

Implementasi Algoritma Yamamoto's Recursive Code Dalam Mengkompresi File Gambar

Lulu Nurhidayanti Nasution

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: lulunurhidayantinasion@gmail.com

Abstrak– Ukuran data yang tersimpan baik itu tersimpan di dalam komputer maupun dalam penyimpanan cloud menjadi salah satu permasalahan dalam dunia komputer. Salah satu solusi terbaik yang dilakukan untuk mengatasi masalah ini adalah dengan menggunakan teknik kompresi. Dalam penelitian ini, penulis menggunakan Algoritma Yamamoto's Recursive Code untuk melakukan kompresi terhadap file gambar. File gambar berekstensi bitmap (bmp) memiliki ukuran lebih kecil dibanding yang berformat jpeg hal ini menimbulkan masalah bila kecepatan unduh penerima lambat atau koneksi internet tidak stabil, informasi tidak bisa langsung diproses dikarenakan waktu transfer data yang terlampaui lama. Untuk itu diperlukan sebuah teknik untuk mengubah ukuran data tersebut menjadi lebih kecil. Teknik ini disebut dengan dikenal dengan kompresi data. Kompresi data adalah suatu proses pengubahan sekumpulan data menjadi bentuk suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan data. Atas dasar tersebut Penerapan Algoritma Yamamoto's Recursive Code Dalam Mengkompresi File Gambar dibuat agar dapat membantu pengguna komputer mengompresi file gambar yang awalnya berukuran besar menjadi ukuran terkecilnya.

Kata Kunci: Kompresi Gambar, Algoritma Yamamoto's Recursive Code

Abstract– The size of the data stored whether it is stored on a computer or in cloud storage is one of the problems in the computer world. One of the best solutions to overcome this problem is to use compression techniques. In this study, the author uses Yamamoto's Recursive Code Algorithm to compress image files. Image files with bitmap (bmp) extension have a smaller size than those in jpeg format, this causes problems if the recipient's download speed is slow or the internet connection is unstable, information cannot be processed immediately because the data transfer time is too long. For that we need a technique to change the size of the data to be smaller. This technique is known as data compression. Data compression is a process of converting a set of data into a form of code to save the need for data storage. On this basis, the application of Yamamoto's Recursive Code Algorithm in Compressing Image Files is made in order to help computer users compress image files that are initially large into their smallest size.

Keywords: Image Compression, Yamamoto's Recursive Code Algorithm

1. PENDAHULUAN

Saat ini kemajuan teknologi yang terus berkembang memudahkan pertukaran informasi di seluruh duniadan saat ini sudah memanfaatkan teknologi cloud computing sebagai media penyimpanan dan sebagai media tempat berbagi file [1][2]. Terbatasnya media penyimpanan (storage) pada suatu cloud storage seperti google drive menjadi suatu masalah jika penggunaanya ingin menyimpan banyak file. Biasanya pengguna cloudstorage menyimpan file-file penting disana, tetapi terkadang banyak yang menyimpan foto. Foto yang sangat banyak diambil melalui kamera handphone terkadang membuat memori pada handphone menjadi penuh, dan hal itu yang mengakibatkan terkadang pengguna handphone memindahkan foto tersebut ke komputer atau laptop, atau solusi lainnya mengupload foto tersebut ke cloud storage karena sayang jika dihapus [3][4]. Masalah yang timbul yaitu kapasitas penyimpanan di handphone dan juga di cloud storage yang terbatas, serta proses upload file ke cloud storage yang lama bisa diatasi jika ukuran file foto dirubah menjadi lebih kecil [5][6]. Masalah file yang banyak atau file yang berukuran besar ini dapat diselesaikan dengan suatu teknik yang dapat mengurangi ukuran suatu data, yaitu kompresi data [7][8] [9].

Saat ini banyak aplikasi kompresi yang tersedia, serta memiliki hasil kinerja yang berbeda-beda pula, tergantung dari algoritma yang digunakan. Ada beberapa algoritma kompresi data, salah satunya yaitu Yamamoto's Recursive Code. Algoritma Yamamoto's Recursive Code termasuk salah satu algoritma kompresi bertipe lossless compression yang berarti ukuran file tidak mengalami kehilangan data ketika dilakukan proses dekompresi.

