

# Analisis Kinerja Kompresi *Huffman* dan BWT pada Steganografi dalam Reduksi Ukuran Gambar

Dikri Maulana<sup>1</sup>, Alam Rahmatulloh<sup>2</sup>

Jurusan Informatika, Fakultas Teknik, Univeristas Siliwangi, Jl. Mугarsari, Kec. Tamansari, Kota Tasikmalaya, Jawa Barat 46196, Indonesia<sup>1,2</sup>

217006111@student.unsil.ac.id<sup>1</sup>, alam@unsil.ac.id<sup>2</sup>

**Abstrak** – Keamanan data menjadi isu krusial dalam era digital, terutama ketika data sensitif harus dikirim atau disimpan di lingkungan yang tidak aman. Penelitian ini bertujuan untuk menganalisis kinerja kombinasi metode kriptografi *Advanced Encryption Standard* (AES), steganografi *End of File* (EOF), serta kompresi *lossless* dalam reduksi ukuran gambar. Tiga skenario diuji: (1) tanpa kompresi, (2) dengan kompresi *Huffman Encoding*, dan (3) dengan kompresi *Burrows-Wheeler Transform* (BWT). Proses dimulai dengan kompresi *main image* menggunakan algoritma *Burrows-Wheeler Transform* (BWT) atau *Huffman Encoding*, dilanjutkan dengan enkripsi menggunakan AES, dan diakhiri dengan penyisipan gambar utama yang telah terenkripsi ke dalam *cover image* menggunakan metode steganografi EOF. Hasil menunjukkan bahwa ukuran *file* tanpa kompresi rata-rata mencapai 1.265.673 bytes. Penggunaan metode kompresi *Huffman* menghasilkan pengurangan ukuran *file* rata-rata sebesar 1,39%, dengan ukuran *file* rata-rata menjadi 1.230.213 bytes. Sementara itu, metode BWT menunjukkan kinerja terbaik, dengan pengurangan ukuran rata-rata sebesar 5,48% dan ukuran *file* rata-rata sebesar 1.187.186 bytes. Temuan ini menunjukkan bahwa meskipun *Huffman* dapat mengurangi ukuran *file*, metode BWT jauh lebih efektif dalam mengidentifikasi pola data, sehingga menghasilkan *file* yang lebih kecil.

**Kata Kunci** – AES, *Burrows-Wheeler Transform*, EOF, *Huffman Encoding*, Kriptografi, Steganografi

## I. PENDAHULUAN

Seiring dengan pesatnya perkembangan teknologi informasi, cara penyimpanan dan transfer data dalam dunia digital mengalami transformasi besar. Meskipun kemajuan ini membawa berbagai keuntungan, tantangan baru muncul terkait dengan perlindungan privasi dan keamanan data yang semakin rentan terhadap ancaman yang terus berkembang. Salah satu masalah utama yang dihadapi dalam era digital ini adalah bagaimana melindungi data sensitif dari akses pihak yang tidak berwenang, baik selama proses penyimpanan maupun transmisi. Kriptografi dan steganografi adalah dua teknik utama yang sering digunakan untuk mengatasi masalah ini [1], [2].

Kriptografi adalah metode yang mengubah data asli (*plaintext*) menjadi bentuk terenkripsi (*ciphertext*) menggunakan algoritma tertentu sehingga data tersebut tidak dapat dibaca tanpa kunci dekripsi [3], [4]. Salah satu algoritma kriptografi yang populer dan luas digunakan

adalah *Advanced Encryption Standard* (AES)[5]. AES dikenal karena memiliki tingkat keamanan yang tinggi serta kecepatan proses enkripsi dan dekripsinya. AES merupakan algoritma kunci simetris, yang berarti menggunakan kunci yang sama untuk proses enkripsi dan dekripsi [6]. Hal ini membuat AES menjadi pilihan utama dalam berbagai aplikasi keamanan digital yang membutuhkan efisiensi waktu dan sumber daya. Meskipun enkripsi AES dapat melindungi isi data dari pihak ketiga, keberadaan data terenkripsi masih dapat diketahui. Oleh karena itu, steganografi digunakan sebagai lapisan tambahan dalam pengamanan data. Steganografi adalah teknik untuk menyembunyikan informasi rahasia di dalam media digital, seperti gambar atau audio, sehingga keberadaan informasi tersebut tidak mudah terdeteksi oleh pihak yang tidak berwenang [7]. Salah satu metode steganografi yang paling umum adalah *End of File* (EOF),

di mana data disisipkan di bagian akhir *file* gambar. Metode ini cenderung tidak mempengaruhi kualitas visual gambar secara signifikan, sehingga data tetap tersembunyi dengan baik.

