SEMARAK ILMU
PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

A Forensic Intelligence System for Identification of Data Originality Based on Signature Files

Nur Widiyasono[1], Randi Rizal[1,2,*], Siti Yuliyanti[1], Siti Rahayu Selamat[2], Mugi Praseptiawan[3]

[1] Department of Informatics, Faculty of Engineering, Siliwangi University, Tasikmalaya, 46115 Indonesia
[2] Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia
[3] Department of Informatics, Institute of Technology Sumatera, ITERA, 35365 Indonesia

ARTICLE INFO

Article history:
Received 19 August 2023
Received in revised form 11 January 2024
Accepted 15 January 2024
Available online 10 July 2024

Keywords:

Signature file; extension; modified file; forensic intelligence; data originality

ABSTRACT

The difficulty of maintaining the authenticity of files is a security problem that must be corrected in the process of developing information technology. One example of a case that often occurs is the modification of the file extension. This happens due to human error or deliberate and automatic factors. The method used for analysing the extension of a file is signature file analysis. This method is used to detect crimes that use techniques to change file extensions to hide content in its original form. Research related to the modification of file extensions using file signature analysis has been done before. However, this research still has many weaknesses, one of which is that the process of checking the file signature and the appropriate file extension is done manually and is too time-consuming. So, the forensic investigation process carried out in this case was not efficient. In this research, as a solution to the above problems, the forensic intelligence system was created to identify file types by automatically matching file extensions and signatures. If the file entered is modified, the output given is the name of the file entered, the size of the file, the file signature, the original extension of the file, and the time the file was uploaded to the application. In addition, this application can restore files with modified extensions to their original extensions. The extensions used for this research experiment amounted to 22 types out of a total of 130 types of extensions.

1. Introduction

Security issues are important aspects [1], [2] that must be corrected in the process of developing information technology [2,3], including security in maintaining data authenticity [5]. It is difficult to maintain data authenticity when computer forensic analysts access files that still exist or have been deleted as well as supporting evidence for reported files, such as: time of investigation, name of the analyst, actual file location [5,6]. The file identification process begins with differentiating file formats, some operating systems use file extensions [8]. File extensions are easily lost or changed by human interaction (including human error) or by automated processes. Many cases in digital

[*] *Corresponding author.*
E-mail address: randirizal@unsil.ac.id

https://doi.org/10.37934/araset.48.1.193204

forensics involve modifying the file extension on one or more files on digital media [9]. Modified files make the analysis process difficult. The process of examining digital media by computer forensics [10] aims to identify, obtain, maintain, recover, analyse, and present the results of identification of files stored electronically on computer media [10-12]. The file identification process begins with differentiating file formats, some operating systems use file extensions. File extension is a character or group of characters added after forming the entire file name [14]. The main purpose of implementing a file extension is to indicate to the operating system which software should be used to access the file. File extensions are easily lost or changed by human interaction (including human error). Many cases in digital forensics involve modifying the file extension on one or more files on digital media[14,15]. Modified files make the analysis process difficult [17].

Overcoming this problem, the implementation of a more comprehensive data analysis method is needed to support computer forensic processes in file identification, including Signature File analysis. Signature files can be used to detect crimes that use techniques to change file extensions to hide content from its original form. However, if the file extension is changed, the application normally used to access the file cannot recognize the file before it is restored as before. Also, it has been done in previous research, namely in research conducted [17-20] explaining file identification in analysing the authenticity of a data. According to research [21], signature files are used for criminal case investigations in identifying [22] and verifying file types so that modified files can be restored and can be read by the operating system [23]. The file extension is a character or group of characters that is added after forming the entire file name. The main purpose of implementing a file extension is to indicate to the operating system which software should be used to access the file [24]. However, from the research that has been done before, there are still some drawbacks, including manually checking file signatures and file extensions is too time consuming and this process can be fully automated using applications such as EnCase and relies entirely on the list of file signatures that have been updated and contain every signature hand file required. Another drawback is that previous studies generally have not been facilitated by existing forensic tools, so further research is needed to see how far metadata might be useful to support the digital investigation process [25,26]. Checking the signature file automatically becomes a solution to overcome the deficiencies in previous research. One of the challenges that will be tried to be solved is related to identifying the authenticity of a data, namely checking or identification is carried out with the help of an application that is designed to read the signature of the input file. The signature file obtained will be compared with the data in the available signature file table to find out the extension of the file [27].

Therefore, in this research the signature file identification process is no longer done with conventional method by checking manually the file signature. In this research, identification of signature files will be carried out automatically through the developed application with signature databases are used to match these predefined patterns with the beginning or specific parts of files. The application is designed to be able to read the signature file from the input file. The signature file data reading results will be compared with the signature file data previously stored in the database to find out the extension of the file. Experiments were carried out on files with original extensions and files with modified extensions.

