JOURNAL OF COMPUTER SCIENCE AND INFORMATICS ENGINEERING
VOL. 05, NO. 1, JANUARY 2026: 28-42
Doi: https://doi.org/10.55537/cosie.v5i1.1355

e-ISSN 2827-8356
p-ISSN 2827-9999

Implementasi Kompresi Huffman untuk Optimasi Pengiriman Data Sensor pada Sistem Monitoring IoT

Implementation of Huffman Compression for Optimizing Sensor Data Delivery in IoT Monitoring Systems

M Khalil Gibran[*1], Mhd Ikhsan Rifki[2], Afandi Sahputra[3], Ahmad Taufik Al Afkari Siahaan[4]
[1,2,4]Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, [3]Sistem Kelistrikan Kapal, Politeknik Pelayaran Malahayati
E-mail: [1]m.khalil1100000202@uinsu.ac.id, [2]rifki.mhdikhsan@uinsu.ac.id, [3]afandi_sahputra@poltekpelaceh.ac.id, [4]ahmadtaufiqalafkary@uinsu.ac.id

Received: October 7, 2025 | Revised: October 17, 2025 | Accepted: November 11, 2025

Abstrak

Pengiriman data secara real-time pada lingkungan Internet of Things (IoT) sering menghadapi keterbatasan bandwidth dan efisiensi energi, yang dapat menimbulkan keterlambatan dalam proses transmisi data berkelanjutan. Penelitian ini melakukan simulasi pengiriman data berbasis ZigBee menggunakan Python untuk menganalisis efisiensi kompresi data sensor pada sistem IoT. Data yang dikirimkan berasal dari empat jenis sensor, yaitu sensor suhu, tekanan, kelembapan, dan getaran, dengan total 100 record data yang dibangkitkan secara acak berdasarkan kondisi ideal lingkungan industri. Metode Huffman Compression diterapkan untuk mengurangi ukuran data hasil pengukuran sebelum dikirimkan. Hasil simulasi menunjukkan bahwa total panjang data yang dikirimkan berhasil direduksi dari 27.632 bit menjadi 13.385 bit, atau mengalami penurunan sebesar 51,56%. Dampaknya, rata-rata waktu transmisi data berkurang dari 0,11053s menjadi 0,0535s, sementara konsumsi energi menurun dari 0,001382J menjadi 0,000669J. Selain itu, penggunaan bandwidth juga berkurang dari 27,63kb menjadi 13,38kb. Hasil ini menunjukkan bahwa penerapan Huffman Compression secara signifikan meningkatkan efisiensi transmisi data, konsumsi energi, dan penggunaan bandwidth pada sistem komunikasi IoT berbasis ZigBee.

Kata kunci: Kompresi Huffman, Optimasi Pengiriman Data, Jaringan Wireless Terbatas, Protokol ZigBee, Internet of Things (IoT)

Abstract

Real-time data transmission in Internet of Things (IoT) environments often faces limitations in bandwidth and energy efficiency, leading to latency during continuous data transfer. This study conducts a ZigBee-based simulation using Python to evaluate data compression efficiency in IoT sensor communication systems. The transmitted data are collected from four types of sensors temperature, pressure, humidity, and vibration with a total of 100 records generated randomly to represent ideal industrial environmental conditions. The Huffman Compression method is applied to reduce the size of sensor measurement data before transmission. Simulation results show that the total data length was reduced from 27,632 bits to 13,385 bits, achieving a 51.56% reduction. Consequently, the average transmission time decreased from 0.11053s to 0.0535s, and energy consumption dropped from 0.001382 J to 0.000669 J. In addition, bandwidth usage was reduced from 27.63kb to 13.38kb. These results indicate that applying Huffman Compression can significantly enhance transmission efficiency, reduce energy consumption, and optimize bandwidth utilization in ZigBee-based IoT communication systems.

e-ISSN 2827-8356
p-ISSN 2827-9999

Keywords: Huffman compression, Data transmission optimization, Limited wireless networks, ZigBee protocol, Internet of Things (IoT)

1. PENDAHULUAN

Kemajuan teknologi yang terus meningkat memberikan peran penting terhadap pemanfaatan jaringan sensor nirkabel dalam berbagai sistem monitoring. Sistem monitoring tersebut dapat diterapkan pada beragam lingkungan sesuai kebutuhan, seperti pertanian cerdas, pengelolaan energi, hingga integrasinya dengan Internet of Things (IoT). Namun, penggunaan jaringan komunikasi nirkabel pada ekosistem IoT seperti Zigbee, LoRa, dan teknologi sejenis memiliki keterbatasan dalam kapasitas bandwidth dan ketersediaan energi, yang secara langsung memengaruhi efisiensi transmisi data. Keterbatasan ini semakin menambah kompleksitas ketika volume data yang dikirim meningkat seiring bertambahnya waktu operasional sistem serta kebutuhan akan fitur instrumentasi tambahan [1]. Komunikasi nirkabel pada lingkungan IoT memiliki keterbatasan serius pada aspek kapasitas bandwidth dan efisiensi energi, yang secara langsung mempengaruhi kinerja transmisi data. Ketika jumlah data meningkat seiring bertambahnya waktu operasional dan jumlah sensor, maka beban transmisi jaringan meningkat secara linier, menyebabkan transmission delay, hilangnya sebagian data, serta sinkronisasi waktu antar-node yang terganggu [2]. Akibatnya, sistem otomatisasi yang bergantung pada kecepatan dan keakuratan data dapat gagal memberikan respon sesuai rules kerja aktuator[3].Untuk mengatasi permasalahan tersebut, berbagai pendekatan telah diusulkan dalam literatur, seperti penggunaan optimasi heuristik, pengkodean efisien, atau teknik kompresi data untuk menekan konsumsi energi dan menghemat bandwidth [4][5]. Teknik kompresi menjadi salah satu solusi yang paling menjanjikan karena dapat menurunkan ukuran data tanpa mengubah informasi aslinya (lossless). Salah satu algoritma yang banyak digunakan adalah Huffman Compression, yang bekerja berdasarkan frekuensi kemunculan simbol dalam data [6]. Simbol dengan frekuensi tinggi akan diberi kode yang lebih pendek, sementara simbol dengan frekuensi rendah diberi kode lebih panjang. Dengan cara ini, ukuran total data yang akan dikirimkan dapat berkurang secara signifikan [7].

