A Low-Cost Disk Imaging Device Using Raspberry Pi 4 for Digital Forensics

Catur Adi Nugroho and Amiruddin Amiruddin

August 28, 2023

A Low-Cost Disk Imaging Device Using Raspberry Pi 4 for Digital Forensics

1st Catur Adi Nugroho
Rekayasa Keamanan Siber
Politeknik Siber dan Sandi Negara
Bogor, Indonesia
caturadi9@gmail.com

2nd Amiruddin Amiruddin
Rekayasa Keamanan Siber
Politeknik Siber dan Sandi Negara
Bogor, Indonesia
amir@poltekssn.ac.id

Abstract— As with crimes that occur in the real world, cybercrime cases require investigation to gather evidence for court proceedings. Digital forensics is a great way to investigate evidence from cybercrime. Investigating a cybercrime to find digital evidence cannot be done inappropriately. Mishandling of evidence can make it invalid. It is necessary to copy the evidence before, so the investigation process is not done directly on the original digital evidence. Intended to avoid or at least minimize damage to evidence. In conducting an investigation, an investigator needs tools and supporting devices that have high accuracy, reliable, and affordable. However, most of the devices on the market are available at quite expensive prices. Therefore, in this study, performed design and construction of a low-cost disk imaging device were carried out using the Raspberry Pi 4 and implemented using Python languages to copy evidence in the digital forensics process. Testing results and analysis of the proposed device show a good performance in terms of speed and accuracy of copies.

Keywords—Digital Forensics, Disk Imaging, Raspberry Pi 4, Low Cost

I. PRELIMINARY

The development of information and communication technology (ICT) brings many benefits to human life in various sectors such as work, learning, business, and many others. ICTs have not only made human work easy, but now they have become the core of the work itself [1]. The documents that used to be stored physically can be stored digitally using ICT devices. However, the development of ICT also can be misused by some parties to commit ICT-related crimes called cybercrime. Cybercrime is an illegal act that targets digital devices or information systems [2]. One example of a cybercrime case that has occurred in Indonesia is the WannaCry ransomware case that occurred in 2017. This case affected several institutions like banks, telecommunications, transportation, and health services. The impact of this crime is the tension of the work process. Thus, cybercrime needs to be a concern for cybersecurity teams to prepare appropriate steps to handle cases in the future [3].

As has happened with crimes in the real world, cybercrime cases are through a judicial process. In the judicial process against cybercrime, besides requiring a firm legal basis (such as laws and regulations), a verification mechanism is also needed through digital forensics. One branch of digital forensics is computer forensics which allows investigators to gather as much information about computer users as possible, find deleted files, reconstruct artifacts, and collect digital evidence [4]. The investigator's task is to recover and analyze information from the victim's computer as evidence of a crime.

The digital evidence acquisition and analysis of cybercrime cases require special treatment because even if a small error happens in the handling of digital evidence can injure evidence and consequently cannot be used as legal evidence in the judicial process [5]. One of the efforts to safeguard evidence is copying data from storage media in a bitstream image and placing it in a safe place. Bitstream is a digital storage method copying every bit of the original data, including hidden files, temp files, file fragments, and files that have not been overwritten [6]. The technique of retrieving bit by bit of data from physical storage media is commonly known as disk imaging [7].

An investigator needs relevant and reliable hardware and software for the disk imaging process. Now many types of disk imaging applications are available in the market, both open-source and paid. Both types of applications certainly have advantages and disadvantages of each. Features of paid apps are tested with standards to ensure their quality, but paid apps require users to purchase or subscribe to the app. While open-source applications have the advantage of being used for free, there is a possibility that they have flaws in the application that damage digital evidence.

In conducting disk imaging, an investigator needs an easy-to-use application. To make easy of use the application, disk imaging devices are built using a graphical user interface (GUI). Python is known for its high-level programming language, which makes it easy to create GUI-based programs [8]. One of the Python modules for creating a GUI is Tkinter which allows users to create simple GUI-based programs with just four lines of code [8]. Python with Tkinter module is an option in building GUI-based disk imaging applications so investigators will be easier to operate the device.

Disk imaging device designed and built using Raspberry Pi 4 as a single-board computer. The devices operated with a GUI-based disk imaging application built using the Python programming language with the Tkinter module. The final stage of this research is testing using several types of tests. The final result of this research is a tested disk imaging device using a Raspberry Pi 4 operated with a GUI-based disk imaging application.