Namun, teknik steganografi EOF memiliki kelemahan. Salah satu kelemahannya adalah peningkatan ukuran *file* yang terjadi setelah data disisipkan, terutama jika ukuran data yang disisipkan cukup besar [8], [9], [10]. Hal ini dapat memengaruhi efisiensi penyimpanan dan transmisi data, terutama ketika ukuran *file* menjadi terlalu besar. Oleh karena itu, diperlukan suatu mekanisme tambahan untuk mengoptimalkan ukuran *file* setelah penyisipan data.

Kompresi *lossless* merupakan solusi yang tepat untuk mengatasi masalah peningkatan ukuran *file* setelah proses steganografi. Kompresi *lossless* memungkinkan pengurangan ukuran *file* tanpa kehilangan data asli [11]. Ada beberapa algoritma kompresi *lossless* yang efektif, salah satunya adalah *Huffman Encoding* dan *Burrows-Wheeler Transform* (BWT). *Huffman Encoding* bekerja dengan memberi kode lebih pendek untuk simbol yang sering muncul dan kode lebih panjang untuk simbol yang jarang muncul, sehingga menghasilkan pengurangan ukuran data secara efisien [12]. Di sisi lain, BWT tidak langsung mengompresi data, tetapi mengubah urutan data sehingga simbol yang sama dikelompokkan bersama, memudahkan proses kompresi lebih lanjut oleh algoritma seperti *Huffman Encoding* [13].

Berdasarkan pernyataan – pernyataan tersebut. Penelitian ini menganalisis kinerja metode pengamanan data yang menggabungkan kriptografi AES, steganografi EOF, serta kompresi *lossless*. Analisis dilakukan dengan membandingkan tiga skenario berbeda: tanpa kompresi, dengan kompresi *Huffman Encoding* dan dengan kompresi *Burrows-Wheeler Transform* (BWT).

Tujuan perbandingan ini adalah untuk mengevaluasi efektivitas setiap metode dalam mengurangi ukuran *file* sekaligus mempertahankan keamanan dan integritas data.

## II. TINJAUAN PUSTAKA

Penelitian dengan judul “Transform Domain Video Steganography Using RSA,

Random DNA Encryption and *Huffman Encoding*”[14]. Penelitian ini bertujuan menggabungkan algoritma RSA, enkripsi DNA acak, pengkodean *Huffman*, dan steganografi DCT untuk komunikasi rahasia yang lebih aman. Hasil yang ditunjukkan tidak disebutkan secara spesifik nilai rasio kompresi dari pengkodean *Huffman* yang digunakan dalam sistem yang diusulkan. Namun, pengkodean *Huffman* umumnya dikenal untuk memberikan rasio kompresi yang baik, tergantung pada karakteristik data yang dikompresi. Penelitian yang lain melakukan analisis dan implementasi kompresi *file* citra menggunakan algoritma *Burrows Wheeler Transform* (BWT). Rasio kompresi dari penggunaan *Burrows Wheeler Transform* (BWT) dalam penelitian tersebut adalah sebesar 69.5%. Ini berarti bahwa 69.5% dari citra asli berhasil dipadatkan, sehingga ukuran citra yang berkurang adalah dari 72 bit menjadi 22 bit, menghasilkan pengurangan sebesar 50 bit [15]. Penelitian selanjutnya melakukan perbandingan metode *Huffman* dan *Run Length Encoding* pada kompresi *file* dokumen. Rasio kompresi yang dihasilkan oleh *Huffman Encoding* pada dokumen bervariasi tergantung pada jenis dokumen tersebut. Dalam penelitian tersebut, rasio kompresi untuk *file Word* adalah sekitar 1,24%, untuk *file Excel* sekitar 1,06%, dan untuk *file PDF* sekitar 1%[16].

