

Pada tahap ini, *macroblock* yang semula berbentuk dua dimenasi diubah menjadi dua dimenasi sehingga dapat dikodekan dengan VLC. Metode *zigzag scanning* melakukan pengurutan pada frekuensi terendah (DC) hingga frekuensi tertinggi (AC).



**Gambar 2.** Zigzag Scanning

a. Kode inisialisasi
Pada tahap ini, dilakukan pengkodean yang menunjukkan total dari koefisien *trailing ones* dan koefisien *non-zero* (selain *trailing ones*). Karena *macroblock* yang digunakan berukuran 4x4, koefisien *non-zero* dapat bernilai dari 0 (tidak ada koefisien *non-zero* dalam *macroblock*) hingga 16 (semua koefisien adalah *non-zero*). Sedangkan *trailing ones* dapat bernilai dari 0 hingga
3. Jika lebih dari tiga *trailing ones* dalam *macroblock*, maka hanya tiga yang dikodekan sebagai *trailing ones*, sisanya dikodekan sebagai koefisien *non-zero*. Hal ini dilakukan untuk efisiensi panjang kode inisial sehingga dapat menghasilkan kode yang efisien. Kode inisial yang dihasilkan didapat dari *lookup table* yang sesuai dengan jumlah *trailing ones* dan koefisien *non-zero*.

b. Pengkodean *Trailling Ones*
Untuk setiap koefisien *trailing ones*, tanda (*sign*) dari koefisien dikodekan dengan satu bit, yaitu "0" jika koefisien bernilai positif, dan "1" jika koefisien bernilai negatif. Pengkodean dimulai dari koefisien pada frekuensi tertinggi (sebelah kanan).

c. Pengkodean sisa koefisien *non-zero*
Koefisien *non-zero* selain *trailing ones* dikodekan menggunakan *Exponential Golomb Coding* (EGC) yang dimulai dari koefisien pada frekuensi tertinggi. EGC merupakan teknik pengkodean *integer* yang akan menghasilkan kode dengan panjang yang minimum untuk bilangan yang kecil, dan akan menghasilkan kode yang lebih panjang jika bilangan yang dikodekan bernilai lebih besar.

**Tabel 3.** Exp-Golomb Codewords

| Num | CodeWord | Num | CodeWord |
|---|---|---|---|
| 0 | 1 | | |
| 1 | 010 | -1 | 100 |
| 2 | 011 | -2 | 101 |
| 3 | 00100 | -3 | 11000 |
| 4 | 00101 | -4 | 11001 |
| 5 | 00110 | -5 | 11010 |
| 6 | 00111 | -6 | 11011 |

**Jurnal Sains Dan Teknologi Informasi**
Vol 1, No 3, Juni 2022, Hal 85-89
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v1i3.2182
Website https://ejurnal.seminar-id.com/index.php/jussi

| | | | |
|---|---|---|---|
| 7 | 0001000 | -7 | 1110000 |
| 8 | 0001001 | -8 | 1110001 |

d. Pengkodean jumlah koefisien nol sebelum koefisien *non-zero* terakhir
Setelah dilakukan *zigzag scanning* pada keluaran *quantizer*, idealnya matriks akan berisi koefisien-koefisien yang terurut mengecil dari koefisien tertinggi pada komponen DC hingga urutan nilai nol pada komponen AC. Akan tetapi, ada kalanya di antara koefisien-koefisien *non-zero* juga akan terdapat koefisien bernilai nol, sehingga harus ikut dikodekan.

e. Pengkodean *running zero*
Setelah menghitung jumlah koefisien nol di antara koefisien-koefisien *nonzero*, selanjutnya adalah menentukan kode yang menunjukkan posisi dari koefisien-koefisien nol tersebut di antara koefisien-koefisien *non-zero.*
Contoh :
Terdapat *Macro Block* dengan nilai sebagai berikut:

**Tabel 4.** Nilai *macroblock* dengan ukuran 4x4

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 32 |
| 111 | 50 | 97 | 118 |
| 0 | 0 | 0 | 0 |
| 110 | 252 | 23 | 83 |

Setelah melakukan proses *zigzag scanning* maka diperoleh nilai matrik dengan nilai:

$$\begin{bmatrix} 0 & 0 & 111 & 0 \\ 50 & 0 & 32 & 97 \\ 0 & 110 & 252 & 0 \\ 118 & 0 & 23 & 83 \end{bmatrix}$$

Dari matrik diatas maka diperoleh total *trailing ones* = 252 (diambil dari nilai matrik tertinggi, sedangkan nilai koefisien *non-zero* selain *trailing ones* = 23 (diambil dari nilai matriks dengan nilai terendah).Proses *Encoding* Sebelum melakukan proses *encoding* maka semua nilai matrik dirubah terlebih dahulu kedalam nilai biner.