2. Methodology

There are four stages carried out in this research, namely: data collection, data analysis and system requirements, system design, implementation and testing can be seen in Figure 1.

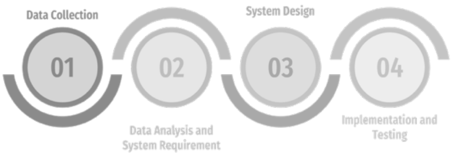

Data Collection
01
02
Data Analysis and System Requirement
System Design
03
04
Implementation and Testing

Fig. 1. Methodology

2.1 Data Collection

The type of data used in this research is primary data by observing and secondary data, namely data obtained from journal publication papers, documentation books, and the internet.

2.2 Data Analysis and System Requirement

The data that has been obtained is then analysed using descriptive analysis method. The method of descriptive analysis is carried out by describing facts which are then followed by analysis, not merely describing, but also providing sufficient understanding and explanation. Software requirements analysis is carried out to explore the needs of the software to be built.

2.3 System Architecture Design

The first process carried out by the program is the input file with the modified extension. The file that has been input will be read by the signature file and compared with the signature file data in the database. If the signature file is in the database, the output "Detected" will appear which contains the input file name, size, signature file, original extension, and time of uploading the file into the program. However, if the signature file is not in the database, a "Not Detected" output will appear where the signature file will be given a statement that it is not supported. Modified files whose signature files are contained in the database can be restored to the original extension by downloading the file in the original extension. The designed flow diagram system architecture is shown in Figure 2.