Hasil penelitian terdahulu yang dilakukan oleh I Gusti Ngurah Jelantik Suryaningrat, Gede Dody Sanjaya, Rosalia Hadi, Ni Luh Gede Pivin Suwirmayanti pada tahun 2018, dengan penelitian Kompresi Citra Digital dengan judul penelitian Kompresi Citra Digital Menggunakan Metode Discrete Cosine Transform sangat baik. Dengan Lossy Compression kompresi citra yang dihasilkan dapat menampilkan bahwa tidak terlihat perbedaannya hasil kompresi dengan citra asli sebelum dikompresi [10].

Putri Fitria pada tahun 2020 melakukan penelitian dengan Penerapan Algoritma Rice Codes pada Aplikasi Kompresi File Gambar bahwa dalam penerapan Algoritma Rice Codes pada Kompresi File Gambar sangat bagus, hasilnya pun tidak berkurang dari file aslinya atau tidak ada pengurangan (isi filenya). Aplikasi yang dirancang telah mampu melakukan kompresi file gambar dengan menggunakan algoritma Rice Codes dan melakukan dekompresi terhadap hasil kompresi dengan algoritma Rice Codes menjadi file asli [11].

Esteria Novebri Simanjuntak melakukan penelitian pada tahun 2021, yaitu Penerapan Algoritma Prefix Code Pada Kompresi File Gambar bahwa proses dekompresi akan dilakukan pada tahapan pengurangan bit akhir gambar dan sistem akan melakukan proses dekompresi dengan menggunakan Algoritma Prefix Code. Pada form kompresi sistem akan melakukan proses kompresi dengan cara memilih gambar yang akan dikompresi dengan menggunakan Algoritma Prefix Code dan akan tersimpan sesuai tempat penyimpanan yang dipilih. [12]

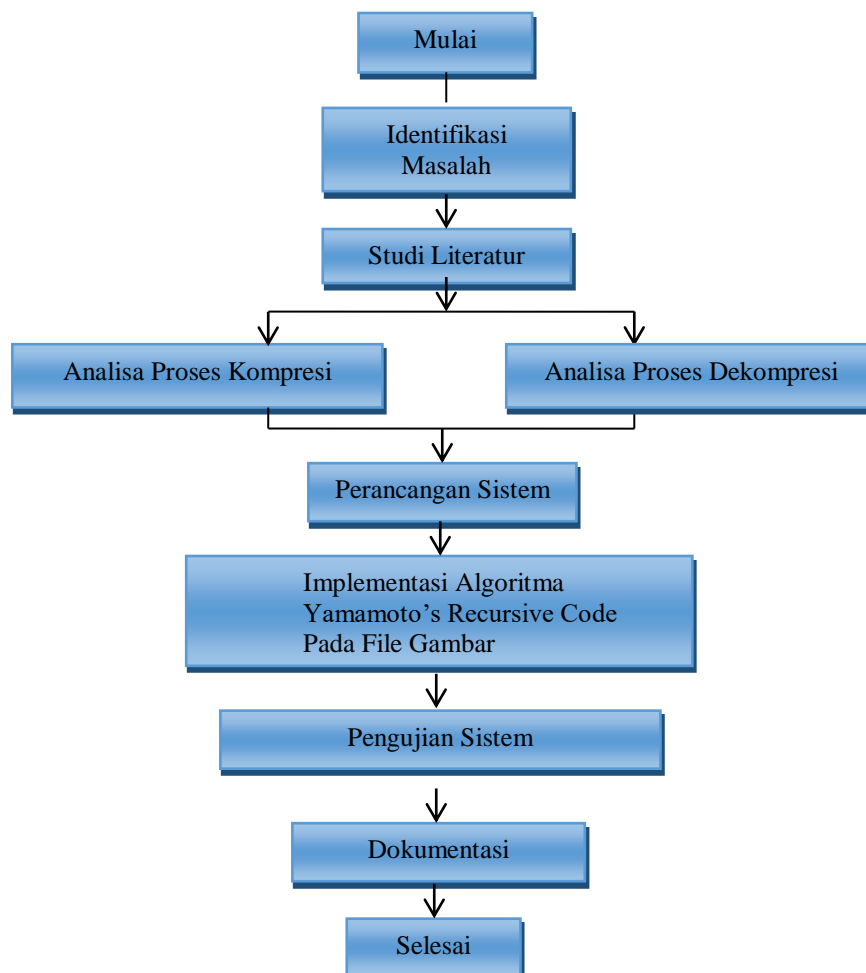
Pada tahun 2019 Andika Satyapratama, Widjianto dan Mahmud Yunus melakukan penelitian yaitu Analisis Perbandingan Algoritma LZW dan Huffman Pada Kompresi File Gambar BMP dan PNG bahwa Algoritma Huffman dapat mengompres *file* BMP dan PNG lebih cepat dibandingkan dengan algoritma LZW. Pada *file* gambar berformat BMP dan PNG, Algoritma LZW dapat menghasilkan rasio kompresi dengan persentase yang lebih kecil dibandingkan dengan algoritma Huffman, sehingga dapat menghasilkan *file* kompresi dengan *file* ukuran yang lebih kecil.[13]