0 = 00000000
0 = 00000000
111 = 01101111
50 = 00110010
32 = 00100000
97 = 01100001
0 = 00000000
110 = 0 1101110
252 = 11111100
0 = 00000000
118 = 01110110
0 = 00000000
23 = 00010111
83 = 01010011

Mengurutkan dari karakter yang memiliki frekuensi terbesar (banyak nilai yang sama) ke frekuensi terkecil. Urutan nilai dapat dilihat pada tabel berikut ini:

**Tabel 5.** Proses Encoding

| Elemen Kode Inisial | Keterangan T1 = 252 , nT16 = 0 | Nilai | Jumlah Bit | Frekuensi | Total Bit |
|---|---|---|---|---|---|
| T1 | 252 | 11111100 | 8 | 4 | 32 |
| T2 | 118 | 1110110 | 7 | 4 | 28 |
| T3 | 111 | 1101111 | 7 | 4 | 28 |
| T4 | 110 | 1101110 | 7 | 4 | 28 |
| T5 | 97 | 1100001 | 7 | 4 | 28 |
| T6 | 83 | 1010011 | 7 | 4 | 28 |
| T7 | 50 | 0110010 | 7 | 4 | 28 |
| T8 | 32 | 0100000 | 7 | 4 | 28 |
| T9 | 23 | 0010111 | 7 | 4 | 28 |
| T10 | 0 | 0000000 | 7 | 4 | 28 |
| T11 | 0 | 0000000 | 7 | 4 | 28 |
| T12 | 0 | 0000000 | 7 | 4 | 28 |

**Jurnal Sains Dan Teknologi Informasi**
Vol 1, No 3, Juni 2022, Hal 85-89
ISSN 2809-610X (Media Online)
DOI 10.47065/jussi.v1i3.2182
Website https://ejurnal.seminar-id.com/index.php/jussi

| | | | | | |
|---|---|---|---|---|---|
| T13 | 0 | 0000000 | 7 | 4 | 28 |
| T14 | 0 | 0000000 | 7 | 4 | 28 |
| T15 | 0 | 0000000 | 7 | 4 | 28 |
| T16 | 0 | 0000000 | 7 | 4 | 28 |
| Hasil | 1111110011101101101111110111011000011010011011001 0 | | | | **452** |
| Decode | 0100000001011100000000000000000000000000000000000000000 0000000000000 | | | | |

Setiap nilai dari tabel diatas dilakukan pengurangan 1 bit nilai yaitu "0" untuk koefisien nilai positif. nilai bit tersebut ditambahkan pada bagian depan.

a. Ratio of compression (Rc)

$$Rc = \frac{512}{452}$$

Rc = 1,10 %

**b.** *Compression Ratio* $(C_R)$

$$Cr = \frac{452}{512} x\ 100\ \%$$

$$C_R = 1{,}13\ \%$$

c. Redudancy (Rd)

$$Rd = 100\% - C_R$$

Rd = 0,98 %

## 4. KESIMPULAN

Berdasarkan dari penelitian yang telah dilakukan, maka hasil akhir dari penelitian tersebut dapat diambil beberapa kesimpulan dari pembahasan sebelumnya. Adapun kesimpulan dimana tahapan file transfer dalam mengirim sebuah file yaitu dengan mimilih file yang akan ditransfer, kemudian file tersebut akan dikompresi lalu file hasil kompresi tersebut dikirim ke penerima lalu dilakukan proses dekompresi oleh penerima dan tahapan tersebut telah berhasil dilakukan. Penerapan algoritma H.264 terhadap file yang akan dikrim berhasil mengurangi ukuran file dengan *ratio of compression* (rc) 2,5%, *compression ratio* (cr) 40%, *redundancy* (rd) 60% sehingga membantu proses transmisi pengiriman file yang berukuran besar akan menjadi lebih cepat dan menghemat kuota internet. Perancangan aplikasi *file transfer* dengan menggunakan *Visual Basic* telah selesai dibangun dan dapat berguna dalam mengirim suatu *file*.

## REFERENCES

[1] D. Salomon and G. Motta, *Handbook Of Data Compression*, Fifth Edit. London: Springer, 2010.
[2] G. M. David Salomon, *H.O.D.COMPRESSION*. Spinger London Dordrecht Heidelberg New York, 2010.
[3] F. G. et Al, "Analisa perbandingan Kompresi dan Dekompresi Menggunakan Algoritma Shannon," vol. 3, no. no.3, pp. 5197–5204, 2016.
[4] 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.
[5] lain E. G. Richardson, "H.264 and MPEg_-4 Video compresion," no. wyle.united kingsotn, 2003.
[6] A.Nugroho, "Rekayasa perangkat lunak Berorientasi Objek dengan metode USDP (Unfied Sotfware DevelopmentProces)."
[7] R. A. S. M.Shalahuddin, *Rekayasa Perangkat Lunak*. Bandung, 2015.
[8] R. Priyanto, "Langsung Bisa Visual Basic NET 2008," 2010.