

**NASKAH PUBLIKASI**

**IMPLEMENTASI METODE  
DWT (DISCRETE WAVELET TRANSFORM) DAN METODE HUFFMAN  
TERHADAP KOMPRESI CITRA DIGITAL MENGGUNAKAN GOOGLE  
COLLABORATORY**



Disusun oleh:

**DENI ARDIYANTO**

**5150411365**

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2020**

NASKAH PUBLIKASI

**IMPLEMENTASI METODE  
DWT (DISCRETE WAVELET TRANSFORM) DAN METODE HUFFMAN  
TERHADAP KOMPRESI CITRA DIGITAL MENGGUNAKAN GOOGLE  
COLLABORATORY**

Disusun oleh:

**DENI ARDIYANTO**

**51504111365**



Telah disetujui oleh pembimbing

Pembimbing

  
M.S. Hendriyawan Ahmad, S.T., M.Eng.

Tanggal 29/12/2020

# IMPLEMENTASI METODE DWT (DISCRETE WAVELET TRANSFORM) DAN METODE HUFFMAN TERHADAP KOMPRESI CITRA DIGITAL MENGGUNAKAN GOOGLE COLLABORATORY

**DENI ARDIYANTO, M.S. HENDRIYAWAN AHMAD**

*Program Studi Informatika, Fakultas Teknologi Informasi dan Elektro  
Universitas Teknologi Yogyakarta*

*Jl. Ringroad Utara Jombor Sleman Yogyakarta*

*E-mail : [deniardi4@gmail.com](mailto:deniardi4@gmail.com)*

## ABSTRAK

*Perkembangan teknologi informasi pada era digitalisasi sekarang ini pada berbagai sektor kehidupan masyarakat baik sektor perekonomian, industri, pariwisata, politik dan sebagainya mengharuskan masyarakat untuk siap menerima berbagai macam teknologi yang berkembang tersebut. Salah satu perkembangan teknologi informasi yang saat ini sudah menjadi bagian aktivitas keseharian dari masyarakat digital adalah pertukaran informasi berupa file, baik itu file foto, video, audio maupun yang lain. Tidak bisa dipungkiri proses kirim mengirim file menjadi rutinitas keseharian masyarakat, dengan banyaknya file yang dikirim akan sebanding dengan kebutuhan tempat penyimpanan yang besar untuk menyimpan file-file tersebut. Metode kompresi citra digital dapat meminimalkan ukuran file citra menjadi lebih kecil, salah satu metode citra digital adalah metode DWT (Discrete Wavelet Transform) dan metode Huffman. Dengan menggunakan kompresi citra digital akan memberikan solusi untuk memperkecil ukuran file citra sehingga kebutuhan memori lebih rendah dan bisa dimaksimalkan. Tujuan dari penelitian ini adalah untuk mendapatkan hasil perbandingan antara metode DWT (Discrete Wavelet Transform) dan metode Huffman dalam melakukan kompresi citra digital. Kedua algoritma menerapkan teknik Lossless Compression dimana hasil kompresi citra hampir sama dengan citra aslinya, tidak ada informasi didalam citra yang hilang namun mengurangi ukuran memorinya. Dari penelitian yang dilakukan dengan 10 data citra uji, memperoleh hasil bahwa metode Huffman menghasilkan nilai kualitas citra (PSNR), MSE lebih baik daripada metode DWT sedangkan untuk rasio kompresi metode DWT lebih unggul.*

**Kata kunci :** Kompresi, Citra, Lossless, DWT, Huffman.

## 1. PENDAHULUAN

Perkembangan teknologi informasi pada era digitalisasi sekarang ini pada berbagai sektor kehidupan masyarakat baik sektor perekonomian, industri, pariwisata, politik dan sebagainya mengharuskan masyarakat untuk siap menerima berbagai macam teknologi yang berkembang tersebut. Salah satu perkembangan teknologi informasi yang saat ini sudah menjadi bagian aktivitas keseharian dari masyarakat digital adalah pertukaran informasi berupa file, baik itu file foto, video, audio maupun yang lain. Berdasarkan hasil riset Wearesosial Hootsuite yang dirilis Januari 2019 pengguna media sosial di Indonesia mencapai 150 juta atau sebesar 56% dari total populasi. Jumlah tersebut naik 20% dari survei sebelumnya. Sementara pengguna media sosial *mobile (gadget)* mencapai 130 juta atau sekitar 48% dari

populasi. (Katadata.co.id, 2019). Efisiensi penggunaan bandwidth dan memory menjadi satu fokus utama yang harus diperhatikan dalam proses pertukaran data. Kompresi bertujuan untuk mengurangi jumlah data yang digunakan untuk mewakili isi file teks, gambar, audio, dan video tanpa mengurangi kualitas data aslinya. Ada dua metode utama kompresi, yaitu metode lossless dan metode lossy.