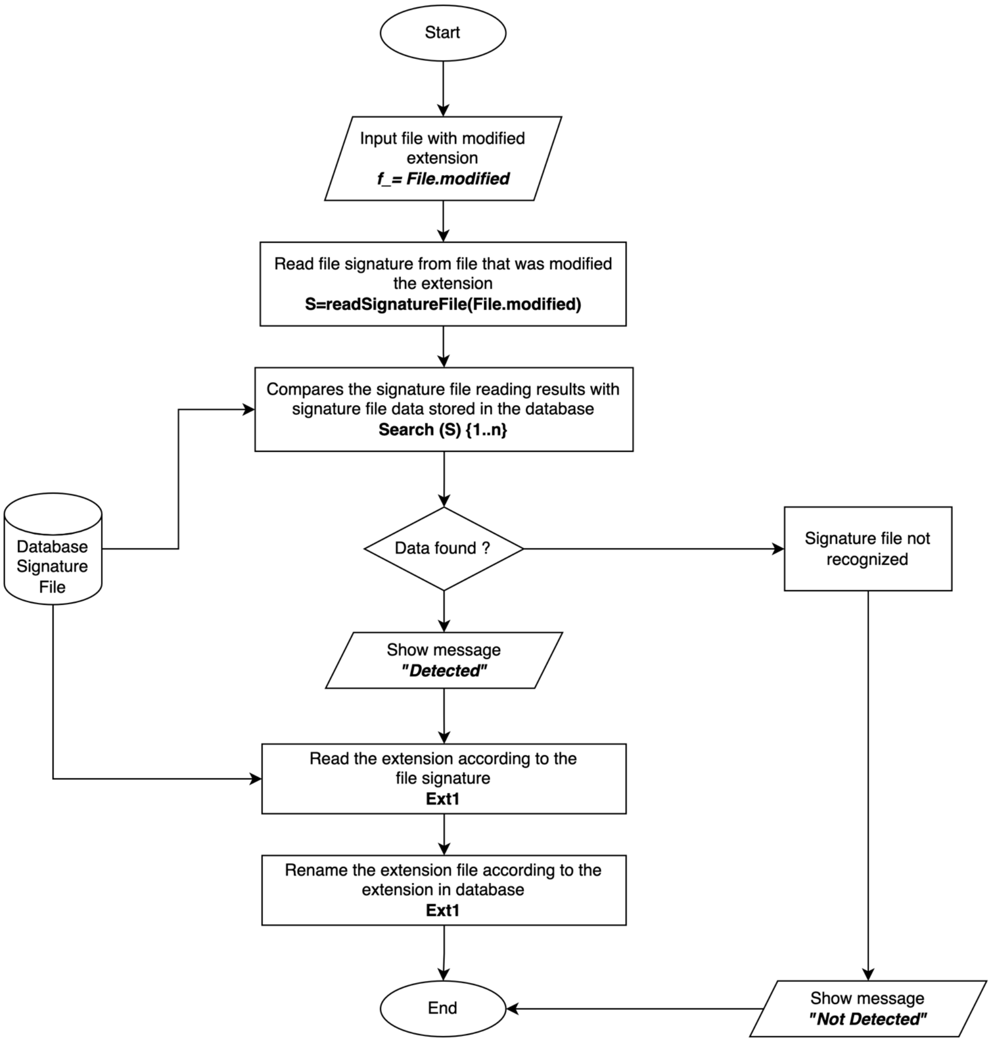

Start
Input file with modified extension
f_= File.modified
Read file signature from file that was modified the extension
S=readSignatureFile(File.modified)
Compares the signature file reading results with signature file data stored in the database
Search (S) {1..n}
Database Signature File
Data found ?
Signature file not recognized
Show message
"Detected"
Read the extension according to the file signature
Ext1
Rename the extension file according to the extension in database
Ext1
End
Show message
"Not Detected"

Fig. 2. Flow diagram system architecture design

The signature file matching algorithm in this research shown Figure 3 is often used in other applications such as plagiarism detection systems and in text processing that requires document matching, in this study it is used to identify the originality of data or file extensions. By using this algorithm, it can reduce the time needed to compare large documents and speed up the process of identifying and processing information. The Figure 3 describes from the first signature file input file, doing a string split and get total value, repetitions to the maximum limit, creates new variable that contains the value of the split signature file array, match the signature file reading results with the signature file data stored in the database and signature file recognized.

