JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 2, Juni 2025, Pp. 1474-1481

IDENTIKASI JENIS *FILE* PADA ARTEFAK DIGITAL MENGGUNAKAN ALGORITMA *K-NEAREST NEIGHBOR*

Ihsan Fawzan*[1)], Ahmad Luthfi[2)]
1. Magister Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, Indonesia
2. Magister Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, Indonesia

Article Info
Kata Kunci: Artefak Digital; *file Corrupt*; Forensik Digital; Identifikasi Jenis *file*; *K-Nearest Neighbor*

Keywords: *Artefak Digital; file Corrupt; Forensik Digital; file Type Identification; K-Nearest Neighbor*

Article history:
Received 11 November 2024
Revised 15 Desember 2024
Accepted 14 Januari 2025
Available online 15 Maret 2025

DOI :
https://doi.org/10.29100/jipi.v10i2.6263

* Corresponding author.
Corresponding Author
E-mail address:
21917030@students.uii.ac.id

ABSTRAK
Permasalahan yang dihadapi dalam penelitian ini berkaitan dengan adanya isu yang timbul akibat kerusakan *file* digital dalam konteks hukum, serta kontribusi penelitian ini dalam mengatasi permasalahan tersebut. Virus, malfungsi sistem, dan *malware* menjadi beberapa penyebab terjadinya *file* rusak sehingga menghambat akses menuju data penting dalam proses hukum. Teknik yang sesuai dalam menganalisis konten *file* dan mengidentifikasi pola menggunakan algoritma untuk mengatasi masalah yaitu menggunakan teknik *content-based*. Penelitian ini memanfaatkan algoritma *K-Nearest Neighbor* dalam *machine learning* untuk mendeteksi jenis *file* pada *file* yang rusak. Penelitian yang mengkaji tentang identifikasi jenis *file* sudah pernah dilakukan sebelumnya, namun masih menggunakan dataset lama yaitu *GovDocs* yang dirilis pada tahun 2009 sehingga perlu adanya penelitian yang menggunakan dataset baru. Penelitian ini memperbarui dataset GovDocs ke dalam NapierOne, yang berkontribusi pada peningkatan aksesibilitas data yang relevan untuk analisis. *Machine learning* digunakan dalam penelitian ini untuk mengklasifikasikan data dan berhasil meningkatkan keterbacaan dokumen meskipun tanpa informasi *header* atau *footer*. Selain itu, penelitian yang penulis lakukan dalam mengidentifikasi jenis *file* ambigu dalam artefak digital menggunakan *K-Nearest Neighbor* memperoleh hasil yang tinggi dengan tingkat akurasi mencapai 86%. Secara keseluruhan, studi ini berkontribusi pada peningkatan aksesibilitas dan keandalan bukti digital dalam konteks hukum, khususnya terkait *file* yang mengalami kerusakan.

ABSTRACT
The problems faced in this research are related to the issues that arise due to damage to digital files in a legal context, as well as the contribution of this research in overcoming these problems. Viruses, system malfunctions, and malware are some of the causes of damaged files, thus preventing access to important data in legal proceedings. The appropriate technique for analyzing file content and identifying patterns using algorithms to solve problems is using content-based techniques. This research utilizes the K-Nearest Neighbor algorithm in machine learning to detect file types in damaged files. Research that examines file type identification has been carried out before, but still uses the old dataset, namely GovDocs, which was released in 2009, so there is a need for research that uses a new dataset. This research updates the GovDocs dataset into NapierOne, contributing to increased accessibility of relevant data for analysis. Machine learning was used in this research to classify data and succeeded in improving document readability even without header or footer information. In addition, the research that the author conducted in identifying ambiguous file types in digital artifacts using K-Nearest Neighbor obtained high results with an accuracy rate of 86%. Overall, this study contributes to improving the accessibility and reliability of digital evidence in legal contexts, especially regarding damaged files.

.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 2, Juni 2025, Pp. 1474-1481

I. PENDAHULUAN

KEJAHATAN siber dalam era digital saat ini semakin mengkhawatirkan, kejahatan tersebut mencakup serangan terhadap informasi pribadi, penipuan online, dan penyebaran *malware* yang merugikan. Fenomena ini tidak hanya mengganggu individu dan perusahaan, tetapi bisa juga menimbulkan ancaman serius terhadap keamanan nasional. Pusiknas Polri menindak 8.831 kasus kejahatan siber pada tahun 2022, jumlah tindak pidana kejahatan siber tersebut mengalami peningkatan signifikan dibandingkan dengan tahun 2021 yaitu mencapai 14 kali lipat lebih tinggi[1]. Kasus dengan jumlah penindakan terbanyak ada pada manipulasi data autentik yaitu sebanyak 3.723 kasus disusul dengan penipuan melalui media elektronik sebanyak 2.131 kasus.

Serangan manipulasi data terjadi ketika penjahat tidak mengambil data, namun melakukan penyesuaian halus dan tersembunyi pada data untuk mendapatkan keuntungan atau efek tertentu. Modifikasi data ini dapat merugikan organisasi seperti halnya pelanggaran data. Manipulasi data dapat mengakibatkan distorsi persepsi karena pergeseran data yang menyebabkan kerugian finansial atau bahkan potensi hilangnya nyawa tergantung pada sistem yang dimaksud dan jenis data yang diubah[2].