Penelitian selanjutnya oleh Muhammad Khoiruddin Harahap pada tahun 2016 yang mengangkat judul Implementasi Algoritma Kompresi Shannon Fano pada Citra Digital bahwa dalam menggunakan Algoritma Shannon Fano dapat digunakan pada teks dan citra digital yang berguna untuk memampatkan ukuran *file* teks maupun citra digital agar menghemat memori penyimpanan [14][15].

Dalam penelitian yang akan dilakukan dicoba untuk mengompresi file gambar menggunakan algoritma Yamamoto's Recursive Code. Penggunaan algoritma Yamamoto's Recursive Codes diharapkan mampu untuk memperkecil ukuran *file* gambar. Hal ini belum pernah dicoba karena masih sedikit referensi penelitian mengenai algoritma Yamamoto's Recursive Code.

2. METODE PENELITIAN

2.1 Kerangka Kerja Penelitian



Gambar 1. Kerangka Kerja Penelitian

2.2 Kompresi File Gambar

Kompresi adalah mengubah data dalam bentuk kumpulan karakter ke dalam format yang dikodekan untuk menghemat kebutuhan penyimpanan dan waktu transmisi data [10][11][16]. Kompresi data sangat penting karena mengurangi kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan *bandwidth*. Terdapat banyak metode untuk mengompresi data. Metode-metode tersebut lahir dari ide yang berbeda-beda. Namun, prinsip dasar yang menjadi dasar tiap metode adalah sama, yaitu mengompresi data dengan menghilangkan *redundancy* dari data asli[17].

Pada suatu teknik yang digunakan dalam proses kompresi data terdapat beberapa faktor atau variabel yang biasa yang digunakan untuk menganalisa kuantitas dari suatu teknik kompresi data tersebut yaitu:

1. *Ratio of Compression(Rc)*
Ratio of Compression (Rc) adalah nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi
Rumus :

$$Rc = \frac{\text{Jumlah bit sebelum dikompresi}}{\text{Jumlah bit sesudah dikompresi}} \times 100\% \quad (1)$$
2. *Compression of Ratio (Cr)*
Compression of Ratio(Cr) adalah persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.
Rumus :

$$Cr = \frac{\text{Jumlah bit sudah dikompresi}}{\text{Jumlah bit sebelum dikompresi}} \times 100\% \quad (2)$$
3. *Space saving(Ss)*
Space saving(Ss) adalah selisih antara data yang belum dikompresi dengan besar data yang dikompresi
Rumus :

$$Ss = 100\% - Cr \quad (3)$$
4. *Time Compression*
Time Compression adalah waktu yang dibutuhkan untuk melakukan proses kompresi dan dekompresi. Semakin kecil waktu yang diperoleh maka semakin efisien metode yang digunakan dalam proses kompresi dan dekompresi itu.
Ada dua teknik yang dapat dilakukan dalam melakukan kompresi
 1. *Lossless Compression*
Lossless Compression merupakan teknik kompresi citra yang tidak menghilangkan informasi sebelumnya, dimana hasil dekompresi dari citra yang terkompresi sama dengan citra aslinya.
 2. *Lossy Compression*
Lossy Compression merupakan teknik kompresi citra yang akan menghilangkan beberapa informasi, dimana hasil dekompresi dari citra yang terkompresi tidak sama dengan citra aslinya, tetapi masih bisa ditolerir oleh persepsi mata.