Metode kompresi akan mengecilkan ukuran file dari ukuran aslinya, sehingga kebutuhan tempat penyimpana akan relatif lebih kecil/berkurang. Pada penelitian ini penulis akan membandingkan metode kompresi citra digital DWT (Discrete Wavelet Transform) dan metode Huffman. Dengan menggunakan kompresi citra digital akan memberikan solusi untuk memperkecil ukuran file citra sehingga kebutuhan memori lebih rendah dan bisa dimaksimalkan.

## 2. LANDASAN TEORI

### 2.1. Kompresi Citra

Dua teknik mendasar dalam melakukan kompresi, yaitu lossless compression yang mengandalkan keutuhan informasi setelah dilakukan kompresi citra. Kelemahannya adalah rasio kompresi citra metode ini sangat rendah. Teknik kedua adalah lossy compression yang mengutamakan rasio kompresi sangat tinggi dengan mengorbankan beberapa informasi yang hilang, tetapi masih bisa ditolelir oleh persepsi mata. Mata tidak dapat membedakan perubahan kecil pada citra.

- a. Efisiensi kompresi : metode kompresi yang efisien adalah metode kompresi yang mampu mengompresi file citra menjadi file yang berukuran paling minimal. Guna menghitung efisiensi kompresi digunakan rasio kompresi (compression ratio) atau disingkat CR. Seperti yang disebutkan oleh persamaan 2.1. sebagai berikut.

$$CR = \frac{C_1}{C_2} \quad (2.1)$$

C1 adalah besar file citra yang belum dikompres, C2 adalah besar file citra setelah terkompres. Terkadang istilah persentase citra yang telah berhasil dimampatkan digunakan untuk mengukur efisiensi kompresi, Seperti yang disebutkan oleh persamaan 2.2. sebagai berikut.

$$\left(1 - \frac{C_1}{C_2}\right) \times 100\% \quad (2.2)$$

- b. Kualitas kompresi (fidelity) : citra hasil rekonstruksi mirip atau sama dengan citra semula tanpa adanya distorsi (informasi yang hilang atau berubah). Kalaupun ada distorsi akibat kompresi sebaiknya ditekan seminimal mungkin. Alat ukur yang sering digunakan adalah Mean Square Error (MSE) seperti pada persamaan 2.3 dan Peak-to-peak Signal to Noise Ratio (PSNR) pada persamaan 2.4 sebagai berikut.

$$MSE = \frac{1}{M \times N} \sum_{i=1}^N \sum_{j=1}^M (f(x, y)' - f(x, y))^2 \quad (2.3)$$

$$PSNR = 20 \times \log_{10} \left( \frac{255}{\sqrt{MSE}} \right) \quad (2.4)$$

M dan N adalah lebar dan tinggi citra. F(x, y)' dan f(x, y) adalah intensitas baris ke-x dan kolom ke-y dari citra hasil kompresi dan citra sebelum kompresi. PSNR mempunyai satuan desiBel (dB). Semakin besar nilai PSNR, maka semakin bagus kualitas citra hasil pemampatan tersebut. Sebaliknya, semakin kecil nilai PSNR, maka kualitas citra hasil pemampatan semakin jelek. Semakin bagus nilai MAE, MSE, RMSE, maka hasil kompresi semakin bagus. (Andono, P. N. dkk., 2017)

### 2.2. Algoritma DWT (Discrete Wavelet Transform)

Kompresi wavelet Haar adalah cara efisien untuk melakukan kompresi citra, baik lossless ataupun lossy. Hal ini bergantung pada nilai rata-rata dan selisih dalam matriks citra untuk menghasilkan matriks yang mempunyai komponen '0' yang banyak. Prinsipnya adalah sebuah matriks citra yang mempunyai komponen '0' banyak, bila disimpan akan membutuhkan memori relatif lebih kecil.