WhisperGate merupakan salah satu *malware* yang digunakan untuk melakukan manipulasi data yang targetnya beberapa organisasi pemerintah, nirlaba, dan teknologi informasi di Ukraina setidaknya sejak Januari 2022[3]. Salah satu teknik yang digunakan pada *malware WhisperGate* yaitu menimpa satu MB awal pada sebuah *file* dengan nilai *0xCC* dan menambahkan ekstensi acak. Teknik ini dapat menimbulkan kesulitan dalam analisis karena *file signature* yang ada pada data sudah hilang dan ekstensi *file* telah menjadi acak. Oleh karena itu, diperlukan metode tambahan untuk menentukan jenis *file* yang sebenarnya.

Selain *WhisperGate*, ada *malware* lain yang dapat digunakan untuk melakukan manipulasi data di antaranya adalah *PowerDuke* dan *Revil*. *PowerDuke* merupakan *backdoor* yang digunakan oleh APT29 pada tahun 2016, umumnya disebarkan melalui lampiran *file* Microsoft Word atau Excel yang mengandung makro berbahaya[4]. *PowerDuke* memiliki kemampuan untuk melakukan manipulasi data dengan perintah untuk menulis data acak di seluruh *file* dan menghapusnya. *Revil* merupakan *Ransomware-as-a-Service* (RaaS) yang menyerang sektor manufaktur, transportasi dan elektronik sejak tahun 2019[5]. Teknik yang digunakan untuk manipulasi data termasuk mengenkripsi *file* pada *host* target dan kemudian menjalankan *query* pada *registry* untuk menemukan ekstensi *file* acak yang akan ditambahkan ke *file* yang telah dienkripsi.

Menurut [6] salah satu cara untuk melakukan identifikasi jenis *file* adalah menggunakan analisis *content-based*. Analisis ini memanfaatkan perhitungan konten *file* untuk mengidentifikasi pola atau *pattern* secara statistik yang kemudian digunakan untuk proses klasifikasi data[7]. Analisis *content-based* dilakukan dengan memeriksa isi *file* yang telah dimanipulasi secara statistik dan membandingkannya dengan *file* asli lainnya. Proses ini memerlukan dataset *file* asli sebagai referensi untuk mengidentifikasi jenis *file* dari *file* yang telah dimodifikasi. Metode ini memungkinkan penggunaan informasi yang terdapat dalam *file* untuk menghasilkan pengelompokan yang lebih akurat berdasarkan karakteristik yang terukur. Artificial intelligence dapat dimanfaatkan untuk melakukan proses analisis statistik guna memudahkan analisis lebih lanjut pada *file* yang telah dimanipulasi. Selain content-based analisis, ada 2 jenis metode lain untuk melakukan identifikasi jenis *file* yaitu *extention-based*[6], [8], [9] dan *magic-bytes based*[8]–[10]. *Extention-based* memiliki kelebihan yaitu penggunaan yang mudah untuk melakukan analisis karena cukup melihat ekstensi *file* yang biasanya terdiri dari 3 karakter setelah titik misalnya ".jpg". Namun teknik ini juga memiliki kelemahan yaitu saat ekstensi *file* dilakukan manipulasi misal mengganti dengan ekstensi tertentu, maka *file* tersebut tidak akan bisa dikenali lagi. Metode selanjutnya adalah *magic-bytes-based* yaitu teknik identifikasi jenis *file* dengan melihat *magic-bytes* atau *file signature* yang ada pada *file*. *Magic-bytes* ini merupakan nilai *hexa* unik yang terdapat pada *header file* dengan panjang yang bervariasi dari dua hingga empat puluh enam *bytes*[10]. Namun, identifikasi jenis *file* menggunakan pendekatan *magic-bytes* juga memiliki kelemahan yaitu hanya dapat digunakan untuk jenis *file binary*, dan hanya jika *file* tersebut memiliki *magic-bytes* yang relevan dengan jenis *file* tersebut.

Penggunaan *artificial intelligence* untuk melakukan klasifikasi telah dilakukan pada penelitian terdahulu di antaranya penggunaan algoritma *Convolutional Neural Network* [11]–[14], *Feedforward Neural Network*[11], [12], [15], *Multilayer Perceptron* [16], [17], *Support Vector Machine* [16], [18], [19], *Decission Tree* [20], dan *K-Nearest Neighbor*[9], [16], [21]. Namun, sebagian besar penelitian yang telah dilakukan bergantung pada dataset lama seperti *GovDocs* yang dikeluarkan pada tahun 2009 sehingga perlu ada penelitian baru yang menggunakan dataset terbaru. *NapierOne* adalah solusi keamanan siber modern yang mengelola berbagai jenis *file*, dengan salah satu fungsinya adalah untuk mendeteksi *ransomware* dan melakukan analisis forensik digital[22]. NapierOne dikembangkan untuk mengatasi masalah dalam hal reproduktifitas dan meningkatkan konsistensi melalui