Sejumlah penelitian terdahulu telah mengeksplorasi penerapan Huffman Compression dalam berbagai konteks. Misalnya, [8] mengembangkan Ultra Energy Efficient Lossless Compression (UEELC) untuk menghemat energi pada Wireless Body Area Networks (WBANs), [9] mengombinasikan Huffman dan LZW untuk kompresi data medis secara hibrida, dan [10] mengintegrasikan Huffman ke dalam model Adaptive Lossless Data Compression (ALDC) untuk meningkatkan efisiensi energi pada jaringan sensor. Selain itu, [11] mengusulkan Huffman Deep Compression (HDC) berbasis deep learning untuk pengelompokan pola data IoT, sementara [12] merancang arsitektur hardware berbasis Canonical Huffman yang diimplementasikan menggunakan FPGA. Meskipun berbagai pendekatan tersebut menunjukkan hasil yang menjanjikan, mayoritas penelitian masih berfokus pada domain tertentu (misalnya kesehatan, body network, atau sistem hardware khusus), belum secara spesifik menganalisis efisiensi kompresi data sensor multivariabel pada sistem IoT industri dengan protokol ZigBee yang memiliki bandwidth terbatas. Selain itu, sebagian besar penelitian menggunakan data eksperimental statis tanpa menilai pengaruh langsung terhadap waktu transmisi, konsumsi energi, dan penggunaan bandwidth secara simultan dalam konteks simulasi real-time. Berdasarkan kesenjangan tersebut, penelitian ini berfokus pada simulasi penerapan Huffman Compression dalam sistem monitoring berbasis IoT menggunakan protokol ZigBee. Data yang dikirimkan berasal dari empat jenis sensor suhu, tekanan, kelembapan, dan getaran yang dibangkitkan secara acak untuk merepresentasikan kondisi ideal lingkungan industri. Proses simulasi dilakukan secara komputasional berbasis Python, dengan pemodelan alur lengkap yang meliputi: proses encoding data sensor, transmisi melalui jaringan ZigBee yang disimulasikan, serta decoding dan rekonstruksi data di sisi penerima. Kontribusi utama penelitian ini adalah Merancang dan

e-ISSN 2827-8356
p-ISSN 2827-9999

mengimplementasikan model simulasi Huffman Compression berbasis Python untuk lingkungan IoT dengan protokol ZigBee, Menganalisis secara kuantitatif pengaruh kompresi terhadap ukuran data, waktu transmisi, konsumsi energi, dan bandwidth. Memberikan pendekatan kompresi ringan dan efisien yang sesuai untuk perangkat IoT berdaya rendah, sehingga dapat menjadi referensi dalam pengembangan sistem monitoring industri yang hemat energi dan bandwidth. Dengan demikian, penelitian ini tidak hanya memperkuat bukti empiris efektivitas Huffman Compression dalam sistem IoT, tetapi juga menawarkan model simulatif yang realistis untuk mengevaluasi efisiensi transmisi pada jaringan ZigBee.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

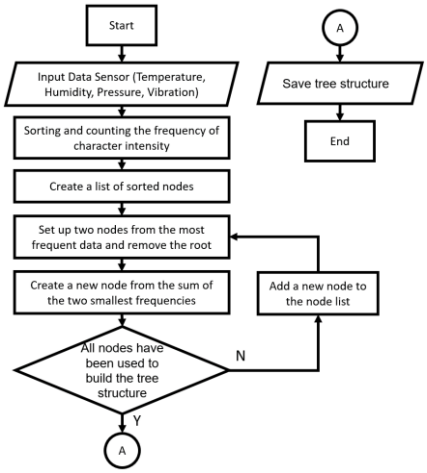
Huffman Compression termasuk ke dalam salah satu kategori pengkodean entropi yang diimplementasikan untuk kebutuhan kompresi data dengan tidak menghilangkan informasi asli data yang dikompresi. *Huffman Compression* bekerja dengan menggunakan prinsip kompresi berlandasakan komputasiu jumlah frekuensi data yang sering muncul dalam serangkaian data, sehingga informasi yang telah dikodekan dan dikompresi dapat disebut dengan istilah kode *Huffman* [13]. Dalam proses pengkodean data, *Huffman compression* dapat mengkompresi panjang bit atau ukuran data menjadi ukuran yang lebih kecil, misalnya jika data merujuk pada jenis teks dengan “TIM” dengan ukuran asli 24 bit. Maka dengan menggunakan *Huffman*, ukuran data dapat dipersingkat dengan hanya memiliki representasi 12 bit, yang artinya *Huffman* memberikan penghematan penggunaan sebanyak 12 bit. Tahapan pemrosesan pengkodean data dilakukan dalam beberapa tahapan yang dapat diuraikan sebagai berikut :

1. Data input merupakan serangkaian data yang dihimpun dari dari hasil pengukuran pada lingkungan sistem monitoring berbasis IoT[14].
2. Data hasil pengukuran sensor akan di-*scanning* akan dihitung jumlah frekuensi kemunculannya, dan dilakukan pengurutan karakter, dimulai dari frekuensi yang paling tinggi menunju frekuensi kemunculan data yang paling rendah. Daftar frekuensi yang telah dibuat dijadikamn sebagai dasar, untuk membentuk struktur pohon *Huffman*.
3. Data yang telah memiliki pasangan simpul, akan dipecah berdasaerkan simpul utama untuk dijadikan akar-akar sesuai dengan nilai pembobotan karakter [15].
4. Pemisahan struktur akar dilakukan secara kontinyu hingga diperoleh satu karakter terkahir, dengan cabang yang memiliki kecenderungan posisi pada bagian kiri, dan diberikan simbol “0”. Sementara cabang yang memiliki kecenderungan posisi ke kana akan diinisialisasi dengan tanda “1”

Berdasarkan pemecahan pohon tersebut, kemudian dibuat daftar dengan mengambil tanda yang telah diinisialisasi sebelumnya untuk membentuk kode baru yang dibentuk melalui susunan proses struktur pohon *Huffman* yang mengahasilkan data baru yang telah dienkripsi dalam bentuk bilangan biner.

Ilutrasi flowchart pengkodean dan kompresi Huffman dapat diuraikan pada Gambar 1.