Misalkan diketahui citra A, setelah dilakukan DWT "Haar", hasil dekomposisinya adalah B. Langkah-langkah kompresi dan rekonstruksi:

- a. Lakukan DWT "Harr" pada citra A, dihasilkan citra dekomposisi B.

- b. Pasang *threshold* T pada B dengan ketentuan seperti persamaan 2.5 berikut:

$$C = \begin{cases} B(x, y) & \text{abs}(B) \geq T \\ 0 & \text{lainnya} \end{cases} \quad (2.5)$$

- c. Rekonstruksi citra disimpan di D, ditampilkan dalam persamaan 2.6:

$$D = IDWT(C) \quad (2.6)$$

Contoh kasus sebagai berikut:

Diketahui citra A *grayscale* 8 bit berukuran 4x4 piksel, akan dilakukan kompresi citra menggunakan DWT Harr.

$$A = \begin{bmatrix} 145 & 20 & 180 & 70 \\ 200 & 30 & 210 & 50 \\ 150 & 40 & 220 & 30 \\ 160 & 50 & 198 & 10 \end{bmatrix}$$

Jawab:

DWT A hasilnya adalah

$$\begin{aligned} LL &= \begin{bmatrix} 197,5 & 255 \\ 200 & 229 \end{bmatrix} & LH &= \begin{bmatrix} -32,5 & -5 \\ -10 & 21 \end{bmatrix} \\ HL &= \begin{bmatrix} 147,5 & 135 \\ 110 & 189 \end{bmatrix} & HH &= \begin{bmatrix} -22,5 & -25 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Citra hasil dekomposisi level ke-1:

$$B = \begin{bmatrix} 197,5 & 255 & -32,5 & -5 \\ 200 & 229 & -10 & 21 \\ 147,5 & 135 & -22,5 & -25 \\ 110 & 189 & 0 & 1 \end{bmatrix}$$

Jumlah elemen yang bukan '0' ada 15.

Bila dipasang *threshold* T=15 dan hasilnya disimpan di C, maka:

$$C = \begin{bmatrix} 197,5 & 255 & -32,5 & 0 \\ 200 & 229 & 0 & 0 \\ 147,5 & 135 & 0 & -25 \\ 110 & 189 & 0 & 0 \end{bmatrix}$$

Jumlah elemen yang bukan '0' ada 10, sedangkan jumlah elemen citra semula ada 15. Dari sini diperoleh rasio kompresi:

$$CR = \frac{15}{10} = 1,5$$

Rekonstruksi citra : D = IDWT(C), yaitu:

$$D = \begin{bmatrix} 156,25 & 8,75 & 182,5 & 72,5 \\ 188,75 & 41,25 & 207,5 & 47,5 \\ 155 & 45 & 209,5 & 19,5 \\ 155 & 45 & 208,5 & 20,5 \end{bmatrix}$$

MSE antara citra awal A dan citra hasil rekonstruksi D adalah :

$$MSE = \frac{1}{4 \times 4} \{ (156,25 - 145)^2 + (8,75 - 20)^2 + \dots + (208,5 - 198)^2 + (20,5 - 10)^2 \}$$

$$MSE = 67,0156$$

(Andono, P. N. dkk., 2017)

### 2.3. Algoritma Huffman

Algoritma Huffman, yang dibuat oleh seorang mahasiswa MIT bernama David Huffman pada tahun 1952, merupakan salah satu metode paling lama dan paling terkenal dalam kompresi teks. Algoritma tersebut digunakan untuk membuat kompresi jenis lossy compression, yaitu pemampatan data dimana tidak satu byte pun hilang sehingga data tersebut utuh dan disimpan sesuai dengan aslinya. Pada sejarahnya, Huffman sudah tidak dapat membuktikan apapun tentang kode apapun yang efisien, tapi ketika tugasnya hampir final, ia mendapatkan ide untuk menggunakan pohon binary untuk menyelesaikan masalahnya mencari kode yang efisien. Pada dasarnya, algoritma Huffman ini bekerja seperti mesin sandi morse, dia membentuk suatu kode dari suatu karakter. Sehingga karakter tersebut memiliki rangkaian bit yang lebih pendek dibandingkan sebelumnya. (Suwardi Firdaus, 2018)