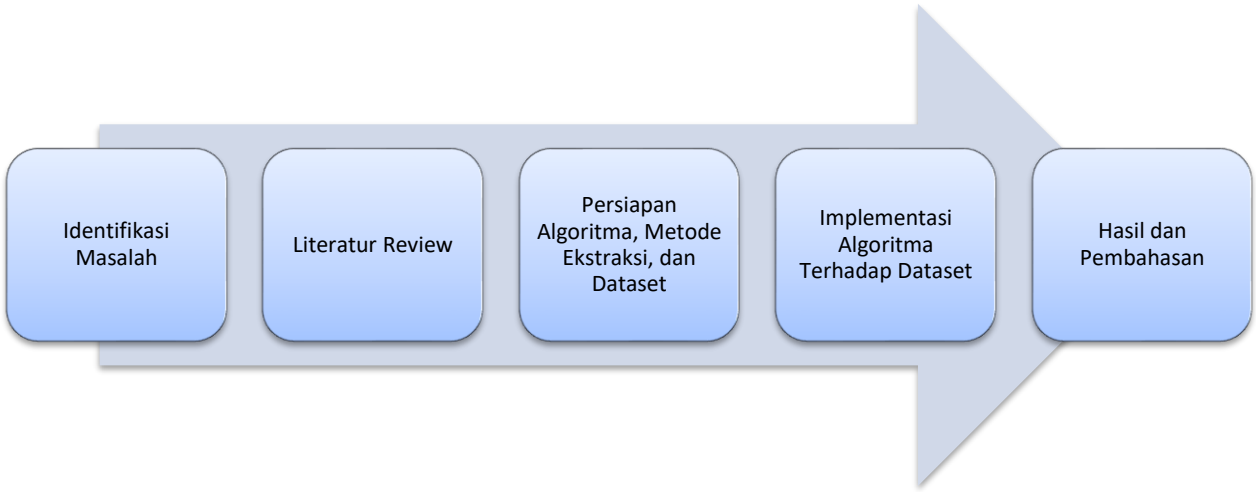
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 2, Juni 2025, Pp. 1474-1481

penyederhanaan proses replikasi dan pengulangan penelitian serta terinspirasi oleh Govdocs1 dan dirancang untuk berfungsi sebagai pelengkap bagi kumpulan data asli tersebut.

Penelitian yang penulis lakukan berfokus pada mengembangkan sebuah model klasifikasi menggunakan dataset baru yaitu *NapierOne* dengan menerapkan algoritma *K-Nearest Neighbor*. Maka dari itu manfaat dari penelitian ini adalah untuk membantu melakukan klasifikasi jenis *file* pada data yang telah dimanipulasi menggunakan algoritma *K-Nearest Neighbor* pada dataset *NapierOne* serta melihat akurasi dari model yang telah dikembangkan.

II. METODE

Prosedur yang dilakukan dalam penelitian ini dibagi menjadi beberapa tahap secara ilmiah sehingga dapat dipertanggung jawabkan dan menghasilkan sebuah solusi logis dan sistematis yang diilustrasikan dalam Gambar 1.



Gambar 1 Metode Penelitian

Penelitian dilakukan dengan prosedur yang telah didefinisikan pada Gambar 1 yang dibagi menjadi 5 tahap yaitu:

Tahapan pertama adalah mengidentifikasi masalah yang ada saat melakukan analisis artifak digital dan ditemukan masalah adanya serangan manipulasi data. *WhisperGate* merupakan salah satu jenis *malware* yang dapat melakukan manipulasi data dengan menimpa satu MB awal pada sebuah *file* dengan nilai 0xCC dan menambahkan ekstensi acak.

Tahapan kedua yaitu melakukan studi literatur untuk mengukur sejauh apa penelitian terdahulu menjawab tantangan yang timbul saat melakukan analisis artefak digital pada data yang telah dimanipulasi. Pada penelitian terdahulu penggunaan *artificial intelligence* banyak digunakan untuk melakukan analisis dengan pendekatan *content-based*, namun dataset yang digunakan masih menggunakan dataset lama yaitu *GovDocs* yang dirilis pada tahun 2009 sehingga perlu adanya pengembangan model *artificial intelligence* dengan dataset baru.

Tahapan ketiga yaitu Persiapan algoritma, metode ekstraksi, dan dataset guna memperbaiki kekurangan yang ada pada penelitian terdahulu. Berdasarkan penelitian terdahulu banyak menggunakan dataset lama seperti *GovDocs*, maka dari itu penelitian yang dilakukan oleh penulis memanfaatkan dataset NapierOne yang dirilis pada tahun 2022 dan menerapkan pendekatan kecerdasan buatan menggunakan algoritma K-Nearest Neighbor, dengan ekstraksi fitur melalui *byte frequency distribution*.

Tahapan keempat yaitu melakukan Implementasi algoritma *K-Nearest Neighbor* terhadap dataset *NapierOne*.

Tahapan kelima yaitu Hasil dan Pembahasan guna mengukur apakah penggunaan algortima *K-Nearest Neighbor* dengan ekstraksi fitur *byte frequency distribution* pada dataset *NapierOne* dapat digunakan untuk melakukan klasifikasi dan identifikasi jenis *file* pada data yang telah dimanipulasi.

A. *Dataset*

Dataset yang digunakan adalah *NapierOne* yang dirilis pada 2022. Dataset tersebut memiliki 41 jenis *file* dengan masing-masing 5000 *file* pada setiap jenisnya. Tabel 1 merupakan informasi jenis *file* yang ada pada dataset *NapierOne*, namun pada penelitian ini hanya 23 jenis *file* saja yang digunakan dan dirinci pada Tabel 2.