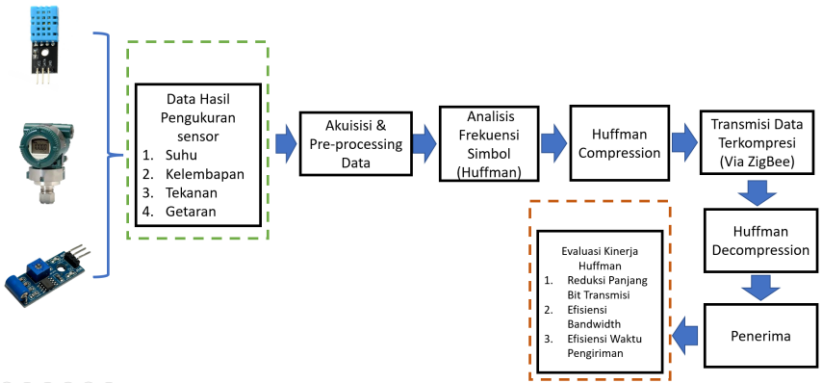
e-ISSN 2827-8356
p-ISSN 2827-9999



Gambar 1. Flowchart Huffman Coding

2.2 Diagram Blok Penelitian

Penelitian yang dilakukan terdiri atas beberapa tahapan sistematis yang dirancang untuk menerapkan metode kompresi Huffmnan pada proses transmisi data sensor pada lingkungan IoT [16]. Tujuan berorientasi pada reduksi panjang bit yang akan ditransmisikan sebagai upaya dalam penghematan waktu transmisi dan, *bandwidth*. Ilustrasi diagram blok penelitian dapat dijabarkan pada Gambar 2.



Gambar 2. Diagram Blok Penelitian

Berdasarkan Gambar 2, diperoleh informasi tahapan penelitian yang secara detail dapat dideskripsikan sebagai berikut :

1. Data Input

   Data yang diinisialisasikan pada metode komprsi Huffman merupakan data yang dibangkitkan secara *random* melalui sensor suhu dan kelembapan, tekanan, dan getaran. Data hasil pengukuran merupakan jenis data dalam format analog dan digital. Jumlah data sensor yang digunakan memiliki jumlah 100 *record* data.

2. Akusisi dan *Pre-processing* Data

   Data yang diterima dalam bentuk analog dan digital akan dikonversi ke dalam data baru dalam bentuk *string*. Proses akuisisi data selanjutnya akan dilanjutkan ke tahapan normalisasi format data dan pembersihan data dari *noise, outlier,* serta berbagai penyimpangan data lainnya [17].

e-ISSN 2827-8356
p-ISSN 2827-9999

3. Komputasi Frekuensi Simbol
   Setelah data dinormalisasi, langkah berikutnya adalah menganalisis frekuensi kemunculan simbol (karakter) dalam setiap *string* data sensor. Frekuensi ini digunakan sebagai dasar pembentukan pohon Huffman. Misalnya, karakter yang paling sering muncul akan diberikan kode biner yang lebih pendek, sementara karakter dengan frekuensi lebih rendah diberikan kode yang lebih panjang. Analisis frekuensi ini dihitung menggunakan persamaan berikut:

$$f(c) = \frac{nc}{n}$$

   Keterangan :
   f(c) : Frekunsi karakter *c*
   nc : Jumlah munculnya karakter *c*
   n : Akumulasi total jumlah karakter *c*
4. Pembentukan Pohon Huffman dan Proses Kompresi
   Berdasarkan distribusi frekuensi karakter, pohon Huffman dibentuk menggunakan pendekatan *bottom-up*. Simbol-simbol yang paling jarang muncul ditempatkan di bagian bawah pohon, dan setiap simpul internal menyimpan jumlah frekuensi dari cabangnya. Proses ini menghasilkan tabel Huffman, yaitu pemetaan antara setiap karakter dengan representasi biner unik. Data sensor yang semula berbentuk string kemudian dikompresi menjadi rangkaian bitstream menggunakan tabel Huffman tersebut.
5. Transmisi Data Terkompresi
   Data yang telah dikompresi selanjutnya dikirimkan melalui media transmisi nirkabel menggunakan modul komunikasi *ZigBee*. Implementasi proses kompresi secara signifikan mengurangi jumlah bit yang dikirim, sehingga berdampak pada efisiensi penggunaan bandwidth dan waktu pengiriman.
6. Dekompresi Data di Sisi Penerima
   Pada sisi penerima, data bitstream yang diterima didekompresi menggunakan pohon Huffman yang sama dengan yang digunakan di sisi pengirim. Proses ini menghasilkan kembali data sensor dalam bentuk string aslinya tanpa kehilangan informasi (lossless). Validasi dilakukan untuk memastikan integritas data hasil dekompresi identik dengan data sebelum dikompresi [18].
7. Evaluasi Kinerja Kompresi Huffman
   Evaluasi kinerja dilakukan dengan tiga metrik utama, yaitu:
   a. Efisiensi Reduksi Panjang Bit
      Persentase efisiensi panjang bit merepresentasikan reduksi bit yang dihasilkan melalui metode *Huffman compression*. Komputasi persentase *Huffman* dapatb dilakukan dengan menggunakan Persamaan berikut:

$$Eff_{bit} = \left(1 - \frac{Pbit\ Terkompresi}{Pbit\ asli}\right) x\ 100\%$$

      Keterangan :
      Eff_{bit} : Efisiensi bit
      $Pbit\ Terkompresi$: Hasil proses kompresi bit
      $Pbit\ asli$: Ukuran bit asli

e-ISSN 2827-8356
p-ISSN 2827-9999

b. Efisiensi *Bandwidth*

Efisiensi *bandwith* merepresentasikan reduksi penggunaan *bandwitdh* setelah data dikompresi dengan menggunakan *Huffman Compression*. Perbandingan dilakukan anatara konsumsi *bandwidth* pada data asli yang ditransmisikan dengan besaran *bandwidth* yang dibutuhkan saat data terkompresi dikirimkan.Kompoutasi persentase perbandingan efisiensi *bandwidth* dapat dihitung dengan menggunakan Persamaan berikut:

$$Eff_{Bandwidth} = \left(1 - \frac{Ukuran\ data\ Terkompresi}{Ukuran\ data\ asli}\right) x\ 100\%$$

Keterangan :

$Eff_{Bandwidth}$: Efisiensi bandwidth
$Ukuran\ data\ Terkompresi$: Hasil proses kompresi bit
$Ukuran\ data\ asli$: Ukuran bit asli

c. Efisiensi Waktu Pengiriman:

Waktu pengiriman data didefinisikan sebagai level durasi perjalanan data yang ditransmisikan dari pengirim ke penerima. Efisiensi waktu pengiriman mendeskripsikan persentase perbandingan penghematan waktu antara waktu pengiriman data asli dibandingkan dengan waktu pengiriman data terkompresi yang ukurannya menjadi lebih kecil. Persentase penghematan waktu pengiriman dapat dikomputasi melalui formulasi Persamaan berikut:

$$Eff_{waktu} = \left(1 - \frac{Waktu\ kirim\ data\ Terkompresi}{waktu\ kirim\ data\ asli}\right) x\ 100\%$$

Keterangan :

Eff_{waktu} : Efisiensi $waktu$
$waktu\ kirim\ data\ Terkompresi$: waktu yang di butuhkan untuk mengirim data hasil kompresi
$waktu\ kirim\ data\ asli$: waktu yang di butuhkan untuk mengirim asli

3. HASIL DAN PEMBAHASAN

Data sensor pada sistem IoT akan dibangkitkan melalui proses pengukuran data suhu, kelembapan, tekanan dan getaran. Data yang dibangkitkan melalui keempat sensor akan di-*record* sebanyak 100 kali, sehingga total keseluruhan data yang dihimpun selama proses pengukuran berjumlah 400 data secara kumulatif. Sampel data yang dihimpun dapat diuraikamn melalui Tabel 1.