2.3 Yamamoto's Recursive Code

Yamamoto's Recursive Code merupakan kode recursif untuk angka positif yang dimana setiap urutanyang diberikan dapat digunakan sebagai *delimiter*, berbeda dengan universal code lainnya yang menggunakan bit "0" sebagai *delimiter* seperti *Elias Omega Code*, *Even-Rodeh Code*, dll. *Delimiter* yang dipakai pada *Algoritma Yamamoto's Recursive Code* lebih pendek dari $\log_2 n$ dalam hampir dari semua angka bulat positif dibandingkan algoritma sebelumnya. Langkah-langkah untuk membangun *Yamamoto's Recursive Code* adalah sebagai berikut :

1. Tentukan *delimiter* f-bit (dimana f adalah integer positif), misal f = 2, dan *delimiter* 00.
2. Tentukan tabel $B_{a,f}(n)$. dengan mengurutkan bilangan biner yang tidak dimulai dengan *delimiter*, dalam hal ini kita contohkan 00.
3. Setelah didapatkan tabel $B_{a,f}(n)$. selanjutnya tentukan tabel $B_{a,f}(n)$ dengan mengeliminasi semua nilai $B_{a,f}(n)$ yang jika digabung membentuk *delimiter*, dalam hal ini kita dapat menentukan Code Yamamoto dengan melakukan perulangan dengan grup $B_{a,f}(n)$ hingga mencapai $B_{a,f}(1)$

3. HASIL DAN PEMBAHASAN

3.1 Analisa Penerapan Metode Algoritma Yamamoto's Recursive Code Pada Kompresi File Gambar

Analisa yang dilakukan dalam penelitian ini meliputi kompresi *file* gambar dengan menerapkan *file Algoritma Yamamoto's Recursive Code*. *File* gambar yang sangat viral, sehingga memerlukan pengamanan yang baik saat didistribusi ataupun disaat untuk mendeteksi keaslian *file* gambar. *File* gambar yang akan dikompres adalah *file* gambar yang mempunyai format bmp.

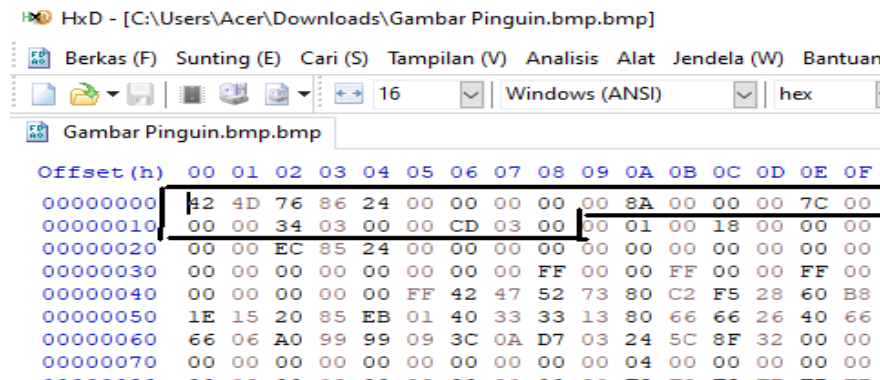
Algoritma *Yamamoto's Recursive Code* merupakan teknik kompresi lossless dan kode recursif untuk angka positif yang dimana setiap urutan yang diberikan dapat digunakan sebagai *delimiter*, berbeda dengan universal code lainnya yang menggunakan bit "0" sebagai *delimiter* seperti *Elias Omega Code*, *Even-Rodeh*, *Stout Code*, dll. *Delimiter* yang dipakai pada *Algoritma Yamamoto's Recursive Code* lebih pendek dari $\log_2 n$ dalam hampir semua angka bulat dibandingkan algoritma sebelumnya.

Tabel 1. Sampel Data

No.	Nama File	Ukuran
1	Gambar Harimau.bmp	149 KB
2	Gambar Gunung.bmp	1.464 KB
3	Gambar Mobil Tank.bmp	6.751 KB

4	Gambar Mario.bmp	148 KB
5	Gambar Pinguin.bmp	2.338 KB

Dari 5 (lima) sampel data, untuk proses analisa diambil hanya 1 sampel saja, kemudian dari sampel tersebut diambil 25 nilai heksa. Untuk mengambil 25 nilai heksa, digunakan software HxD. Untuk lebih jelasnya dapat dilihat pada gambar 2 berikut.