TABEL 1
JENIS *FILE* PADA DATASET *NAPIERONE*

APK	BIN	BMP	CSS	CSV	DOC

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 2, Juni 2025, Pp. 1474-1481

DOCX	DWG	ELF	EPS	EPUB	EXE
GIF	GZIP	HTML	ICS	JS	JPG
JSON	MKV	MP3	MP4	ODS	OXPS
PDF	PNG	PPT	PPTX	PS1	RAR
SVG	TAR	TIF	TXT	WEBP	XLS
XLSX	XML	ZIP	ZLIB	7Zip	

TABEL 2
JENIS *FILE* YANG DIGUNAKAN DALAM PENELITIAN

APK	BIN	BMP	BMPW	DOC	DOCX
DWG	ELF	EPUB	EXE	GIF	GIFW
JPG	MKV	MP3	MP4	ODS	OXPS
PDF	PNG	PNGW	PPT	PPTX	

B. *Ekstraksi Fitur*

Byte frequency distribution digunakan untuk mengekstraksi fitur dari dataset *NapierOne*. *Byte frequency distribution* ini menunjukkan setiap nilai byte dalam rentang 0 hingga 255 dan dapat direpresentasikan dalam bentuk histogram[18]. *Byte frequency distribution* sering digunakan untuk melakukan kompresi data[23] , kriptografi[23], deteksi malware[24]–[27], dan analisis *file*[28], [29] sehingga dipilih dalam penelitian ini sebagai pendekatan ekstraksi fitur.

Gambar 2 adalah contoh *output* dari dataset *NapierOne* yang telah menjalani proses ekstraksi fitur dengan menggunakan *byte frequency distribution*.

```
>>> dataset.head()
      0     1     2     3     4     5     6     7     8  ...   248   249   250   251   252   253   254   255 type
0  14225  2179  1728  2065  1775  1736  1857  2242  1768  ...  2196  2315  2192  2286  2289  2348  2377  2418  apk
1  61688 10818  8136  9698  7890  7386  7613  8175 11813  ...  9007  9454  8960  9535  9488  9671  9646 11814  apk
2  81275 10977  7757 10518  7438  6978  7116  7718 13003  ...  8165  8389  8218  9029  8663  8854  9046 17087  apk
3  67897  7480  4247  6132  3887  3429  3842  3749  9132  ...  3854  4121  3975  4301  4156  4263  4379 10800  apk
4  66296  7675  4483  6610  4380  3839  4261  4232  9394  ...  4487  4727  4489  4910  4916  4765  4972 10336  apk
```

Gambar 2 Hasil ekstraksi fitur menggunakan *NapierOne*

C. *K-Nearest Neighbor*

Setelah mengekstrak fitur, algoritma *K-Nearest Neighbor* digunakan untuk melakukan kalkulasi sehingga menghasilkan sebuah model yang dapat digunakan untuk melakukan identifikasi jenis *file* pada artefak digital dengan nilai K yang digunakan pada penelitian ini adalah 5. *K-Nearest Neighbor* dipilih untuk menjadi algoritma yang digunakan pada penelitian ini karena *K-Nearest Neighbor* merupakan salah satu *supervised machine learning* yang digunakan untuk melakukan klasifikasi[30]. Selain *K-Nearest Neighbor*, algoritma *Support Vector Machine* juga dapat digunakan untuk melakukan klasifikasi jenis *file*, namun setelah dilakukan percobaan hasilnya kurang memuaskan karena mencapai tingkat akurasi yang rendah yaitu 39.7%[31].

Implementasi *K-Nearest Neighbor* pada dataset *NapierOne* dibantu menggunakan python dengan *library scikit-learn*. Sebelum dilakukan proses pelatihan, dataset dilakukan proses ekstraksi fitur menggunakan *byte frequency distribution* sehingga diperoleh nilai unik dari rentang 0 sampai 255. Teknik *cross validation* digunakan untuk menentukan nilai K yang optimal, yaitu melakukan iterasi dari K=3 sampai K=31 dan diperoleh nilai K yang paling optimal yaitu K=5 dengan nilai skor rata-rata 86%. Model di evaluasi dengan melihat akurasi, presisi dan recall yang dihasilkan berdasarkan proses pelatihan. Penilaian akurasi, presisi, dan recall dibantu juga menggunakan *library scikit-learn*.

D. *Spesifikasi Hardware dan Software*

Studi ini memanfaatkan perangkat keras dan perangkat lunak pendukung untuk melakukan perhitungan dengan machine learning, yang memungkinkan identifikasi jenis *file* dalam artefak digital. Tabel 3 menyajikan daftar perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini.

TABEL 3
SPESIFIKASI *PERANGKAT KERAS* DAN *PERANGKAT LUNAK* YANG DIGUNAKAN

No	Jenis	Nama
1.	*Hardware*	Harddisk 1TB
2.	*Hardware*	Prosesor Intel i7 4750HQ

3.	*Hardware*	RAM 12GB DDR3
4.	*Software*	Sci-kit Learn
5.	*Software*	Windows 10 Pro
6.	*Software*	Python3.11

III. HASIL DAN PEMBAHASAN

Percobaan yang telah dilakukan untuk mengidentifikasi jenis *file* dalam artefak digital memanfaatkan algoritma *K-Nearest Neighbor*. Pengujian ini menggunakan *NapierOne* sebagai dataset, dengan *byte frequency distribution* sebagai metode ekstraksi fitur. *Confusion Matrix* digunakan untuk mengevaluasi kinerja model yang dihasilkan oleh algoritma *K-Nearest Neighbor* yang ditunjukan pada Tabel 4.