Algoritma Huffman menggunakan prinsip pengkodean yang mirip dengan kode Morse, yaitu tiap karakter (simbol) dikodekan hanya dengan rangkaian beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkaian bit yang pendek dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang. Berdasarkan tipe peta kode yang digunakan untuk mengubah pesan awal (isi data yang diinputkan) menjadi sekumpulan codeword, algoritma Huffman termasuk kedalam kelas algoritma yang menggunakan metode static. Metoda static adalah metoda yang selalu menggunakan peta kode yang sama, metoda ini membutuhkan dua fase (two-pass).

Fase pertama untuk menghitung probabilitas kemunculan tiap simbol dan menentukan peta kodenya, dan fase kedua untuk mengubah pesan menjadi kumpulan kode yang akan ditransmisikan. Sedangkan berdasarkan teknik pengkodean simbol yang digunakan, algoritma Huffman menggunakan metode symbolwise. Metoda symbolwise adalah metoda yang menghitung peluang kemunculan dari setiap simbol dalam satu waktu, dimana simbol yang lebih sering muncul diberi kode lebih pendek dibandingkan simbol yang jarang muncul.

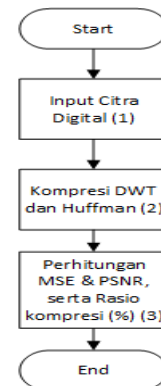
Proses encoding untuk satu karakter dimulai dengan membuat Huffman Tree terlebih dahulu. Setelah itu, kode untuk satu karakter dibuat dengan menyusun nama string biner yang dibaca dari akar sampai ke daun Huffman Tree. Hasil dari penyusunan nama string biner tersebut menjadi kode biner yang baru untuk setiap karakter. Karakter yang memiliki frekuensi kemunculan yang besar akan memiliki kode biner yang lebih pendek dari pada karakter yang memiliki frekuensi kemunculan yang kecil.

Decoding merupakan kebalikan dari encoding. Decoding berarti menyusun kembali data dari string biner menjadi sebuah karakter kembali. Decoding dapat dilakukan dengan dua cara, yang pertama dengan menggunakan Huffman Tree dan yang kedua dengan menggunakan tabel kode Huffman. Melakukan proses decoding dengan menggunakan Huffman Tree dilakukan dengan cara menelusuri Huffman Tree dan mencatat string biner yang ada pada cabang sampai dengan karakter yang dicari ditemukan. Sedangkan menggunakan tabel kode Huffman adalah dengan cara membuat sebuah tabel yang berisi karakter yang diencoding beserta string biner yang baru dibentuk dari Huffman Tree. (Dharmawan, G., 2008)

### 3.1 Desain dan Pembuatan Program

#### a) Flowchart Program

Algoritma proses kerja dari sistem rekomendasi yang akan dibangun terlihat pada Gambar 3.1.



Gambar 3.1 Flowchart Program

1. Input citra digital  
Pada tahap ini, yang pertama kali dilakukan adalah menjalankan program yang telah dibuat pada Google Collab. Setelah itu dilakukan proses input data berupa citra digital yang telah dipersiapkan. Total file citra yang diuji adalah berjumlah 30 buah.
2. Kompresi DWT dan Huffman  
Pada tahap ini, dilakukan proses kompresi menggunakan dua buah algoritma secara

bergantian untuk masing-masing file citra. Pada sistem ini masing-masing algoritma terpisah antara algoritma DWT dan Huffman dengan nama file `dwtImageFix.ipynb` untuk algoritma DWT dan `huffmanFix.ipynb` untuk algoritma Huffman.