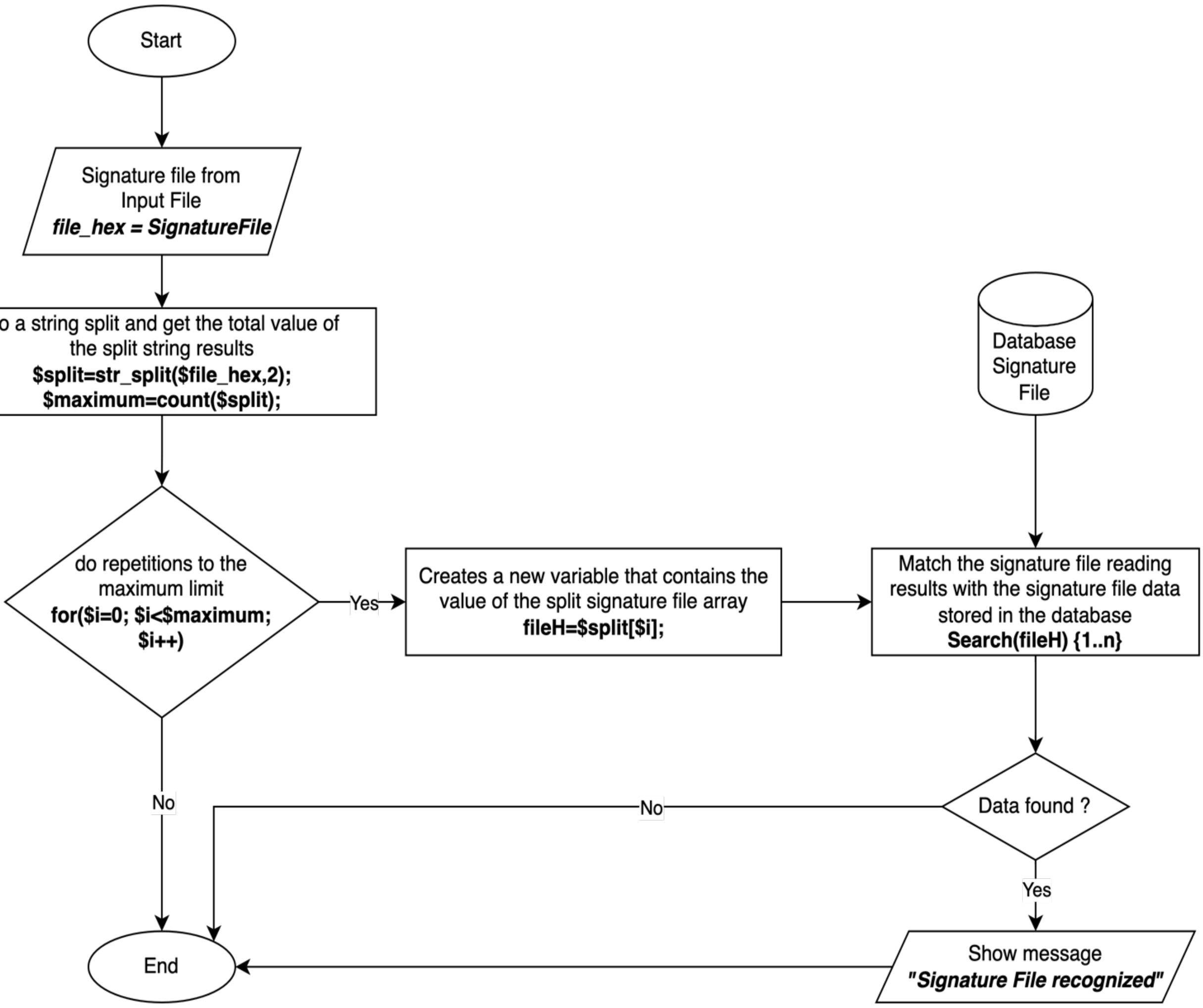

Start
Signature file from Input File
file_hex = SignatureFile
Do a string split and get the total value of the split string results
$split=str_split($file_hex,2);
$maximum=count($split);
do repetitions to the maximum limit
for($i=0; $i<$maximum; $i++)
Yes
Creates a new variable that contains the value of the split signature file array
fileH=$split[$i];
Database Signature File
Match the signature file reading results with the signature file data stored in the database
Search(fileH) {1..n}
Data found ?
No
No
Yes
Show message
"Signature File recognized"
End

Fig. 3. Algorithm matching signature File

2.4 Implementation and Testing

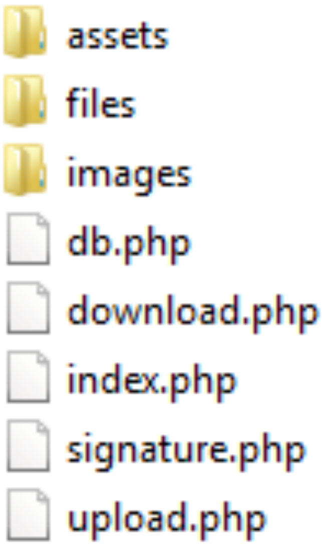
From the results of data analysis, software requirements, and design, the next stage is the process of implementing the software by creating a web-based application as forensic intelligence to identify signature-based file originality. System testing is carried out to match the signature files of each document with each other. Examination of matching results and identification of documents deemed similar or not original. Then analyse the results and check the accuracy in identifying documents that are not original.

3. Results

In this section, it is describing the research result obtained thought the stages in Figure 1. Explained of implementation proses of forensic signature file and testing with 31 sample file which be tested for every extension.

3.1 Process of Forensic Signature File

Process of implementation is done by preparing to build an application with the following file structure which can see in the Figure 4 and general description in Table 1.



assets
files
images
db.php
download.php
index.php
signature.php
upload.php

Fig. 4. File structure in signature file system

Table 1
General description of each directory and file in the system

No	File Name / Directory	Description
1.	css	Directory containing files for customizing the app's UI
2.	files	Directory for storing User uploaded files
3.	js	Directory containing files for customizing the app's UI
4.	scss	Directory containing files for customizing the app's UI
5.	vendor	Directory containing files for customizing the app's UI
6.	change.php	File to display Upload button to change Extension
7.	changef.php	The main file that has the function of changing the extension and storing data into the database
8.	check.php	The file displays the Upload Button to find out if the file has been modified or not
9.	db.php	The file that contains the database link function
10.	error.php	The file displays that the file has been modified so that it cannot be uploaded to the server
11.	index.php	In the initial appearance of the application, you can see the history of files that have been checked for file signatures

3.2 The Testing of Forensic Signature File

In this testing phase, 31 sample files have been prepared to be tested for each extension, 9 files with unmodified extensions as shown in Table 2 and 22 files with modified extensions as shown in Table 3.

Table 2
Scenario testing extension without modification

No	File Name	Initial extension	Modified extension
1.	file_example_AVI_480_750kB.avi	.avi	-
2.	file_example_favicon.ico	.ico	-
3.	file_example_GIF_500kB.gif	.gif	-
4.	file_example_JPG_500kB.jpg	.jpg	-
5.	file_example_MP3_1MG.mp3	.mp3	-
6.	file_example_PNG_1MB.png	.png	-
7.	file_example_TIFF_1MB.tiff	.tiff	-
8.	Firefox Setup 47.0.rar	.rar	-
9.	VENUS.docx	.docx	-

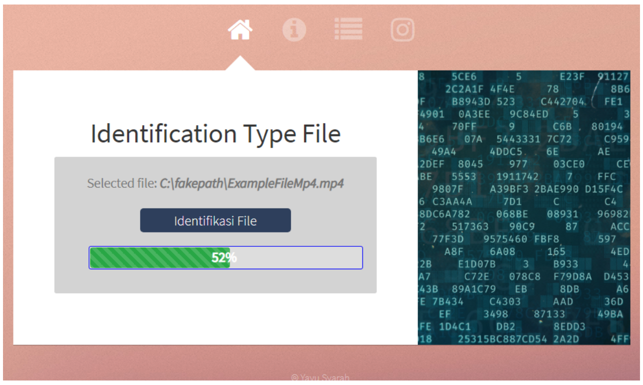
The first test was carried out on files with unmodified extensions, tests were carried out using 9 extensions that represent file types such as image, audio, video, compressed, icon, animation. This test is carried out to find out the output given by the application on each file contained in the database or not in the signature file database.