TABEL 4
CONFUSION MATRIX

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Rumus yang digunakan untuk melakukan perhitungan pada *Confusion Matrix* adalah sebagai berikut:

$$\frac{(TP+TN)}{(TP+FP+TN+TN)} \; x \; 100\% \quad (1)$$

$$\frac{TP}{(FP+TP)} \; x \; 100\% \quad (2)$$

$$\frac{TP}{(TP+FN)} \; x \; 100\% \quad (3)$$

Persamaan (1) merupakan persamaan yang digunakan untuk menghitung *accuracy*. Persamaan (2) merupakan persamaan yang digunakan untuk menghitung *precision*. Persamaan (3) merupakan persamaan yang digunakan untuk menghitung *recall*.
Keterangan:
TP = *True Positive*
TN = *True Negative*
FP = *False Positive*
FN = *False Negative*

Hasil ekstraksi fitur dengan menggunakan *byte frequency distribution* pada dataset *NapierOne* kemudian digunakan sebagai *input* untuk mendeteksi jenis *file* melalui *machine learning* dengan algoritma *K-Nearest Neighbor*. *Output* dari model yang dihasilkan adalah prediksi klasifikasi untuk jenis *file* yang paling mendekati. Penelitian ini dilakukan dengan metode *K-Nearest Neighbor* melalui tahap pembelajaran dan proses identifikasi jenis *file*.

K-Nearest Neighbor dapat mengklasifikasikan jenis *file* yang *corrupt* berdasarkan pola yang diperoleh dari setiap kategori *file* dalam dataset *NapierOne*. Tabel 5 menampilkan model hasil identifikasi jenis *file* yang dilakukan dengan algoritma *K-Nearest Neighbor* pada dataset *NapierOne*.

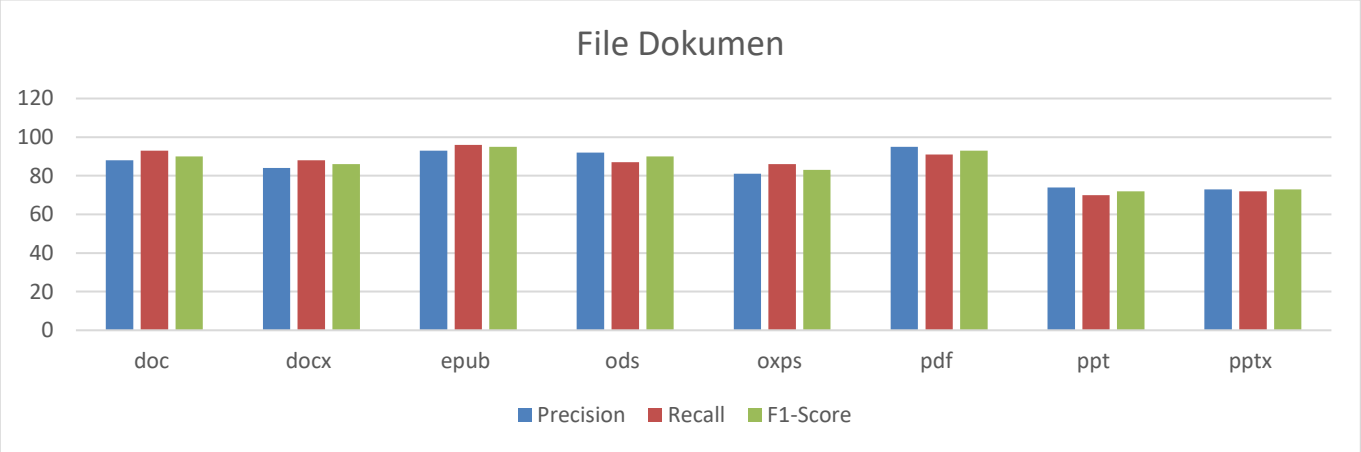
TABEL 5
HASIL PENGUJIAN MENGGUNAKAN ALGORTIMA *K-NEAREST NEIGHBOR*

Jenis *file*	Precision	Recall	F1-score	Support
apk	0.84	0.89	0.87	1491
bin	0.80	0.89	0.84	1442
bmp	0.99	0.99	0.99	1396
bmpw	0.95	0.85	0.90	839
doc	0.88	0.93	0.90	1376
docx	0.84	0.88	0.86	1495
dwg	1.00	1.00	1.00	1537
elf	0.84	0.76	0.80	1319
epub	0.93	0.96	0.95	1481
exe	0.88	0.86	0.87	1439
gif	0.97	0.99	0.98	1487
gifw	0.92	0.93	0.92	1308
jpg	1.00	1.00	1.00	1510
mkv	0.39	0.39	0.39	1431
mp3	0.98	1.00	0.99	1507
mp4	0.41	0.41	0.41	1499
ods	0.92	0.87	0.90	1481
oxps	0.81	0.86	0.83	1530
pdf	0.95	0.91	0.93	1544

png	0.98	0.98	0.98	1505
pngw	0.94	0.79	0.86	1461
ppt	0.74	0.70	0.72	643
pptx	0.73	0.72	0.73	1275
Accuracy			0.86	31996
Macro avg	0.86	0.85	0.85	31996
Weighted avg	0.86	0.86	0.86	31996