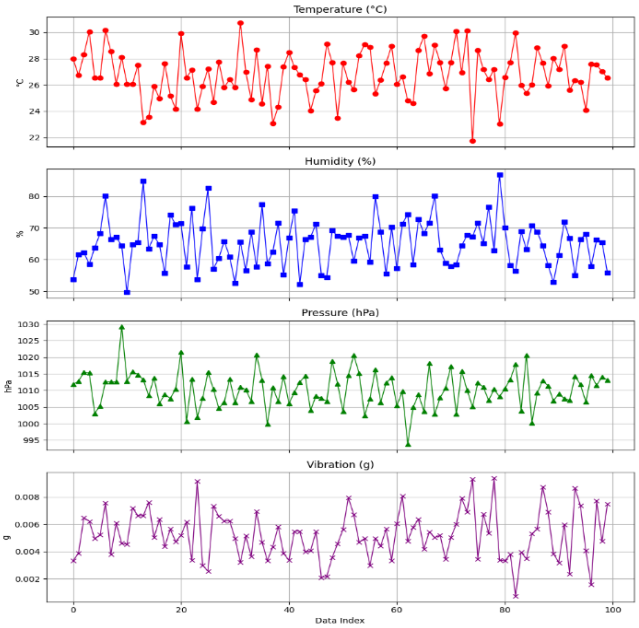
Tabel 1. Data Hasil Pengukuran Sensor

***Record* ke**	**Temperature (°C)**	**Humidity (%)**	**Pressure (hPa)**	**Vibration (g)**
0	27.99	53.68	1011.79	0.00334
1	26.72	61.63	1012.80	0.00388
2	28.30	62.26	1015.42	0.00649
…	…	….	…..	….
97	27.52	66.23	1011.54	0.00771

e-ISSN 2827-8356
p-ISSN 2827-9999

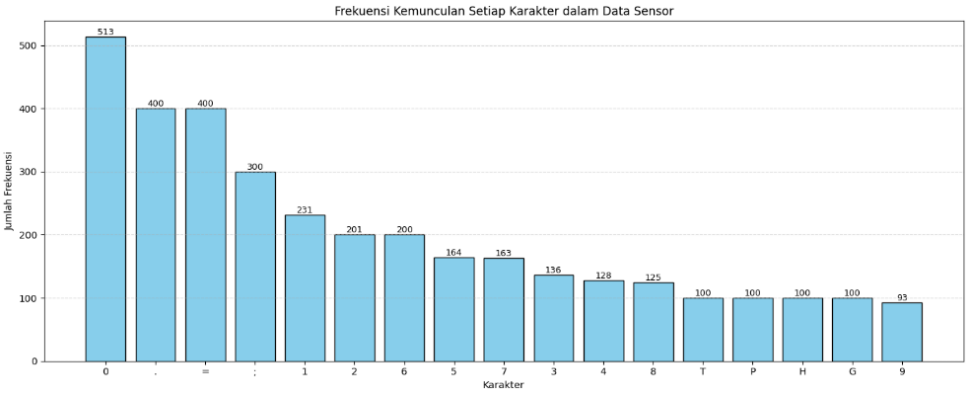
98	27.01	65.47	1014.06	0.00477
99	26.53	55.86	1013.15	0.00748

Berdasarkan uraian data pada Tabel 1, data yang dihimpun akan dijadikan sebagai output pada tahapan akuisisi dan *pre-processing*. Tahapan tersebut akan mengkomputasi data untuk keperluan normalisasi dan monitoring *outlier* serta *noise* yang dapat menyebabkan penyimpangan data yang terlalu signifikan. Visualisasi data untuk 100 *record* data yang dihimpun dapat diilustrasikan melalui Gambar 3.



Gambar 2. Visualisasi Data Hasil Pengukuran Sensor

Proses dilanjiutkan dengan mengimplementasikan fungsi komputasi, untuk menghitung jumlah intensitas frekuensi karakter alphabet, numerik dan simbol. Hasil komputasi yang diperoleh akan difungsi akan menghitung jumlah karakter pada hasil gabungan data yang hasil pengukuran sensor yang akan dikirimkan. Kode-kode pada data sensor akan ditransformasikan dalam bentuk format *string*, dan akan digabungkan secara kumulatif. Sehingga hasil perhitungan jumlah setiap karakter pada data lima sampel pertama dapat lihat pada grafik Gambar 3 dan Tabel 2.



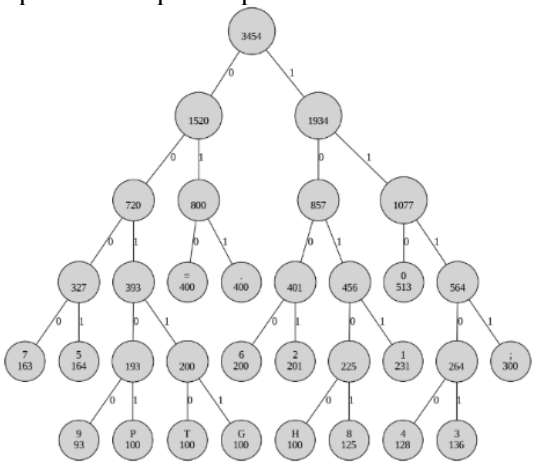
Gambar 3. Grafik Frekuensi Jumlah Karakter

Tabel 2. Frekuensi Jumlah Karakter

e-ISSN 2827-8356
p-ISSN 2827-9999

No	Karakter	Frekuensi	No	Karakter	Frekuensi
1	0	513	**10**	3	136
2	.	400	**11**	4	128
3	=	400	**12**	8	125
4	;	300	**13**	T	100
5	1	231	**14**	P	100
6	2	201	**15**	H	100
7	6	200	**16**	G	100
8	5	164	**17**	9	93
9	7	163			

Pada grafik Gambar 3, dan Tabel 2 dapat dideskripsikan urutan karakter yang paling sering muncul pada data hasil pengukuran yang telah digabungkan secara kumulatif. Kemudaian karakter pada data disusun mulai dari urutan yang paling tinggi frekuensinya ke frekuensi data yang paling rendah [19]. Tahapan selanjutkan dilanjutkan dengan melakukan proses konversi karakter data *string* dengan dengan mengimplementasikan struktur pohon *Huffman*. Kemudian dilanjutkan dengan proses pembentukan kode *Huffman* untuk proses pengkodean dan Berdasarkan hasil pemetaan karakter dengan menggunakan frekuensi kemunculan karakter, struktur pohon *Huffman* dapat dideskripsikan pada Gambar 5.