Table 3
Scenario testing extension without modification

No	File Name	Initial extension	Experimental result
1.	file_example_AVI_480_750kB.avi	.avi	Detected
2.	file_example_favicon.ico	.ico	Detected
3.	file_example_GIF_500kB.gif	.gif	Detected
4.	file_example_JPG_500kB.jpg	.jpg	Detected
5.	file_example_MP3_1MG.mp3	.mp3	Detected
6.	file_example_PNG_1MB.png	.png	Detected
7.	file_example_TIFF_1MB.tiff	.tiff	Detected
8.	Firefox Setup 47.0.rar	.rar	Detected
9.	VENUS.docx	.docx	Not Detected

In the experimental results of Table 3, there were 9 extensions that were tested without going through the modification process. After conducting the experiment, there were 8 extensions that were detected and 1 extension that was not detected. The extension is declared detected if the extension is contained in the signature file database.

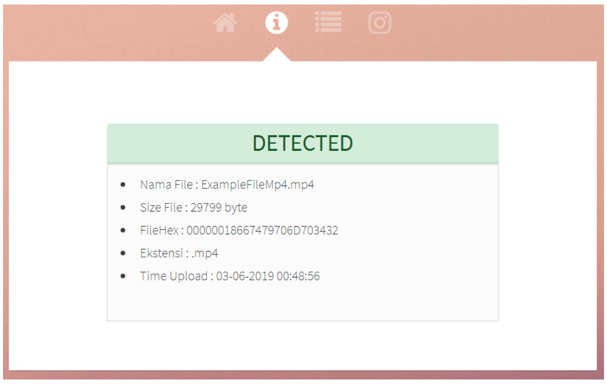
The following Figure 5 and Figure 6 below is an example of the display of the signature file identification process and the results show that the detected file will display information in the form of the file name, file size, signature file/hex file, extension, and upload time of the file to the application. The file Venus.docx not detected because the .docx extension is not available in the database so the signature file is not readable by the system.



Identification Type File

Selected file: *C:\fakepath\ExampleFileMp4.mp4*

Fig. 5. File identification process view



DETECTED

- Nama File : ExampleFileMp4.mp4
- Size File : 29799 byte
- FileHex : 00000018667479706D703432
- Ekstensi : .mp4
- Time Upload : 03-06-2019 00:48:56

Fig. 6. File detected view

The next stage of the experiment is to create a test scenario using the modified data file. As shown in Table 4 and Table 5 describes the data with file signature that was modified.

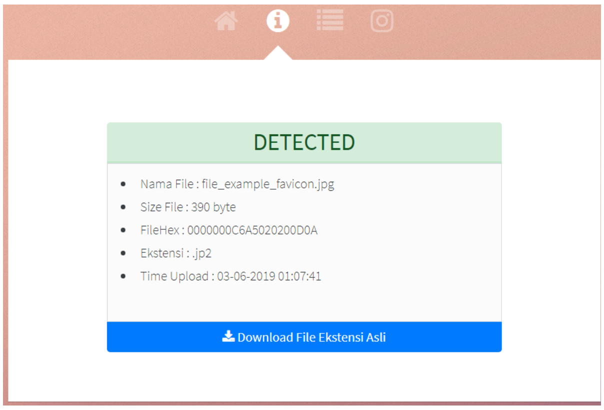
Table 4
Scenario testing extension with modification

No	File Name	Initial extension	Modified extension
1.	20051210-w50s.jpg	.flv	.jpg
2.	bell_206.zip	.ps	.zip
3.	Car-speakers-590x90.ico	.swf	.ico
4.	ExampleFileMp4.ps	.mp4	.ps
5.	exampleZip.rtf	.zip	.rtf
6.	file_example_AVI_480_750kB.bmp	.avi	.bmp
7.	file_example_favicon.jpg	.ico	.jpg
8.	file_example_GIF_500kB.hlp	.gif	.hlp
9.	file_example_JPG_500kB.rar	.jpg	.rar
10.	file_example_MP3_1MG.cab	.mp3	.cab
11.	file_example_PNG_1MB.jp2	.png	.jp2
12.	file_example_TIFF_1MB.gif	.tiff	.gif
13.	Firefox Setup 47.0.tiff	.rar	.tiff
14.	FULLXREF.zip	.HLP	.zip
15.	images.mp4	.psd	.mp4
16.	journeymans-song.flv	.mid	.flv
17.	relax.swf	.jp2	.swf
18.	sample.avi	.cab	.avi
19.	Sample.mp3	.rtf	.mp3
20.	sample.ico	.wmf	.ico
21.	VENUS.docx	.BMP	.docx
22.	Firefox Setup 47.0.iso	.exe	.iso