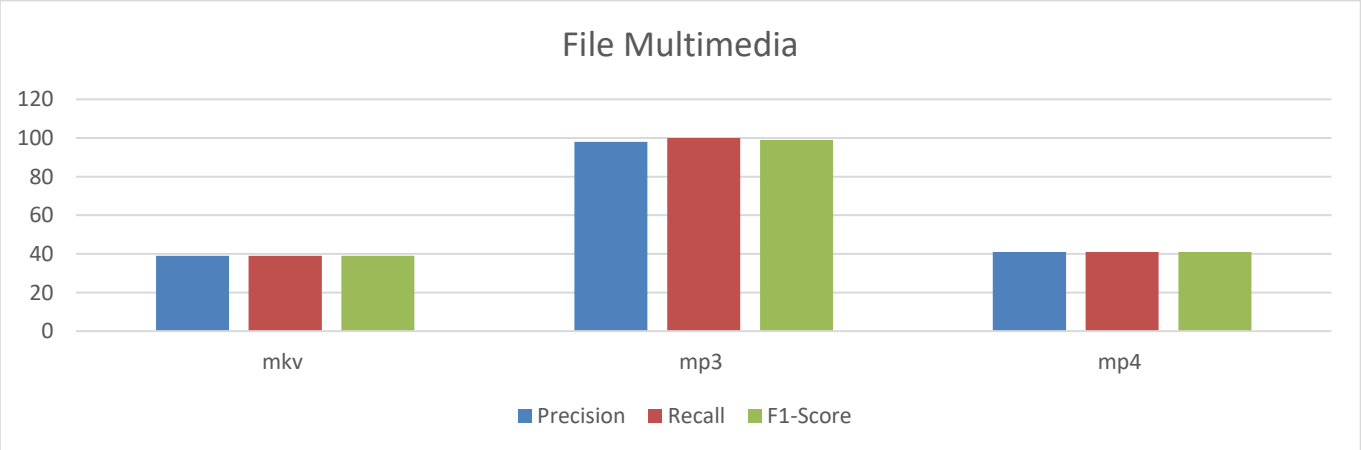
Hasil pengujian model *K-Nearest Neighbor* memperoleh nilai akurasi sebesar 86%, presisi 86%, recall 85%, dan F1-score 85%. Berdasarkan pengujian ini, identifikasi jenis *file* dalam artefak digital menggunakan algoritma *K-Nearest Neighbor* dapat dilakukan dengan tingkat akurasi mencapai 86%

Pengujian ini menunjukkan nilai akurasi, presisi, dan recall yang lebih tinggi dibandingkan dengan hasil penelitian [21], yang mencatat akurasi sebesar 72%, presisi 74%, dan recall 72% menggunakan algoritma *K-Nearest Neighbor*. Penelitian tersebut juga menggunakan algoritma yang sama tetapi dengan pendekatan ekstraksi fitur yang berbeda, yaitu *Byte2Vec*. Selain itu, hasil dari penelitian ini juga memiliki nilai akurasi yang lebih tinggi dibandingkan dengan penelitian [9] yang menggunakan sampel ukuran dari 64, 128, 256, 512, 1000, 2000, dan 4000 *bytes*.



Gambar 3 Hasil Pengujian pada *file* Dokumen

Gambar 3 menggambarkan hasil pengujian yang dilakukan pada *file* dokumen. Penerapan distribusi frekuensi byte pada dataset NapierOne dengan algoritma K-Nearest Neighbor memperoleh nilai tinggi, dengan presisi *file* jenis dokumen yang melebihi 70%.



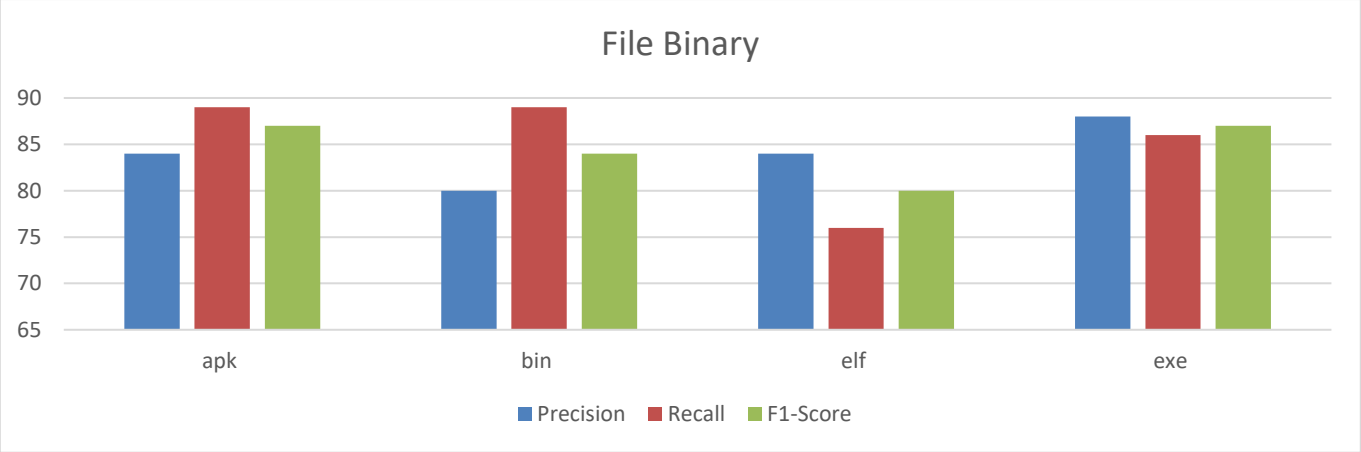
Gambar 4 Hasil Pengujian pada *file* Multimedia