## III. ANALISA DAN PERANCANGAN SISTEM

Langkah-langkah metode penelitian dalam studi ini telah dirancang dan divisualisasikan untuk memberikan gambaran yang jelas tentang alur penelitian dari tahap awal hingga analisis akhir.



Gambar 1 Metode Penelitian

Pada Gambar 1 menunjukkan tahapan yang ditempuh, mulai dari studi literatur, pemilihan data sampel, hingga evaluasi hasil kompresi dengan pendekatan eksperimental pada tiga metode kompresi data yang berbeda.

### A. Studi Literatur

Studi pustaka dilakukan untuk mendapatkan informasi dengan cara membaca literatur dan referensi yang sesuai dan mendukung melalui media internet, jurnal, dan buku.

#### 1. Kriptografi *Advanced Encryption Standard* (AES)

Algoritma *Advanced Encryption Standard* (AES) merupakan proses algoritma yang cepat dan kuat[17]. Tiga varian utama algoritma AES adalah AES-128, AES-192, dan AES256, dan ketiganya memiliki kualitas enkripsi yang lebih baik dibandingkan dengan algoritma RSA[6]. Setiap tipe menggunakan kunci internal yang berbeda, yaitu kunci bulat, untuk setiap proses putaran. Dalam penelitian ini, algoritma AES-128 digunakan, dan proses putaran enkripsinya dilakukan sepuluh kali ( $a=10$ ).

#### 2. Steganografi *End of File* (EOF)

Steganografi *End of File* (EOF) adalah salah satu teknik steganografi yang menyisipkan data pada bagian akhir sebuah *file* gambar tanpa memodifikasi struktur atau konten utama *file* tersebut[18]. Pesan yang disisipkan menggunakan metode steganografi EOF memang tidak terbatas jumlahnya, karena pada dasarnya, data yang disisipkan ditempatkan di bagian akhir *file* tanpa mengganggu struktur aslinya. Selama ada ruang memori yang cukup untuk menampung *file* tersebut, data dapat disisipkan sebanyak yang diinginkan. Namun, ada efek samping utama dari teknik ini, yaitu ukuran berkas akan menjadi lebih besar dari ukuran aslinya.

#### 3. Kompresi *Huffman Encoding*

*Huffman Encoding* adalah algoritma kompresi *lossless* yang didasarkan pada frekuensi simbol. Simbol yang sering muncul diberi kode yang lebih pendek, sedangkan simbol yang jarang muncul diberi kode lebih Panjang. Berikut adalah langkah-langkah metode kompresi menggunakan algoritma :

- Analisis Data: Langkah pertama adalah membuat tabel frekuensi yang mencatat berapa kali setiap simbol ASCII muncul. Tabel ini terdiri dari dua kolom, yaitu simbol ASCII dan frekuensi kemunculannya.

- Pemilihan Simpul Awal: Dua simbol dengan frekuensi kemunculan terendah dipilih untuk menjadi simpul pertama pada pohon *Huffman*.
- Pembuatan Simpul Induk: Kedua simpul tersebut digabungkan menjadi satu simpul induk, yang menyimpan jumlah frekuensi dari kedua simpul awal tersebut.
- Penggantian Simpul: Kedua simpul yang sudah digabungkan dihapus dari tabel, lalu simpul induk tadi dimasukkan ke dalam tabel untuk menggantikan mereka. Simpul induk ini akan digunakan sebagai acuan untuk membentuk pohon *Huffman*.
- Pengulangan Proses: Langkah ini diulangi sampai hanya tersisa satu simpul dalam tabel. Simpul terakhir ini akan menjadi simpul akar pohon *Huffman*.
- Penandaan Cabang: Setiap simpul pada cabang kiri (dengan frekuensi lebih besar) diberi nilai 0, sementara simpul pada cabang kanan (dengan frekuensi lebih kecil) diberi nilai 1.
- Proses Pembacaan: Pembacaan dilakukan dari simpul akar hingga mencapai simpul daun dengan mengikuti nilai pada setiap cabang yang dilalui.