Gambar 4. Struktur Pohon Huffman

Pembentukan pohon *Huffman* digunakan untuk menghasilkan representasi data biner berdasarkan frekuensi jumlah karakter yang muncul pada data [20]. Pohon *Huffman* memiliki akar pohon yang diposisikan dengan *node* yang paling tinggi yang bernilai 3454. Nilai ini mendeskripsikan total karakter yang terdapat dalam susunan data hasil instrumentasi keempat sensor. Sementara *node* daun merepresentasikan karakter dan frekuensi, pada Gambat dapat dilihat karakater “G”, “T”, “H”, “P” yang masing-masing memiliki nilai 100 sesuai dengan frekuensi kemunculan karakter-karakter tersebut dalam didalam data. Pada pohon *Huffman* jalur yang digunakan dalam merepresentasikan kode biner karakter “G”, didasari oleh jalur akar menuju daun karakter “G”.

e-ISSN 2827-8356
p-ISSN 2827-9999

Sehingga kode biner hasil perubahan karakter "G" melewati biner 0,0,1,1,1 menuju G (100), jalur ini membentuk serangkaian biner 00111 yang digunakan dalam merubah karakter "G". Secara detail konversi kode *Huffman* untuk karakter G dapat dideskripsikan pada Tabel 3.

Tabel 3. Tabel Konversi Kode Huffman

Karakter	Frekuensi	Kode Huffman	Karakter	Frekuensi	Kode Huffman
0	513	110	7	163	0000
=	400	010	3	136	11101
.	400	011	4	128	11100
;	300	1111	8	125	10101
1	231	1011	P	100	00101
2	201	1001	T	100	00110
6	200	1000	G	100	00111
5	164	0001	H	100	10100
			9	93	00100

Proses konversi tabel *huffman* memiliki perbedaan yang signifikan antar karakter yang ditransformasikan. Regulasi transformasi berorientasi pada penggunakan kode biner paling singkat untuk karakter yang paling sering muncul, dan sebaliknya kode biner paling panjang akan diterapkan pada karakter yang paling panjang. Hasil proses konversi untuk setiap karakter pada sampel kelima data pertama dapat diilustrasikan pada Tabel 4.

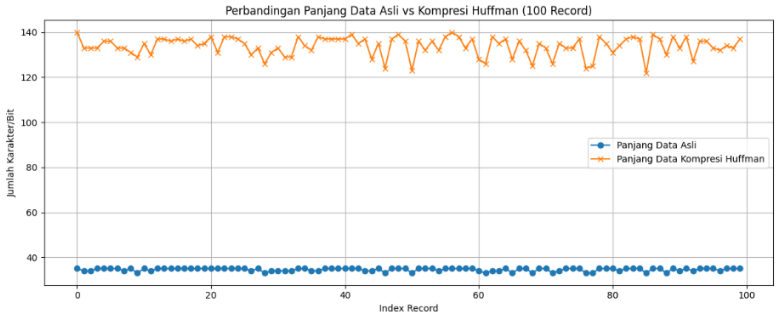
Tabel 4. Hasil Konversi Data menggunakan Kode Huffman

	***Record* Data ke-0**
Data Asli	T=27.99;H=53.68;P=1011.79;G=0.00334
Kode Huffman	0011001010010000011001000010011111010001000011110101110001010111110010101010111101011101101100000010011110011101011001111011011101111011100
	***Record* Data ke-1**
Data Asli	T=26.72 H=61.63 P=1012.8 G=0.00388
Kode Huffman	0001100111001001010011101100111110000001100111100101001011011110001001111010111011000100000111111111101000110101010110101100000000
	***Record* Data ke-2**
Data Asli	T=26.53 H=63.71 P=1003.11 G=0.00496
Kode Huffman	0001100111001001010110110110111110000001100101100100111011101111000100111101011010110010111011101111101000110101010110100010011111001
	…
	***Record* Data ke-97**
Data Asli	T=27.52;H=66.23;P=1011.54;G=0.00771
Kode Huffman	0011001010010000011000110011111101000101000100001110011110111110010101010111101011101101100011110011110011101011001111011000000000001011
	***Record* Data ke-98**
Data Asli	T=27.01;H=65.47;P=1014.06;G=0.00477
Kode Huffman	0011001010010000011110101111111010001010000001011111000000011110010101010111101011111000111101000111100111010110011111011011100000000000

e-ISSN 2827-8356
p-ISSN 2827-9999

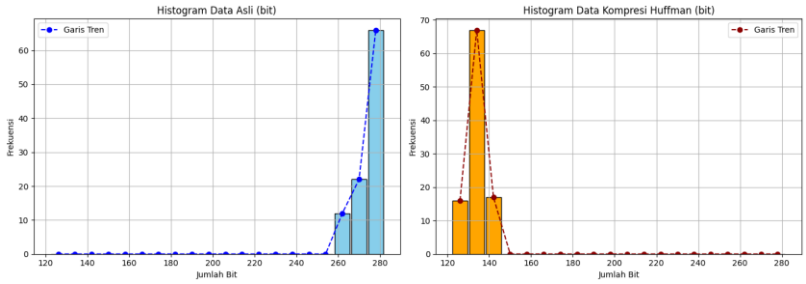
***Record* Data ke-99**	
Data Asli	T=26.53;H=55.86;P=1013.15;G=0.00748
Kode Huffman	00110010100110000110001111011111101000100001000101110101100011110010101 01011110101111101011101100011111001110101100111101100000111001010 1

Berdasarkan uraian Tabel 3. secara visualisasi, data asli memiliki ukuran panjang data yang lebih singkat dibandingan dengan data hasil pengkodean *huffman*. Namun dalam prinsip pengiriman data dengan menggunakan jaringan komunikasi wireless, format data asli akan dikonversi dengan menggunakan pengkodean tertentu sesuai dengan konfigurasinya. Sehingga setiap karakter pada data asli akan ditransformasikan, termasuk penggunaan karakter "*spasi*", "=", "." serta berbagai karakter tanda baca lainnya. Transformasi format data dapat dilakukan dengan menggunakan kode ASCII, maka jika diasumsikan satu karakter data asli diwakili dengan 8 bit ASCII, dengan total 35 karakter, maka panjang bit pada hasil pengkodean akan memiliki kuantitas bit sebanyak 280 bit. Sementara pada pengkodean *huffman*, panjang total bit terkode dan terkompresi memiliki kuantitas bit sebanyak 147 bit. Sehingga hasil pengkodean dan kompresi menggunakan *huffman compression* memiliki panjang bit yang lebih singkat dibandingkan data asli yang dikonversi sesaat sebelum proses pengiriman dilakukan. Ilustrasi grafik perbandingan seluruh panjang data asli dan data hasil kompresi *Huffman* secara kumulatif dapat direpresentasikan pada grafik Gambar 6.