Table 5
Scenario testing extension with modification

No	File Name	Initial extension	Modified extension	Experimental result
1.	20051210-w50s.jpg	.flv	.jpg	The signature file is detected, and can be restored
2.	bell_206.zip	.ps	.zip	The signature file is detected, and can be restored
3.	Car-speakers-590x90.ico	.swf	.ico	The signature file is detected, and can be restored
4.	ExampleFileMp4.ps	.mp4	.ps	The signature file is detected, and can be restored
5.	exampleZip.rtf	.zip	.rtf	The signature file is detected, and can be restored
6.	file_example_AVI_480_750k.bmp	.avi	.bmp	The signature file is detected, and can be restored
7.	file_example_favicon.jpg	.ico	.jpg	The signature file is detected, and can be restored
8.	file_example_GIF_500kB.hlp	.gif	.hlp	The signature file is detected, and can be restored
9.	file_example_JPG_500kB.rar	.jpg	.rar	The signature file is detected, and can be restored
10.	file_example_MP3_1MG.cab	.mp3	.cab	The signature file is detected, and can be restored
11.	file_example_PNG_1MB.jp2	.png	.jp2	The signature file is detected, and can be restored
12.	file_example_TIFF_1MB.gif	.tiff	.gif	The signature file is detected, and can be restored
13.	Firefox Setup 47.0.tiff	.rar	.tiff	The signature file is detected, and can be restored
14.	FULLXREF.zip	.HLP	.zip	The signature file is detected, and can be restored
15.	images.mp4	.psd	.mp4	The signature file is detected, and can be restored
16.	journeymans-song.flv	.mid	.flv	The signature file is detected, and can be restored
17.	relax.swf	.jp2	.swf	The signature file is detected, and can be restored
18.	sample.avi	.cab	.avi	The signature file is detected, and can be restored
19.	Sample.mp3	.rtf	.mp3	The signature file is detected, and can be restored
20.	sample.ico	.wmf	.ico	The signature file is detected, and can be restored
21.	VENUS.docx	.BMP	.docx	The signature file is detected, and can be restored
22.	Firefox Setup 47.0.iso	.exe	.iso	The file is not detected, and cannot be restored

The file with the name file_example_favicon.jpg identification process is carried out as shown in Figure 7, the result states that the file has been modified. The output displayed is the name of the uploaded file, the size of the file, the file/hex file signature, the original extension, and the time the file was uploaded to the application. The identification results explained that the initial extension of the file was JP2. Because the JPG and JP2 extensions are in the database, these files can be detected and can be restored by clicking the "Download Original File Extension" button.

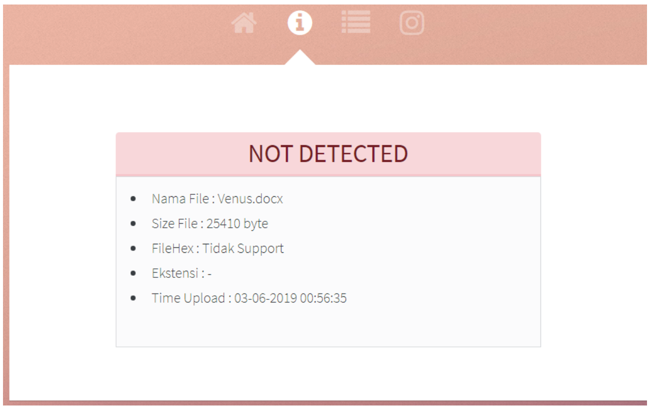


DETECTED

- Nama File : file_example_favicon.jpg
- Size File : 390 byte
- FileHex : 0000000C6A5020200D0A
- Ekstensi : .jp2
- Time Upload : 03-06-2019 01:07:41

Download File Ekstensi Asli

Fig. 7. View file detected in image



NOT DETECTED

- Nama File : Venus.docx
- Size File : 25410 byte
- FileHex : Tidak Support
- Ekstensi : -
- Time Upload : 03-06-2019 00:56:35

Fig. 8. View file not detected in difference ext.