Gambar 4 menunjukkan hasil pengujian yang dilakukan pada *file* multimedia. Penerapan *byte frequency distribution* pada dataset *NapierOne* dengan algoritma *K-Nearest Neighbor* memperoleh nilai yang rendah untuk *file* dengan format MKV dan MP4, sementara *file* dengan format MP3 memperoleh nilai yang tinggi.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 10, No. 2, Juni 2025, Pp. 1474-1481

Gambar 5 Hasil Pengujian pada *file* Image

Gambar 5 menunjukkan hasil pengujian pada *file* gambar. Penerapan *byte frequency distribution* pada dataset *NapierOne* dengan algoritma *K-Nearest Neighbor* menghasilkan nilai tinggi, dengan presisi *file* jenis gambar melebihi 90%.



Gambar 6 Hasil Pengujian pada *file* Binary

Gambar 6 menyajikan hasil pengujian yang dilakukan pada *file* biner. Penerapan *byte frequency distribution* pada dataset *NapierOne* dengan algoritma *K-Nearest Neighbor* menunjukkan nilai rendah untuk *file* dengan format BIN dan ELF, sementara *file* dengan format APK dan EXE menunjukkan nilai yang tinggi.

IV. KESIMPULAN

Artefak digital memiliki peranan yang signifikan dalam digital forensik, terutama dalam memberikan bukti yang dapat dipertanggung jawabkan dalam konteks kasus kejahatan di hadapan hukum. Namun dalam proses akuisisi data, ahli forensik dapat menemui *malware* tertentu, seperti *WhisperGate*, yang memiliki kemampuan untuk memanipulasi data. Oleh karena itu, identifikasi jenis *file* terlebih dahulu menjadi langkah penting yang perlu dilakukan.

Penelitian ini menganalisis metode untuk mengidentifikasi jenis *file* pada artefak digital dengan memanfaatkan algoritma *K-Nearest Neighbor* pada dataset *NapierOne* melalui teknik ekstraksi fitur *byte frequency distribution*. Hasil pengujian menunjukan bahwa penerapan algoritma *K-Nearest Neighbor* dengan dataset *NapierOne* dan ekstraksi fitur *byte frequency distribution* untuk identifikasi jenis *file* dapat mencapai akurasi sebesar 86%.

DAFTAR PUSTAKA

[1] "Kejahatan Siber di Indonesia Naik Berkali-kali Lipat | Pusiknas Bareskrim Polri." https://pusiknas.polri.go.id/detail_artikel/kejahatan_siber_di_indonesia_naik_berkali-kali_lipat (accessed Jun. 16, 2023).

[2] R. A. Nettles, C. Merulla, and S. Warzala, "Data Manipulation: Attacks and Mitigation – CSIAC," 2019. https://csiac.org/articles/data-manipulation-attacks-and-mitigation/ (accessed Jul. 01, 2024).

[3] P. Taylor, B. Security, M. Brenton, and Z. G. I. Security, "WhisperGate, Software S0689 | MITRE ATT&CK®," 2024. https://attack.mitre.org/software/S0689/.