#### 4. Kompresi *Burrows-Wheeler Transform* (BWT)

Algoritma *Burrows-Wheeler Transform* (BWT) adalah algoritma yang mengubah blok data menjadi susunan baru, di mana karakter yang ada tetap sama, namun urutannya berubah. Transformasi BWT menyusun karakter secara berkelompok sehingga lebih banyak karakter yang sama muncul berurutan, yang pada akhirnya meningkatkan peluang menemukan pola karakter yang berulang[19].

### B. Pemilihan Data Sampel

Tabel 1 menunjukkan lima gambar yang dipilih untuk digunakan sebagai media *cover* dalam sistem pengamanan data. Setiap gambar memiliki ukuran piksel dan ukuran memori yang berbeda-beda.

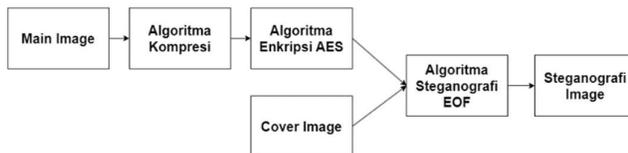
TABEL 1  
SAMPLE DATA

Gambar Cover	Ukuran Pixel	Ukuran Memori (bytes)
Wortel.png	1600 x 1600	524.409 bytes
Mobil.png	1324 x 712	427.664 bytes
Orang.png	1121 x 1564	830.959 bytes
Komputer.png	1080 x 1080	223.814 bytes
Pisang.png	847 x 798	441.903 bytes

Setiap gambar yang ditampilkan dalam tabel di atas akan digunakan sebagai media *cover* untuk menyisipkan *main image* yang bernama **rahasia.png**. Gambar utama ini memiliki ukuran piksel sebesar **2000 pixel x 1334 pixel** dan ukuran *file* sebesar **631.949 bytes**. Proses penyisipan ini akan dilakukan menggunakan metode steganografi, di mana data dari *main image* akan disembunyikan dalam *cover image* yang telah dipilih.

### C. Implementasi Algoritma Kompresi

Pada gambar berikut, ditampilkan alur sistem yang diterapkan dalam penelitian, yang mencakup langkah-langkah proses dari kompresi gambar utama, enkripsi menggunakan algoritma AES, hingga penyisipan data ke dalam *cover image* menggunakan steganografi EOF.



Gambar 2 Desain Alur Sistem

Desain alur sistem pada Gambar 2 menggabungkan teknik kriptografi AES, steganografi EOF, dan kompresi untuk menyembunyikan data dalam *cover image*. Tahap pertama dimulai dengan mengompresi *main image* menggunakan algoritma *Burrows-Wheeler Transform* (BWT) atau *Huffman Encoding*. Kompresi ini dilakukan untuk mengurangi ukuran *main image* sebelum memasuki tahap enkripsi, sehingga proses

penyimpanan dan transmisi menjadi lebih efisien. Setelah itu, *main image* yang telah dikompresi akan dienkripsi menggunakan algoritma AES, menghasilkan *ciphertext* yang aman dan terlindungi dari akses tidak sah. Pada langkah terakhir, *main image* yang telah dienkripsi ini disisipkan ke dalam *cover image* menggunakan metode steganografi EOF. Dengan teknik ini, data disimpan di bagian akhir *file cover image* tanpa mengubah tampilan visualnya, sehingga *cover image* tetap tampak seperti biasa, namun mengandung data yang disisipkan secara aman.

## IV. IMPLEMENTASI DAN PEMBAHASAN

Analisis yang dilakukan dalam penelitian ini telah diimplementasikan ke dalam aplikasi web untuk mempermudah pengguna dalam menjalankan proses enkripsi dan dekripsi.



Gambar 3 Halaman Enkripsi

Gambar 3 menunjukkan halaman aplikasi untuk enkripsi tanpa menggunakan metode kompresi, yang dilengkapi dengan dua *file upload*, yaitu untuk memilih *main image* dan *cover image*. Selain itu, terdapat tombol untuk masuk ke halaman enkripsi menggunakan metode kompresi, yaitu *Huffman Encoding* atau *Burrows-Wheeler Transform* (BWT).