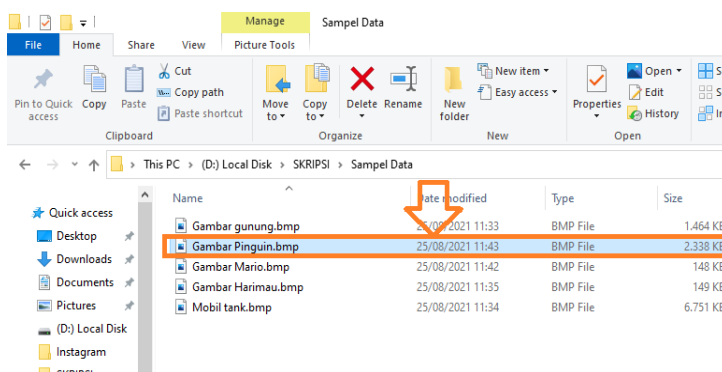


Gambar 2. Sampel Data Dari HxD

3.2 Implementasi dan Hasil Pengujian

Kompresi data memampatkan data besar menjadi ukuran kecil, mengurangi alokasi memori. Untuk menganalisis file gambar, file sampel harus dikembalikan untuk mendapatkan nilai datadalam file gambar sebagai nilai heksadesimal. Berikut adalah langkah-langkah untuk kompres dan dekomposisi file gambar. Analisa proses kompresi *file* gambar dengan menggunakan *Yamamoto's Recursive Code*.

a) Memasukkan File Gambar



Gambar 3. Import File Microsoft Excel

b) Melakukan Pembacaan File Gambar

Adapun bilangan hexadecimal dari *file* gambar tersebut adalah 42, 4D, 76, 86, 24, 00, 00, 00, 00, 00, 8A, 00, 00, 00, 7C, 00, 00, 00, 34, 03, 00, 00, CD, 03, 00. Nilai data ini dimasukkan kedalam tabel untuk dilakukan pembacaan frekuensi.

Tabel 2. Nilai File Gambar

Nilai	Frekuensi
42	1
4D	1
76	1
86	1
24	1
00	14
8A	1
7C	1
34	1
03	2
CD	1

Total Nilai	25
--------------------	-----------

c) Diurutkan dari karakter dengan frekuensi tertinggi

Tabel 3. Nilai Bit File Gambar

Nilai		Bit	Frek	Bit x Frek
Hexa	Biner			
00	00000000	8	14	112
42	01000010	8	1	8
4D	01001101	8	1	8
76	01110110	8	1	8
86	10000110	8	1	8
24	00100100	8	1	8
8A	10001010	8	1	8
7C	01111100	8	1	8
34	00110100	8	1	8
03	00000011	8	2	16
CD	11001101	8	1	8
Total Bit				200

d) Merubah Nilai heksa menjadi Codeword

Tabel 4. Heksa ke Codeword

No	Nilai Heksa	Frek	Codeword	Bit	Bit x Frek
1	00	14	100	3	42
2	42	1	101 00	5	5
3	4D	1	110 00	5	5
4	76	1	111 00	5	5
5	86	1	101 010 00	8	8
6	24	1	101 011 00	8	8
7	8A	1	101 100 00	8	8
8	7C	1	101 101 00	8	8
9	34	1	101 110 00	8	8
10	03	2	101 111 00	8	16
11	CD	1	110 0100 00	9	9
Total Bit		122			

e) Hasil Dekompresi

Tabel 5. Hasil Dekompresi

No.	Yamamoto's Recursive Code	Nilai Heksa
1	10100	42
2	11000	4D
3	11100	76
4	10101000	86
5	10101100	24
6	100	00
7	100	00
8	100	00
9	100	00
10	100	00
11	10110000	8A
12	100	00
13	100	00
14	100	00
15	100	7C
16	10110100	00
17	100	00
18	100	00
19	100	34
20	10111000	03
21	10111100	00

22	100	00
23	110010000	CD
24	10111100	03
25	100	00

Adapun perbedaan file gambar sebelum dikompresi dan sesudah dikompresi dapat dilihat dari hasil pengujian dibawah ini:

Tabel 6. Hasil Perbedaan Dekompresi

No	Sebelum Dikompresi		Sesudah Dikompresi		
	Nama File Gambar	Ukuran	Nama File Gambar Terkompresi	Ukuran	Kinerja (CR)
1	Gambar Harimau.bmp	149 KB	Gambar Harimau.yrc	74.5 KB	50%
2	Gambar Gunung.bmp	1.464 KB	Gambar Gunung.yrc	805 KB	55%
3	Gambar Mobil Tank.bmp	6.751 KB	Gambar Mobil Tank.yrc	4.338 KB	65%
4	Gambar Mario.bmp	148 KB	Gambar Mario.yrc	88 KB	60%
5	Gambar Pinguin.bmp	2.338 KB	Gambar Pinguin.yrc	1.589 KB	68%

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat ditarik beberapa kesimpulan dari pembahasan sebelumnya dari hasil akhir penelitian, Kesimpulannya ialah Prosedur kompresi *file* gambar menggunakan algoritma yamamoto's recursive code telah berhasil dilakukan dengan *file* gambar dalam format .bmp dan proses kompresi berjalan sesuai dengan teknik kompresi sedangkan *Algoritma yamamoto's recursive code* telah diterapkan dan telah ditunjukkan bahwa *file* gambar yang besar dapat dikompres ke ukuran yang lebih kecil. Aplikasi untuk mengompresi *file* gambar telah berhasil dirancang dan dibangun menggunakan aplikasi *Microsoft Visual Studio 2008* dengan menerapkan *Algoritma Yamamoto's Recursive Code*.