3. Penghitungan MSE dan PSNR serta rasio kompresi (%)

Pada tahap ini dilakukan proses penghitungan nilai dari MSE dan PSNR untuk masing-masing algoritma. Penghitungan tersebut dilakukan dengan cara membandingkan nilai dari citra hasil kompresi yang telah selesai melewati proses kompresi dan dibandingkan dengan citra sebelum melalui proses kompresi untuk dilihat nilai MSE dan PSNR dan dapat dilihat mana yang lebih baik. Nilai MSE dan PSNR digunakan untuk proses analisis citra kompresi sehingga dapat dibandingkan dan dapat pula menentukan algoritma terbaik untuk kompresi data citra. Selain tahap penghitungan nilai MSE dan PSNR, terdapat proses rasio kompresi dalam bentuk persen (%) yang didapat dari membandingkan besar ukuran file citra hasil proses kompresi dengan besar ukuran file citra sebelum proses kompresi dalam bentuk persen (%) sehingga, dapat dibandingkan dengan file citra lainnya.

b) Pembuatan Program

Sistem ini akan dibangun dalam bentuk aplikasi pembelajaran dalam bentuk Google Colab. Bahasa pemrograman yang digunakan berupa Bahasa Pemrograman Python, kemudian dengan *software open source* dari Google yaitu Google Collaboratory

### 3.2 Implementasi Sistem

Penelitian perbandingan metode DWT (*Discrete Wavelet Transform*) dan metode Huffman terhadap kompresi citra digital yang dilakukan diharapkan dapat menghasilkan kesimpulan yang sesuai, dengan melakukan penelitian ini penulis dapat memperoleh data hasil penelitian berupa perbandingan metode DWT dengan metode Huffman untuk kompresi citra digital baik dari segi galat error MSE dan PSNR maupun rasio hasil kompresi, yang diterapkan dalam 30 data uji terbagi atas masing-masing 10 data uji untuk tiap-tiap format citra digital antara lain .BMP, .JPG dan .PNG.

### 3.3 Pengujian Sistem

Hasil pengujian terhadap citra data uji yang telah dilakukan menunjukkan perbedaan pada setiap

satu data uji yang diuji hal ini berlaku untuk masing-masing metode pengujian, perbedaan tersebut terletak baik pada rasio kompresi, MSE maupun PSNR.

Dilakukan analisis terhadap semua parameter yang telah ditetapkan yang hasil uji ini digunakan untuk menarik kesimpulan pada penelitian ini. Sumber keseluruhan citra uji berasal dari citra dari search engine google dengan kata kunci : “citra hewan jpg 800 x 600” untuk format citra .jpg dan “citra png 800 x 600” untuk format citra .png. Untuk data uji citra dengan format .bmp tidak ditemukan citra dengan search engine google, oleh karena itu untuk mendapatkan citra data uji dengan format .bmp yaitu dengan mengkonvert citra hewan format .jpg menjadi format .bmp sehingga didapatkan data uji citra dengan format .bmp.

### 3.4 Perangkat Pendukung Penelitian

Adapun perangkat pendukung berupa perangkat lunak (*Software*) dan perangkat keras (*Hardware*) dengan spesifikasi sebagai berikut:

- a) Kebutuhan Perangkat Lunak (*Software*)
  - Sistem Operasi : Windows 10 Education
  - Pengelola Kata : MS. Office Word 2013
  - Editor dan Engine : Google Collaboratory
  - Bahasa Pemrograman : Python
- b) Kebutuhan Perangkat Keras (*Hardware*)
  - Merk Komputer : HP Notebook 14-AC003TX
  - Penyimpanan : 500 GB Hardisk
  - RAM : 4096 MB
  - Processor : Processor Intel(R) Core(TM) i5-5200U 2.20GHz

## 4. HASIL DAN PEMBAHASAN

Optimasi dan sekaligus menjawab permasalahan terkait kebutuhan memori, untuk menjawab permasalahan tersebut rasio kompresi menjadi solusi yaitu dengan hasil rasio kompresi yang tinggi menunjukkan tingkat kebutuhan memori semakin berkurang.

Berikut ini ditampilkan rata-rata hasil pengujian yang ditampilkan pada Tabel 4.1 seperti berikut.

Rata-rata (30 data uji)	Metode	Rasio (%)		
		.BMP	.JPG	.PNG
	DWT	10.9230432075	10.6847973028	-18.8036111147
	Huffman	5.5276944152	4.92804617657	61.6940729168

Selanjutnya untuk melihat tingkat kualitas citra dapat diperoleh dari nilai MSE dan PSNR, berikut hasil MSE dan PSNR untuk metode DWT ditampilkan pada Tabel 4.2 berikut.