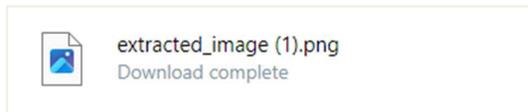
Gambar 4 Hasil Enkripsi

Gambar 4 memperlihatkan hasil dari proses enkripsi, di mana *main image* yang telah dienkripsi.



Gambar 5 Halaman Dekripsi

Gambar 5 menampilkan halaman aplikasi untuk proses dekripsi, yang memungkinkan pengguna untuk mengembalikan gambar terenkripsi ke bentuk aslinya.



Gambar 6 Hasil Dekripsi

Gambar 6 menunjukkan hasil dari proses dekripsi, di mana gambar yang telah didekripsi kembali ke bentuk gambar utama yang dapat dikenali.

V. PENGUJIAN

Dari ketiga pengujian metode yang dilakukan menghasilkan data sebagai berikut :

TABEL 2  
UJI TANPA KOMPRESI

Cover Image	Ukuran Pixel	Ukuran Cover Image setelah disisipkan (bytes)
Wortel.png	1600 x 1600	1.325.123 bytes
Mobil.png	1324 x 712	1.228.378 bytes
Orang.png	1121 x 1564	1.631.673 bytes
Komputer.png	1080 x 1080	1.024.528 bytes
Pisang.png	847 x 798	1.242.617 bytes

Tabel 2 menunjukkan bahwa tanpa menggunakan metode kompresi, gambar-gambar tersebut disimpan dalam ukuran asli mereka. Ini berarti semua data piksel disimpan tanpa pengolahan atau pengurangan ukuran.

TABEL 3  
UJI KOMPRESI HUFFMAN

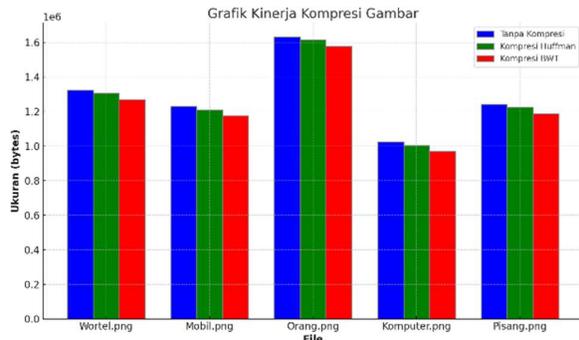
Cover Image	Ukuran Pixel	Ukuran Cover Image setelah disisipkan (bytes)
Wortel.png	1600 x 1600	1.306.511 bytes
Mobil.png	1324 x 712	1.209.766 bytes
Orang.png	1121 x 1564	1.613.061 bytes
Komputer.png	1080 x 1080	1.005.916 bytes
Pisang.png	847 x 798	1.224.005 bytes

Tabel 3 menunjukkan bahwa kompresi Huffman menghasilkan ukuran file yang lebih kecil dibandingkan dengan ukuran tanpa kompresi dengan pengurangan ukuran 1,39% dibandingkan tanpa kompresi.

TABEL 4  
UJI KOMPRESI BWT

Cover Image	Ukuran Pixel	Ukuran Cover Image setelah disisipkan (bytes)
Wortel.png	1600 x 1600	1.270.307 bytes
Mobil.png	1324 x 712	1.173.562 bytes
Orang.png	1121 x 1564	1.576.857 bytes
Komputer.png	1080 x 1080	969.712 bytes
Pisang.png	847 x 798	1.187.801 bytes

Tabel 4 menunjukkan bahwa kompresi BWT menunjukkan hasil yang paling mengesankan dengan pengurangan rata-rata 5,48% dibandingkan dengan ukuran tanpa kompresi.