The comparison table with previous research shows that this research has a clear contribution, for example reading signature files in previous research was done manually and research is now done automatically and is able to detect metadata and restore the original file extension.

Table 6
Comparison with previous research

		Scope of research									
		Preparation			Method	File Type					
No.	Author	Acquisition	Identification	Verification	Analysis of signature file	Analysis using forensic	Image	Video	Audio	Compressed	Document
1.	Moh. Subli *et al.,* [20]	No	Yes	No	No	Yes	Yes	Yes	Yes	No	Yes
2.	Aanahita *et al.,* [26]	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No
3.	D Hamdi *et al.,* [19]	No	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No
4.	Nur *et al.,* (proposed)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

4. Conclusions

Each file has its own signature file, making it easier for developers to carry out forensic investigations of a file that has been uploaded. The program created can detect file signature values from 21 extensions. This research conducted experiments on 33 files, there were 9 files with unmodified extensions, 22 files with modified extensions, 1 file with the extension removed and 1 file using more than one extension. The results of the experiment state that, for files whose extensions are not modified, the extension is declared valid if the extension is contained in the database. Files whose extensions are modified by changing the file extension can be concluded that the extension can be detected and restored if the initial extension and the modified extension are in the database. Finally, files with omitted extensions can still be detected if the file extension is in the database and for files that use more than one extension, it can be concluded that the extension to be used by the file is the most recent extension, if the final extension is not the same as the original extension, you can return as long as the modified extension and the original extension are present in the database. For the future work, there are still many file extensions that are still not supported. Thus, further research is expected that all types of file extensions can perform Signature File Forensic Process.

Acknowledgement

This research was supported by the Institute for Research, Community Services and Education Quality Assurance, Siliwangi University, and Ministry of Research, Technology and Higher Education of the Republic of Indonesia with decree number 1656/UN58/PP/2023.

References

[1] Brazevich, Dmitrii S., Zhanna S. Safronova, Tatyana N. Kosheleva, and Alla V. Biryukova. "Analysis of the Problems of Ensuring Information Security in the Terms of the Contemporary Society." *Open Journal of Social Sciences* 8, no. 2 (2020): 231-241. DOI: 10.4236/jss.2020.82018

[2] Ting, T. T., Z. H. Eu, S. B. Lim, and K. S. Chong. "Analysis of Information Security Awareness within Users' Preference, Practice and Knowledge."

[3] Razikin, Khairur, and Benfano Soewito. "Cybersecurity decision support model to designing information technology security system based on risk analysis and cybersecurity framework." *Egyptian Informatics Journal* 23, no. 3 (2022): 383-404. DOI: 10.1016/j.eij.2022.03.001

[4] Tariq, Usman, Irfan Ahmed, Ali Kashif Bashir, and Kamran Shaukat. "A critical cybersecurity analysis and future research directions for the internet of things: a comprehensive review." *Sensors* 23, no. 8 (2023): 4117. DOI: 10.3390/s23084117

[5] Jabłoński, Janusz, and Silva Robak. "Information systems development and usage with consideration of privacy and cyber security aspects." In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 1-8. IEEE, 2019. DOI: 10.15439/2019F261

[6] Hegarty, Rob, and John Haggerty. "SlackStick: Signature-based file identification for live digital forensics examinations." In *2015 European Intelligence and Security Informatics Conference*, pp. 24-29. IEEE, 2015. DOI: 10.1109/EISIC.2015.28

[7] Prakash, Vijay, Alex Williams, Lalit Garg, Claudio Savaglio, and Seema Bawa. "Cloud and edge computing-based computer forensics: Challenges and open problems." *Electronics* 10, no. 11 (2021): 1229. DOI: 10.3390/electronics10111229

[8] Dubettier, Adrien, Tanguy Gernot, Emmanuel Giguet, and Christophe Rosenberger. "File type identification tools for digital investigations." *Forensic Science International: Digital Investigation* 46 (2023): 301574. DOI: 10.1016/j.fsidi.2023.301574

[9] Alotaibi, May A., Mohammed A. AlZain, Ben Soh, Mehedi Masud, and Jehad Al-Amri. "Computer forensics: dark net forensic framework and tools used for digital evidence detection." *International Journal of Communication Networks and Information Security* 11, no. 3 (2019): 424-431. DOI: 10.17762/ijcnis.v11i3.4407

[10] Javed, Abdul Rehman, Waqas Ahmed, Mamoun Alazab, Zunera Jalil, Kashif Kifayat, and Thippa Reddy Gadekallu. "A comprehensive survey on computer forensics: State-of-the-art, tools, techniques, challenges, and future directions." *IEEE Access* 10 (2022): 11065-11089. DOI: 10.1109/ACCESS.2022.3142508

[11] Otieno, George Raburu, and Lawrence Dinga. "Legal issues in computer forensics and digital evidence admissibility." (2020).