Gambar 5. Perbandingan Hasil Panjang Data Asli dan Data Huffman Compression

Visualisasi lainnya, dapat diilsutrasikan dengan menggunakan skema histogram data untuk menentukan perilaku distribusi data antara data asli dan data hasil pengkodean dan kompresi *Huffman Compression*. Ilutrasi histogram perbandingan dapat divisualisasikan melalui Gambar 7.

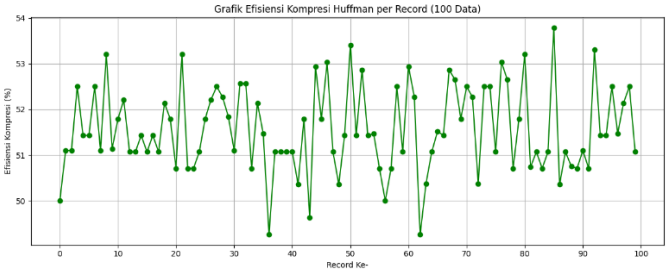


Gambar 6. Visualisasi Histogram Panjang Data Asli dan Data Kompresi Menggunakan Huffman Compression

Berdasarkan grafik histogram Gambar 5, dapat dideskripsikan bahwa distribusi histigram data asli berada pada interval panjang 260 bit hingga 280 bit, sementara pada histogram data hasil kompresi Huffman distribusi sebaran data berada pada interval panjang 130 hingga 140 bit. Berdasarkan interval panjang bit tertinggi kedua histogram, huffman compression mampu

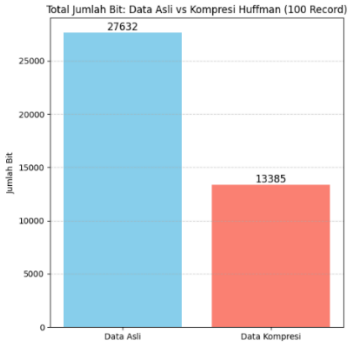
e-ISSN 2827-8356
p-ISSN 2827-9999

mereduksi panjang bit sebesar sekitar 50%. Sehingga dampaknya dapat menghemat energi dan konsumsi bandwidth yang dibutuhkan dalam mentransmisikan data dengan jumlah bit yang lebih singkat. Sementara itu, visualisasi hasil perbandingan rasio kompresi juga dapat dideskripsikan melalui rasio efisiensi kompresi huffman setiap baris data terkompresi sebanyak 100 record data dapat diuraikan pada Gambar 6.



Gambar 7. Grafik Kompresi Huffman per Record Data (%)

Berdasarkan visualisasi Gambar 8. interval nilai efisiensi ratio kompresi *huffman* berkisar antara 49,26% - 53, 79 %, dengan rerata persentase ratio berada pada level 51.56%. Sehingga berdasarkan interval ratio tersebut, kinerja *huffman compression* secara rata-rata mampu mereduksi pengukuran bit menjadi sekitar setengah (50%) lebih singkat dibandingkan dengan data asli. Analisis persentase reduksi juga dapat dilakukan dengan mengidentifikasi jumlah bit data asli dan data terkompresi dengan menggunakan *huffman compression*, sebagaimana yang dideskripsikan pada Gambar 9.



Gambar 8. Perbandingan Jumlah bit Secara Kumulatif

Berdasarkan visualisasi Gambar 7, dapat dideskripsikan, total panjang data yang dikirimkan memiliki ukuran 27632 bit menjadi 13385 bit, dengan persentase reduksi panjang berada pada level 51.56%. Ilutrasi panjang bit dan informasi statistik kinerja Huffman Compression untuk sampel lima baris data pertama dapat dijabarkan pada Tabel 4 dan Tabel 5.

Tabel 5. Sampel Panjang Bit Data Sensor Sebelum dan Sesudah Kompresi Huffman

Record	Data Asli (Bit)	Data Terkompresi (Bit)	Rasio Kompresi	Efisiensi (%)
0	280	140	2.00	50.00
1	272	133	2.05	51.10
2	272	133	2.05	51.10
3	280	133	2.11	52.50

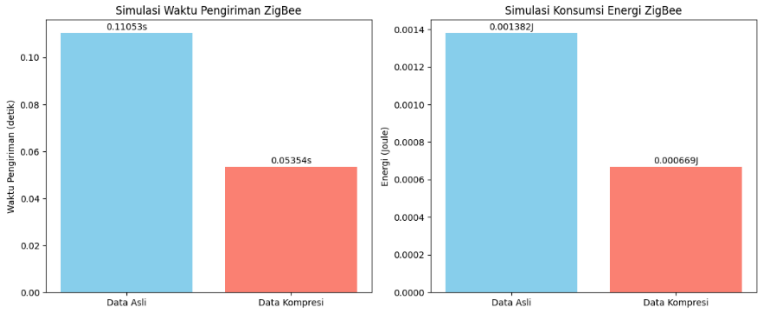
e-ISSN 2827-8356
p-ISSN 2827-9999

4	280	136	2.06	51.43
...	...	...	...	...
95	280	133	2.11	52.50
96	272	132	2.06	51.47
97	280	134	2.09	52.14
98	280	133	2.11	52.50
99	280	137	2.04	51.70

Tabel 6. Statistik Hasil Komputasi Rasio Kompresi Huffman

Statistik Rasio Kompresi Huffman		
No	**Statistik**	**Rasio**
1	Rata-rata Rasio Kompresi	2.07
2	Efisiensi Kompresi Maksimum	2.16
3	Efisiensi Kompresi Minimum	1.97
4	Rentang Rasio Kompresi	0.19

Setelah melakukan analisis performa *Huffman Compression* melalui aspek reduksi bit, maka selanjutnya dilakukan pengujian dan analisis reduksi waktu transmisi data. Evaluasi ini bertujuan untuk menganalisis rerata persentase pengurangan waktu transmisi dan konsumsi energi pada data hasil kompresi. Visualisasi grafik waktu transmisi dan konsumsi energi disajikan pada Gambar 10.