Gambar 7 Grafik Hasil

Berikut pada gambar 7 adalah grafik hasil kinerja kompresi gambar berdasarkan data yang ada. Grafik ini menunjukkan ukuran *file* dalam *bytes* untuk masing-masing gambar setelah diterapkan tiga metode kompresi: tanpa kompresi, kompresi *Huffman*, dan kompresi *BWT*. Kinerja kompresi data pada lima file gambar dengan tiga metode berbeda, terlihat adanya variasi efektivitas dalam pengurangan ukuran file. Pada metode tanpa kompresi, ukuran file rata-rata mencapai 1.265.673 bytes, menunjukkan bahwa tidak terjadi pengurangan ukuran sama sekali. Sementara itu, metode kompresi *Huffman* mampu memberikan sedikit pengurangan dengan rata-rata sebesar 1,39%, sehingga menjadi pilihan yang lebih efisien dibandingkan tanpa kompresi, meskipun pengurangannya masih terbatas. Metode kompresi *Burrows-Wheeler Transform* (*BWT*) menunjukkan kinerja paling optimal, dengan rata-rata pengurangan ukuran file sebesar 5,48%. Hasil dari *BWT* lebih optimal dari penelitian yang melakukan perbandingan metode *Huffman* dan *Run Length Encoding* pada kompresi *file* dokumen. Rasio kompresi yang dihasilkan oleh *Huffman Encoding* pada dokumen rasio kompresi untuk *file Word* adalah sekitar 1,24%, untuk *file Excel* sekitar 1,06%, dan untuk *file PDF* sekitar 1% [16].

## VI. KESIMPULAN

Penelitian ini menunjukkan adanya perbedaan efektivitas dalam pengurangan ukuran file di antara tiga metode kompresi data yang diuji. Metode tanpa kompresi menghasilkan ukuran file terbesar karena tidak ada pengurangan data. Metode kompresi *Huffman* mampu mengurangi ukuran file meskipun hasilnya masih terbatas, sementara

*Burrows-Wheeler Transform* (*BWT*) menunjukkan kinerja paling optimal dengan ukuran file yang lebih kecil. Temuan ini mengarah pada saran agar penelitian selanjutnya mengeksplorasi kombinasi metode, seperti *Huffman* dan *BWT*, untuk meningkatkan efisiensi kompresi data.

## REFERENSI

- [1] W. Rista Maya, T. Komputer, S. Triguna Dharma, and S. Komputer, "Implementasi Kriptografi Pengamanan Data Nilai Siswa Menggunakan Algoritma DES," *Jurnal Sains Manajemen Informatika dan Komputer*, vol. 21, no. 1, pp. 1–9, 2022, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jis>
- [2] D. Ramalinda and A. Rachmat Raharja, "Strategi Perlindungan Data Menggunakan Sistem Kriptografi Dalam Keamanan Informasi," *Journal of International Multidisciplinary Research*, 2024, [Online]. Available: <https://journal.banjaresepacific.com/index.php/jimr>
- [3] Yusfrizal, "RANCANG BANGUN APLIKASI KRIPTOGRAFI PADA TEKS MENGGUNAKAN METODE REVERSE CHIPER DAN RSA BERBASIS ANDROID," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 3, no. 2, 2019.
- [4] A. B. Nasution, "IMPLEMENTASI PENGAMANAN DATA DENGAN MENGGUNAKAN ALGORITMA CAESAR CIPHER DAN TRANSPOSISI CIPHER," *Jurnal Teknologi Informasi*, vol. 3, no. 1, 2019.
- [5] Galih Yuga Pangestu, Asep Id Hadiana, and Puspita Nurul Sabrina, "Kriptografi Untuk Enkripsi Ganda Pada Gambar Menggunakan Algoritma AES (Advanced Encryption Standard) Dan RC5 (Rivest Code 5)," *INFORMATICS AND DIGITAL EXPERT (INDEX)*, vol. 4, no. 1, pp. 25–32, 2022, [Online]. Available: <https://e-journal.unper.ac.id/index.php/informatika>