[12] Villar-Vega, H. F., L. F. Perez-Lopez, and J. Moreno-Sanchez. "Computer forensic analysis protocols review focused on digital evidence recovery in hard disks devices." In *Journal of Physics: Conference Series*, vol. 1418, no. 1, p. 012008. IOP Publishing, 2019. DOI: 10.1088/1742-6596/1418/1/012008

[13] Pedapudi, Srinivasa Murthy, and Nagalakshmi Vadlamani. "Digital forensics approach for handling audio and video files." *Measurement: Sensors* 29 (2023): 100860. DOI: 10.1016/j.measen.2023.100860

[14] F. Morgado, 'Brief Introduction to Word and File Extensions', in *Microsoft Word Secrets*, Berkeley, CA: Apress, 2017, pp. 1–5. DOI: 10.1007/978-1-4842-3078-7_1

[15] W. Pranoto, I. RIadi, and Y. Prayudi, 'Live Forensics Method for Acquisition on the Solid State Drive (SSD) NVMe TRIM Function', *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 129–138, May 2020. DOI: 10.22219/kinetik.v5i2.1032

[16] K. Karampidis and G. Papadourakis, ‘File Type Identification - Computational Intelligence for Digital Forensics’, *The Journal of Digital Forensics, Security and Law*, 2017. DOI: 10.15394/jdfsl.2017.1472

[17] Bill Nelson, Amelia Phillips, and Chris Steuart, *Information Security: Guide To Computer Forensics and Investigations*, Sixth. Boston, USA, 2019.

[18] A. Farjamfar, Mohd. T. Abdullah, R. Mahmod, and N. I. Udzir, ‘Multimedia Files Signature Analysis in Blackberry Z10’, *Journal of Applied Sciences*, vol. 15, no. 4, pp. 668–674, Mar. 2015. DOI: 10.3923/jas.2015.668.674

[19] D. Hamdi, F. Iqbal, T. Baker, and B. Shah, ‘Multimedia File Signature Analysis for Smartphone Forensics’, in *2016 9th International Conference on Developments in eSystems Engineering (DeSE)*, 2016, pp. 130–137. DOI: 10.1109/DeSE.2016.22

[20] Moh Subli, Bambang Sugiantoro, and Yudi Prayudi, ‘Metadata Forensik Untuk Mendukung Proses Investigasi Digital’, *Jurnal Data Manajemen dan Teknologi Informasi*, vol. 18, no. 1, pp. 44–50, 2017.

[21] R. Rizal, R. Ruuhwan, and S. Chandra, ‘Signature File Analysis Using The National Institute Standard Technology Method Base on Digital Forensic Concepts’, *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 3, p. 364, Sep. 2020. DOI: 10.32493/informatika.v5i3.6073

[22] H. Kaur and M. Kumar, ‘Signature identification and verification techniques: state-of-the-art work’, *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 2, pp. 1027–1045, Feb. 2023. DOI: 10.1007/s12652-021-03356-w

[23] A. Chevalier *et al.*, ‘The Mutational Signature Comprehensive Analysis Toolkit (musicatk) for the Discovery, Prediction, and Exploration of Mutational Signatures’, *Cancer Research*, vol. 81, no. 23, pp. 5813–5817, Dec. 2021. DOI: 10.1158/0008-5472.CAN-21-0899

[24] A. Afrizal, N. Dwi, W. Cahyani, and E. Jadied, ‘Analysis and Implementation of Signature Based Method and Structure File Based Method for File Carving’, *Indonesia Journal of Computing (Indo-JC)*, vol. 6, no. 1, pp. 13–22, 2021. DOI: 10.34818/indojc.2021.6.1.457

[25] S. Sourabh, D. Chauhan, V. Singh, and M. Chauhan, ‘Analysis of Digital Data by File Signature Method on Android Version 9’, *Recent Advances in Computer Science and Communications*, vol. 15, no. 8, Jul. 2022. DOI: 10.2174/2666255813666201216114643

[26] A. Farjamfar, Mohd. T. Abdullah, R. Mahmod, and N. I. Udzir, ‘Multimedia Files Signature Analysis in Blackberry Z10’, *Journal of Applied Sciences*, vol. 15, no. 4, pp. 668–674, Mar. 2015. DOI: 10.3923/jas.2015.668.674

[27] J. Sammons and M. Cross, ‘Software problems and solutions’, in *The Basics of Cyber Safety*, Elsevier, 2017, pp. 53–74. DOI: 10.1016/B978-0-12-416650-9.00003-6