[4] "PowerDuke, Software S0139 | MITRE ATT&CK®," 2020. https://attack.mitre.org/software/S0139/.
[5] E. Millington, "REvil, Software S0496 | MITRE ATT&CK®," 2014. https://attack.mitre.org/software/S0496/.
[6] K. Karampidis and G. Papadourakis, "*File* Type Identification - Computational Intelligence for Digital Forensics," *J. Digit. Forensics, Secur. Law*, vol. 12, no. 2, 2017, doi: 10.15394/jdfsl.2017.1472.
[7] S. K. Konaray, A. Toprak, G. M. Pek, H. Akçekoce, and K. Deniz, "Detecting *File* Types Using Machine Learning Algorithms," *2019 Innov. Intell. Syst. Appl. Conf.*, pp. 1–4, 2019.
[8] A. Dubettier, T. Gernot, E. Giguet, and C. Rosenberger, "*File* type identification tools for digital investigations," *Forensic Sci. Int. Digit. Investig.*, vol. 46, p. 301574, 2023, doi: 10.1016/j.fsidi.2023.301574.
[9] Y. Wang, Z. Su, and D. Song, "*File* Fragment Type Identification with Convolutional Neural Networks," in *Proceedings of the 2018 International Conference on Machine Learning Technologies*, 2018, pp. 41–47, doi: 10.1145/3231884.3231889.
[10] D. J. Hickok, D. R. Lesniak, and M. C. Rowe, "*File* Type Detection Technology," *38th Midwest Instr. Comput. Symp.*, pp. 23–28, 2005.
[11] A. Bhat, A. Likhite, S. Chavan, and L. Ragha, "*File* Fragment Classification using Content Based Analysis," *ITM Web Conf.*, vol. 40, p. 03025, Aug. 2021, doi: 10.1051/itmconf/20214003025.
[12] L. Hiester, "*File* Fragment Classification Using Neural Networks with Lossless Representations Networks with Lossless Representations," *Undergrad. Honor. Theses*, pp. 1–32, 2018, [Online]. Available: https://dc.etsu.edu/honors/454.
[13] M. Al Neaimi, H. Al Hamadi, C. Y. Yeun, and M. J. Zemerly, "Digital Forensic Analysis of *File*s Using Deep Learning," in *2020 3rd International Conference on Signal Processing and Information Security (ICSPIS)*, 2020, pp. 1–4, doi: 10.1109/ICSPIS51252.2020.9340141.
[14] G. Mittal, P. Korus, and N. Memon, "FiFTy: Large-Scale *File* Fragment Type Identification Using Convolutional Neural Networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, no. 1, pp. 28–41, 2021, doi: 10.1109/TIFS.2020.3004266.
[15] K. Vulinovic, L. Ivkovic, J. Petrovic, K. Skracic, and P. Pale, "Neural networks for *file* fragment classification," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings*, 2019, pp. 1194–1198, doi: 10.23919/MIPRO.2019.8756878.
[16] M. Masoumi, A. Keshavarz, and R. Fotohi, "*File* fragment recognition based on content and statistical features," *Multimed. Tools Appl.*, vol. 80, no. 12, pp. 18859–18874, May 2021, doi: 10.1007/s11042-021-10681-x.
[17] S. K. Venkata and A. Green, "Computational Intelligence to aid Text *File* Format Identification," pp. 1–14, 2019, [Online]. Available: http://eprints.rclis.org/38969/.
[18] F. Wang, T. Quach, J. Wheeler, J. B. Aimone, and C. D. James, "Sparse Coding for N-Gram Feature Extraction and Training for *File* Fragment Classification," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 10, pp. 2553–2562, Oct. 2018, doi: 10.1109/TIFS.2018.2823697.
[19] C. Zhang and R. Green, "Communication security in internet of thing: Preventive measure and avoid DDoS attack over IoT network," *Simul. Ser.*, vol. 47, no. 3, pp. 8–15, 2015.
[20] K. Nguyen, D. Tran, W. Ma, and D. Sharma, "Decision tree algorithms for image data type identification," *Log. J. IGPL*, vol. 25, no. 1, pp. 67–82, 2017, doi: 10.1093/jigpal/jzw045.
[21] M. E. Haque and M. E. Tozal, "Byte embeddings for *file* fragment classification," *Futur. Gener. Comput. Syst.*, vol. 127, pp. 448–461, Feb. 2022, doi: 10.1016/j.future.2021.09.019.
[22] S. R. Davies, R. Macfarlane, and W. J. Buchanan, "NapierOne: A modern mixed *file* data set alternative to Govdocs1," *Forensic Sci. Int. Digit. Investig.*, vol. 40, p. 301330, Mar. 2022, doi: 10.1016/j.fsidi.2021.301330.
[23] F. De Gaspari, D. Hitaj, G. Pagnotta, L. De Carli, and L. V. Mancini, "Reliable detection of compressed and encrypted data," *Neural Comput. Appl.*, vol. 34, no. 22, pp. 20379–20393, 2022, doi: 10.1007/s00521-022-07586-7.
[24] S. Yu, S. Zhou, L. Liu, R. Yang, and J. Luo, "Detecting malware variants by byte frequency," *J. Networks*, vol. 6, no. 4, pp. 638–645, 2011, doi: 10.4304/jnw.6.4.638-645.
[25] G. Y. Kim, J. Y. Paik, Y. Kim, and E. S. Cho, "Byte Frequency Based Indicators for Crypto-Ransomware Detection from Empirical Analysis," *J. Comput. Sci. Technol.*, vol. 37, no. 2, pp. 423–442, 2022, doi: 10.1007/s11390-021-0263-x.
[26] N. Singh and S. S. Khurmi, "ByteFreq: Malware clustering using byte frequency," *2016 5th Int. Conf. Reliab. Infocom Technol. Optim. ICRITO 2016 Trends Futur. Dir.*, pp. 333–337, 2016, doi: 10.1109/ICRITO.2016.7784976.
[27] X. Huang, L. Ma, W. Yang, and Y. Zhong, "A Method for Windows Malware Detection Based on Deep Learning," *J. Signal Process. Syst.*, vol. 93, no. 2–3, pp. 265–273, 2021, doi: 10.1007/s11265-020-01588-1.
[28] T. Suzuki, "Experimental Comparison of ASCII Art Extraction Methods : a Run-Length Encod ing based Method and a Byte Pattern based Method," vol. 8, no. 2, pp. 57–68.
[29] N. F. B. A. KADIR, "Statistical Byte Frequency Analysis for Identifying Jpeg," no. January, 2015.
[30] A. Kumar, K. S. Kuppusamy, and G. Aghila, "FAMOUS: Forensic Analysis of MObile devices Using Scoring of application permissions," *Futur. Gener. Comput. Syst.*, vol. 83, pp. 158–172, Jun. 2018, doi: 10.1016/j.future.2018.02.001.
[31] T. Xu, M. Xu, Y. Ren, J. Xu, H. Zhang, and N. Zheng, "A *File* Fragment Classification Method Based on Grayscale Image," *J. Comput.*, vol. 9, no. 8, pp. 1863–1870, 2014, doi: 10.4304/jcp.9.8.1863-1870.