- [6] B. E. Widodo and A. S. Purnomo, "IMPLEMENTASI ADVANCED ENCRYPTION STANDARD PADA ENKRIPSI DAN DEKRIPSI DOKUMEN RAHASIA DITINTEKAM POLDA DIY," *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 2, pp. 69–77, Dec. 2020, doi: 10.20884/1.jutif.2020.1.2.21.
- [7] I. Pujiyanto and D. Darwis, "UJI KETAHANAN CITRA DIGITAL TERHADAP MANIPULASI ROBUSTNESS PADA STEGANOGRAPHY," 2021. [Online]. Available: <http://jim.teknokrat.ac.id/index.php/informatika>
- [8] A. D. Cahyono, M. Yasin, and U. N. Malang, "Implementasi steganografi menggunakan metode end of file (EOF) dalam pengamanan data (Studi kasus pada file AVI, MP3, dan JPEG)." [9] Toni Sahara Pandapotan and Taronisokhi Zebua, "Analisis Perbandingan Least Significant Bit dan End of File Untuk Steganografi Citra Digital Menggunakan Matlab," *Prosiding Seminar Nasional Inovasi dan Teknologi Informasi SNITI*, pp. 11–12, 2016.
- [10] A. Rohmanu, "IMPLEMENTASI KRIPTOGRAFI DAN STEGANOGRAFI DENGAN METODE ALGORITMA DES DAN METODE END OF FILE," *Jurnal Informatika SIMANTIK*, vol. 2, no. 1, 2017, [Online]. Available: [www.jurnal.stmikcikarang.ac.id](http://www.jurnal.stmikcikarang.ac.id)
- [11] Devi Ayu Lestari, "ANALISIS PERBANDINGAN ANTARA METODE KOMPRESI DATA UNTUK PENGURANGAN UKURAN FILE DALAM SISTEM KOMPUTER," *Jurnal Teknologi Pintar*, vol. 3, no. 11, pp. 2023–2024, 2023.
- [12] S. Shree Narashiman and N. Chandrachoodan Professor, "NEURAL NETWORK-ENHANCED LOSSLESS TEXT COMPRESSION," *AlphaZip*, 2023.
- [13] D. Cenzato and Z. Lipták, "A survey of BWT variants for string collections," Jul. 01, 2024, *Oxford University Press*. doi: 10.1093/bioinformatics/btae333.
- [14] S. Mumthas and A. Lijiya, "Transform Domain Video Steganography Using RSA, Random DNA Encryption and Huffman Encoding," in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 660–666. doi: 10.1016/j.procs.2017.09.152.
- [15] A. S. Harahap, "Analisis Dan Implementasi Kompresi File Citra Menggunakan Algoritma Burrows Wheeler Transform," *Jurnal Sains Dan Teknologi Informasi*, vol. 1, no. 1, pp. 6–12, 2021, doi: 10.47065/jussi.v1i1.962.
- [16] B. Hari Prasetyo and D. Prabowo, "Perbandingan Metode Huffman dan Run Length Encoding Pada Kompresi Document," *INFOTEKJAR: JURNAL NASIONAL INFORMATIKA DAN TEKNOLOGI JARINGAN*, vol. 5, no. 1, 2020, doi: 10.30743/infotekjar.v5i1.2892.
- [17] R. Amalia and P. Rosyani, "Implementasi Algoritma AES dan Algoritma XOR pada Aplikasi Enkripsi dan Dekripsi Teks Berbasis Android," *Faktor Exacta*, vol. 11, no. 4, p. 370, Dec. 2018, doi: 10.30998/faktorexacta.v11i4.2878.
- [18] A. Eka Putri, A. Kartikadewi, and L. A. A. Rosyid, "Implementasi Kriptografi Dengan Algoritma Advanced Encryption Standard (AES) 128 Bit Dan Steganografi Menggunakan Metode End Of File (EOF) Berbasis Java Desktop Pada Dinas Pendidikan Kabupaten Tangerang," *Applied Information Systems and Management (AISM)*, vol. 3, 2020.
- [19] R. Syahputra, S. D. Nasution, and A. Fau, "PENINGKATAN RASIO KOMPRESI ALGORITMA RLE UNTUK KOMPRESI TEKS DENGAN MENGGUNAKAN ALGORITMA BWT DAN SEQUITUR," *Jurnal Teknologi Komputer dan Sistem Informasi*, 2023.