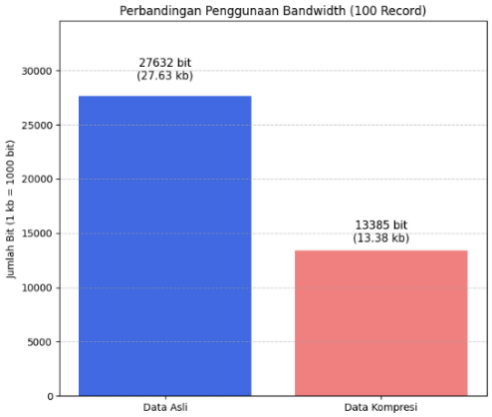


Gambar 9. Visualisasi Grafik Waktu Transmisi Dan Konsumsi Energi

Berdasarkan visualisasi grafik pada Gambar 8, rerata waktu pengiriman transmisi data tanpa dikompresi memiliki durasi 0,1105 s untuk data asli, sementara pengiriman data yang telah direduksi menjadi 0,0535 s. Sementara itu, Konsumsi energi rata-rata data asli berada pada level 0,001382J sedangkan data yang telah dikompresi dengan menggunakan *Huffman Compression*, memiliki konsumsi energi rata-rata sebesar 0,000669J. Kedua aspek waktu transmisi data dan konsumsi mengalami rerata penurunan sebesar 51,56%. Sehingga peranan *Huffman Compression* mampu mereduksi panjang data asli yang berdampak langsung pada waktu pengiriman bandwith dalam jaringan komunikasi ZigBee.

Sementara itu, analisis juga dilakukan pada penggunaan *bandwidth* saat data ditransmisikan dari pengirim menuju penerima melalui jaringan komunikasi. Visualisasi hasil perbandingan penggunaan *bandwidth* dapat disajikan pada Gambar 11.

e-ISSN 2827-8356
p-ISSN 2827-9999



Gambar 10. Simulasi Bandwidth via Zigbee (100 Record)

Berdasarkan visualisai Gambar 11, aspek penggunaan *bandwidth* pada data asli yang semula 27,32 kb, menjadi 13,38 kb dengan rerata persentase efisiensi mencapai level 57,11%. Hal ini mengindikasikan bahwa penerapan *huffman compression* dapat mengurangi waktu saat proses pengiriman data pada sistem IoT. Reduksi penggunaan ini sangat membantu dalam komunikasi pada jaringan *ZigBee* pada sistem IoT yang memiliki identitas daya rendah yang hanya berkisar sekitar 250 kbps. Performa *Huffman compression* juga dievaluasi dengan pengujian *decode* dan *decompression* pada sis penerima. Visualisasi hasil pengujian dapat diilustrasikan pada Gambar 12.

	Record	Decoded
0	T=27.99;H=53.68;P=1011.79;G=0.00334	T=27.99;H=53.68;P=1011.79;G=0.00334
1	T=26.72;H=61.63;P=1012.8;G=0.00388	T=26.72;H=61.63;P=1012.8;G=0.00388
2	T=28.3;H=62.26;P=1015.42;G=0.00649	T=28.3;H=62.26;P=1015.42;G=0.00649
3	T=30.05;H=58.58;P=1015.27;G=0.00622	T=30.05;H=58.58;P=1015.27;G=0.00622
4	T=26.53;H=63.71;P=1003.11;G=0.00496	T=26.53;H=63.71;P=1003.11;G=0.00496
...	...	...
95	T=24.07;H=68.08;P=1006.54;G=0.00406	T=24.07;H=68.08;P=1006.54;G=0.00406
96	T=27.59;H=57.93;P=1014.5;G=0.00157	T=27.59;H=57.93;P=1014.5;G=0.00157
97	T=27.52;H=66.23;P=1011.54;G=0.00771	T=27.52;H=66.23;P=1011.54;G=0.00771
98	T=27.01;H=65.47;P=1014.06;G=0.00477	T=27.01;H=65.47;P=1014.06;G=0.00477
99	T=26.53;H=55.86;P=1013.15;G=0.00748	T=26.53;H=55.86;P=1013.15;G=0.00748

100 rows × 2 columns

Gambar 11. Visualisasi Hasil Pengujian Decode dan Decompression

Data yang telah dikodekan dan dikompresikan di-*reverse* melalui fungsi *decode* dan *decompression* pada sisi penerima. Sehingga data akan dikembalikan pada format data hasil pengukuran semula. Hasil *decode* dan *decompression Huffman* pada Gambar 10. Menunjukkan bahwa tidak ada perbedaan antara data yang dikirimkan dengan data yang diterima.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, implementasi metode *huffman compression* memberikan dampak yang signifikan pada dalam mengoptimalisasi pengiriman data pada lingkungan sistem IoT dengan menggunakan ZigBee. Penggunaan sensor pada sistem IoT yang berjumlah empat sensor dapat dikodekan dengan menggunakan transformasi karakater menjadi

e-ISSN 2827-8356
p-ISSN 2827-9999

kode biner, serta dapat direduksi ukurannya melalui proses *huffman compression*. Hasil visualisasi *huffman compression* mampu memberikan penghematan ukuran panjang bit data sebesar 51.56%. Sehingga hal ini berpengaruh pada penghematan waktu transmisi dan penggunaan *bandwitdh* selama proses pengiriman data. Besaran level reduksi waktu transmisi mencapai rasio persentase penghematan sebesar 40,90%, sedangkan pada penggunaan *bandwidth* mencapai rasio efisiensi sebesar 57,11%.

Secara keseluruhan proses komputasi yang meliputi pengkodean, kompresi ukuran data, proses *inverse* decode dan decompresi memberikan hasil penghematan dan efisiensi penggunaan energi, *bandwidth* serta dapat menjadi alternatif yang dapat dipertimbangkan untuk digunakan pada jaringan ZigBee yang hanya memiliki kapasitas *bandwidth* sekitar 250 kbps. Selain itu, pengujian decode dan decompresi yang dilakukan menghasilkan tidak ada data yang berubah selama proses tersebut dilakukan. Data yang dikirimkan dan diterima memiliki identitas informasi dan konten yang sama.