Metode	.BMP		.JPG		.PNG	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
DWT	338.03	32.47339 35813	344.166	32.294659 9197	1615.2 89	34.726 441935 1
Huffman	174.9839 35439	50.25015 24957	145.44491 4216	48.112808 1142	263.78 421	73.999 996066 2

Dari hasil rata-rata dari 30 data citra uji, menghasilkan nilai MSE dari kedua algoritma. Nilai MSE yang baik adalah memiliki nilai paling rendah, karena semakin rendah nilai MSE maka nilai rata-rata kerusakan tidak besar dan citra tidak banyak kehilangan informasi. Sedangkan nilai PSNR yang baik ialah memiliki nilai tertinggi, karena semakin tinggi nilai PSNR maka nilai kualitas semakin tinggi dan citra terkompresi semakin mendekati citra aslinya.

## 5. PENUTUP

### 5.1. Kesimpulan

Berdasarkan dari penelitian yang sudah dilakukan serta uraian rumusan masalah yang telah dijelaskan pada bab-bab sebelumnya, maka dapat disimpulkan yaitu tingkat efisiensi memori file hasil kompresi diukur dari besarnya rasio kompresi yang dihasilkan. Semakin besar nilai rasio kompresi menandakan semakin efisien memori hasil kompresi.

Algoritma DWT memiliki nilai rata-rata rasio kompresi lebih baik untuk format .bmp sebesar 10.92% serta .jpg sebesar 10.68 % akan tetapi untuk format .png algoritma Huffman lebih baik dengan nilai sebesar 61.69%. Semakin rendah nilai MSE menandakan semakin baik kualitas citra terkompresi, Semakin tinggi nilai PSNR maka semakin baik kualitas citra terkompresi. Untuk rasio kompresi metode DWT tipe citra .bmp dan .jpg rasio lebih baik daripada metode Huffman tetapi untuk perhitungan MSE dan PSNR metode Huffman tingkat kualitas citra lebih baik.

Berdasarkan permasalahan penelitian mengenai efisiensi media penyimpanan dan transmisi citra digital, maka dapat disimpulkan bahwa algoritma yang tepat untuk dipilih adalah algoritma dengan nilai rasio kompresi lebih baik yaitu algoritma DWT sedangkan untuk kualitas citra yaitu metode Huffman.

### 5.2. Saran

Berdasarkan kesimpulan di atas, peneliti mencantumkan beberapa saran antara lain:

- Dapat dikembangkan dengan menambahkan metode kompresi lainnya.
- Dapat dikembangkan dengan basis maupun platform yang lain misalnya berbasis mobile (android/iOs).

## DAFTAR PUSTAKA

- Andono, P.N., T.Sutojo dan Muljono (2017), *Pengolahan Citra Digital*, A. Pramesta, Ed. ed. 1 Yogyakarta: CV. Andi Offset.
- Imam Digmi (2018), *Google Colab Gratis Untuk Belajar Deep Learning- Journal Today*.
- Katadata.co.id (2019), *Berapa Pengguna Media Sosial Indonesia?/Databoks*, (<https://databoks.katadata.co.id/datapublish/2019/02/08/berapa-pengguna-media-sosial-indonesia>) akses 28 April 2019.
- Prayoga, E. dan Suryaningrum, K.M. (2018), *Implementasi Algoritma Huffman Dan Run Length Encoding Pada Aplikasi Kompresi Berbasis Web*, *Jurnal Ilmiah Teknologi Informasi Terapan*, IV(2), 92–101.
- Putra, D. (2010), *Pengolahan Citra Digital*, Westriningsih, Ed. ed. 1 Yogyakarta: Andi Offset.
- Rustamaji, H.C., Mariani dan Yuwono, B. (2014), *Aplikasi Kompresi Data Menggunakan Metode Huffman Statik Pada Perangkat Mobile Berbasis Android*, *Telematika*, 11(1), 9–18.
- Suma'inna dan Alam, D. (2014), *Kompresi Citra Berwarna Menggunakan Transformasi Wavelet*, *Jurnal Matematika Integratif*, 10(1), 55–62.
- Wibowo, A. (2012), *Kompresi Data Menggunakan Metode Huffman*, *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*, 2(1), 47–51.
- Widhiarta, P. (2008), *Pengantar Kompresi Data*, Surabaya.