REFERENCES

- [1] S. B. Ginting, "Perbandingan Algoritma Yamamoto's Recursive Code Dan Additive Code Dalam Kompresi File Video," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 5, no. 1, 2021.
- [2] F. W. Pakpahan, "Perancangan Aplikasi Kompresi File Gambar Dengan Menerapkan Algoritma Unary Coding," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 1, no. 1, pp. 6–12, 2020.
- [3] S. Simanjuntak, "Implementasi Metode Taboo Code Untuk Kompresi File Video," *J. Sains dan Teknol. Inf.*, vol. 1, no. 3, pp. 79–84, 2022.
- [4] R. Manik, "Implementasi Algoritma Elias Gamma Code Untuk Kompresi File Audio Hasil Rekaman Dari Aplikasi Wising Karaoke," *Resolusi Rekayasa Tek. Inform. dan Inf.*, vol. 2, no. 6, pp. 265–273, 2022.
- [5] E. Siahaan, "Implementasi Algoritma Elias Gamma Code Untuk Kompresi File Teks Pada Aplikasi Watpad," *Resolusi Rekayasa Tek. Inform. dan Inf.*, vol. 2, no. 2, pp. 76–84, 2021.
- [6] J. Sisca, "Penerapan Algoritma Elias Delta Code Untuk Kompresi File Video Pada Aplikasi Video Downloader," *Resolusi Rekayasa Tek. Inform. dan Inf.*, vol. 1, no. 4, pp. 254–264, 2021.
- [7] K. Geofandy, E. A. Nathaniel, and H. Agung, "Kompresi File Menggunakan Konversi Biner Hexadecimal dan Algoritma Huffman Encoding," *J. Ilm. Teknol. Infomasi Terap.*, vol. 5, no. 3, pp. 36–46, 2019.
- [8] R. Handayani, "Perancangan Aplikasi Kompresi File Audio Menerapkan Algoritma Universal Codes," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 5, no. 1, 2021.
- [9] N. Aftikasyah, "Penerapan Algoritma Yamamoto's Recursive Code Untuk Mengkompresi File Dokumen," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 5, no. 1, 2021.
- [10] G.D.Sanjaya, R.Hadi, N.Luh, and G.Pivin, "Kompresi Citra Digital Menggunakan Metode Discrete Cosine Transform," *Sintak*, pp. 38–44, 2018.
- [11] P. Fitria, "Penerapan Algoritma Rice Codes Pada Aplikasi Kompresi File Gambar," vol. 1, no. 3, pp. 158–165, 2020.
- [12] E. N. Simanjuntak, "BEES : Bulletin of Electrical and Electronics Engineering Penerapan Algoritma Prefix Code Pada Kompresi File Gambar," vol. 1, no. 3, pp. 96–100, 2021.
- [13] and T. I. A. Satyapratama, M. Yunus, P. Studi, "Kompresi File Gambar Bmp Dan Png," *Ejurnal Stimata*, vol. 6, no. 69–81, p. 2, 2015.
- [14] M. K. Harahap, "Implementasi Algoritma Kompresi Shannon – Fano pada Citra Digital," pp. 252–258, 2017.
- [15] S. H. Silitonga and S. D. Nasution, "Implementasi Algoritma Boldi-Vigna Codes Untuk Kompresi File Audio Pada Aplikasi Pemutar Audio Berbasis Web," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 6, no. 1, pp. 586–595, 2023.
- [16] Y. Sihura, T. Zebua, and H. Hutabarat, "Kinerja Algoritma Yamamoto's Recursive Code dan Algoritma Fixed Length Binary Encoding pada Kompresi File PDF," *J. Comput. Syst. Informatics*, vol. 3, no. 4, pp. 303–312, 2022.
- [17] A. Fau and N. A. Hasibuan, "PLAGIARISM DETECTION IN RESEARCH OF LECTURERS AND STUDENTS AT BUDI DARMA UNIVERSITY APPLYING THE TURBO ALGORITHM BOYER MOORE," *INFOKUM*, vol. 10, no. 03, pp. 57–63, 2022.