DAFTAR PUSTAKA

[1] I. Nassra and J. V Capella, "Data compression techniques in IoT-enabled wireless body sensor networks: A systematic literature review and research trends for QoS improvement," *Internet of Things*, vol. 23, p. 100806, 2023, doi: https://doi.org/10.1016/j.iot.2023.100806.

[2] M. Mutiarani and R. R. Ritonga, "IoT LED Control System Implementation and Optimization Using Wi-Fi Through a Telegram Bot," *J. Comput. Sci. Informatics Eng.*, vol. 4, no. 1, pp. 10–20, 2025, doi: https://doi.org/10.55537/cosie.v4i1.959.

[3] L. Uziah, L. D. Samsumar, and A. Subki, "Penerapan Internet of Things (IoT) untuk Pemantauan dan Pengendalian Otomatis Pemupukan Tanaman Bawang Merah di Desa Perampuan," *J. Comput. Sci. Informatics Eng.*, vol. 3, no. 4, pp. 199–210, 2024, doi: https://doi.org/10.55537/cosie.v3i4.946.

[4] A. Ghosh, A. Raha, and A. Mukherjee, "Energy-Efficient IoT-Health Monitoring System using Approximate Computing," *Internet of Things*, vol. 9, p. 100166, 2020, doi: https://doi.org/10.1016/j.iot.2020.100166.

[5] S. Al Fallah, M. Arioua, and A. El Oualkadi, "Lightweight Secure Compression Scheme for Green IoT Applications," *Procedia Comput. Sci.*, vol. 236, pp. 363–370, 2024, doi: https://doi.org/10.1016/j.procs.2024.05.042.

[6] L. D. Samsumar, A. Subki, and A. Akbar, "Pemanfaatan Teknologi Iot Dalam Sistem Pemupukan Otomatis Untuk Meningkatkan Efisiensi Dan Produktivitas Tanaman Tembakau," *J. Comput. Sci. Informatics Eng.*, vol. 3, no. 4, pp. 170–183, 2024, doi: https://doi.org/10.55537/cosie.v3i4.933.

[7] J. N. Jakawa, F. Gonten, D. U. Emmanuel, D. A. Pandok, and P. C. Maikano, "Systematic Survey Analysis of the Application of Artificial Intelligence Base Network on Grid Computing Techniques," *J. Inf. Syst. Technol. Res.*, vol. 3, no. 3, pp. 125–135, 2024, doi: https://doi.org/10.55537/jistr.v3i3.908.

[8] M. J. Khani and Z. Shirmohammadi, "UEELC: An Ultra Energy Efficient Lossless Compression Method for Wireless Body Area Networks," in *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, 2020, pp. 1–7. doi: 10.1109/ICSPIS51611.2020.9349604.

[9] H. Mohammadi, A. Ghaderzadeh, and A. Sheikh Ahmadi, "A Novel Hybrid Medical Data Compression Using Huffman Coding and LZW in IoT," *IETE J. Res.*, vol. 69, no. 11, pp. 7831–7845, Nov. 2023, doi: 10.1080/03772063.2022.2052985.

[10] L. K. Ketshabetswe, A. M. Zungeru, C. K. Lebekwe, and B. Mtengi, "Energy-efficient Algorithms for lossless Data Compression Schemes in Wireless Sensor Networks," *Sci.*

e-ISSN 2827-8356
p-ISSN 2827-9999

African, vol. 23, p. e02008, 2024, doi: https://doi.org/10.1016/j.sciaf.2023.e02008.

[11] A. S. Nasif, Z. A. Othman, N. S. Sani, M. K. Hasan, and Y. Abudaqqa, "Huffman Deep Compression of Edge Node Data for Reducing IoT Network Traffic," *IEEE Access*, vol. 12, pp. 122988–122997, 2024, doi: 10.1109/ACCESS.2024.3452669.

[12] E. Guguloth, S. Vadtya, T. Kudithi, M. R. Shanmugam, and A. N. Banoth, "FPGA implementation of high throughput encoder and decoder design of lossless canonical Huffman machine," *Results Eng.*, vol. 26, p. 105037, 2025, doi: https://doi.org/10.1016/j.rineng.2025.105037.

[13] S. A. Babu, R. J. S. Raj, A. X. VM, and N. Muthukumaran, "DCT based Enhanced Tchebichef Moment using Huffman Encoding Algorithm (ETMH)," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 522–527. doi: 10.1109/ICICV50876.2021.9388504

[14] T. A. Selian, N. Akbar, and M. I. Hidayat, "Image Classification Based On Color Using Thresholding Method," *JITCoS J. Inf. Technol. Comput. Syst.*, vol. 1, no. 1, pp. 1–6, 2025. doi: https://doi.org/10.65230/jitcos.v1i1.5

[15] Z. H. Sain, H. H. Loupias, R. A. Susanti, R. Serban, and C. C. Thelma, "Innovative Integration of Computer Network Technology in Modern Educational Systems," *J. Inf. Syst. Technol. Res.*, vol. 3, no. 3, pp. 117–124, 2024, doi: https://doi.org/10.55537/jistr.v3i3.902.

[16] A. Nugroho and N. F. Rahmadani, "Web-based Visit List Information System at the Ministry of Religious Affairs of Deli Serdang Regency," *J. Metrokom Media Tek. Elektro dan Komput.*, vol. 1, no. 2, pp. 58–74, 2024, doi: https://doi.org/10.1307/metrokom.v1i2.79.

[17] R. A. Yahya, R. A. Siregar, and B. A. Sitompul, "Application of K-Means Cluster Algorithm to Determine Student Achievement," *JITCoS J. Inf. Technol. Comput. Syst.*, vol. 1, no. 1, pp. 16–24, 2025. doi: https://doi.org/10.65230/jitcos.v1i1.9

[18] F. Aulia, D. Suryandi, and J. Nainggolan, "Implementation of Finite State Automata on Pizza Vending Machine System," *JITCoS J. Inf. Technol. Comput. Syst.*, vol. 1, no. 1, pp. 7–15, 2025. doi: https://doi.org/10.65230/jitcos.v1i1.3

[19] C.-C. Chen, C.-C. Chang, and K. Chen, "High-capacity reversible data hiding in encrypted image based on Huffman coding and differences of high nibbles of pixels," *J. Vis. Commun. Image Represent.*, vol. 76, p. 103060, 2021. doi: 10.1016/j.jvcir.2021.103060

[20] M. Meftah, A. A. Pacha, and N. Hadj-Said, "DNA encryption algorithm based on Huffman coding," *J. Discret. Math. Sci. Cryptogr.*, vol. 25, no. 6, pp. 1831–1844, 2022. doi: 10.1080/09720529.2020.1818450