II. BASIC THEORY

A. Digital Forensics

According to Kapoor's research [9], digital forensics is a branch of computer science focused on the identification, recovery, investigation, validation, and complications of information collected by research work carried out to collect evidence of crimes committed. The purpose of digital forensics is to find digital evidence of an incident. Forensic investigations use digital and physical evidence with scientific procedures and approaches to find conclusions [10]. From the definition, knowing that the key to digital forensics is evidence, collected information from digital devices such as CDs, DVDs, flash drives, floppy disks, memory cards, cell phones, and RAM [10].

B. Disc Imaging

It is a part of computer forensics to retrieve data bit by bit data from physical storage media, including hidden files, temp files, file fragments, and files that have not been overwritten [6]. After that, it is also necessary to carry out the preservation stage of the copy of evidence using MD5 or SHA1 included on the label of each file to distinguish it from the original data to ensure data integrity [11]. The purpose of this disk imaging technique is to copy evidence data as material for analysis by investigators. Disk imaging tool requires several specifications, including accuracy of duplicate creation, source disk integrity, verifying image file integrity, and error logging.

C. Raspberry Pi

The Raspberry Pi is a portable, low-power PC device developed by the Raspberry Pi Foundation. Prices of the Raspberry Pi range from $5 to $35. The Raspberry Pi board contains a Broadcom-based ARM Processor, Graphics Chip, RAM, GPIO, and other connectors for external devices. The operation of the Raspberry Pi requires additional hardware such as a keyboard, mouse, display assembly, power supply, and an SD card containing the operating system that acts as a hard disk for operation. Raspberry Pi operated with an open-source operating system based on Linux. In support of its performance, the Raspberry Pi has also launched various accessories such as cameras, Gertboards, and several other modules [12].

D. Hash

A hash is a function in computer science to perform compression of an input string into a fixed length. This function includes authentication purposes, digital signatures, pseudo-number generation, steganography, and digital timestamps [13]. The hashing provides security for achieving the integrity of the data.

There are various types of hashes based on the design of the generation. The popular design types are the Message Digest (MDx) family and the Secure Hash Algorithm (SHA) family. The MDx family in question consists of MD2, MD4, and MD5. MD2 is slower than the other two MDx, while MD4 has received the most attention in practice. Due to security concerns, a new version was launched called MD5 to replace the previous MD4 hash and became a milestone in hash development [13].

SHA was designed on the same principles as MD4 in 1993 by the National Institute of Standards and Technology (NIST). In 2002, NIST released a revised version of the standard FIPS180-2 and defined three new versions of SHA with lengths 256, 384, and 512. In October 2008, FIPS 180-2 replaced FIPS 180-3. All versions of SHA are based on the same MD4 principles but updated every version[13]

III. Research Method

The method used in disk imaging device development is the System Development Life Cycle (SDLC) methodology with the Waterfall approach. The Waterfall Approach was selected because it is appropriate when the requirements can define in the planning section. This method resembles the shape of a waterfall in which each process flows without returning to the previous process [14]. The waterfall approach includes planning, analysis, design, and implementation stages. The Waterfall Approach consists of planning, analysis, design, and implementation stages.

A. Planning

In this stage, we defined the overview of the application. We used the specifications of the disk imaging device stated in the research [15] because the proposed device will have the main functions, name, bitstream copy, logging, and hashing. Bitstream copy is the core function of the Disk Imaging Application for copying all files from the flash drive, for all ordinary files, hidden files, temp files, file fragments, and files that have not been overwritten [6]. The logging function plays a role in recording the imaging process. The hashing generates the hash value of the resulting image to validate.

B. Analysis

This stage determined the requirements and specifications of the application to be built based on the planning stage. The application requirements break into Functional and non-functional. The functional requirements based on research [15] comprise bitstream copy, logging, and hashing functions. There is also a disk read additional function to find out connected devices. The non-functional requirements of disk imaging applications based on research [15] include the use of a programming language, namely Python, and the determination of the use of hash algorithms, namely MD5 and SHA256. In addition, this study also used the Raspberry Pi 4 device at the implementation stage.

C. Design

This stage used the Unified Modeling Language (UML) modeling to show the application as a whole. UML design includes use case diagrams, activity diagrams, and sequence diagrams. The function contained in the disk imaging application illustrated a Use case diagram. The function workflow mentioned in the use case diagram depicted the activity diagram. The Sequence diagram describes The workflow for each feature detail and the method used for each function.

The description of the disk imaging application starts by reading the connected flash drive device to the disk imaging device and continues to select the disk. After selecting the correct device, the user enters some inputs. After inputting correctly, the process will generate the source hash value and continue with the bitstream copy process. Next, the disk imaging device generated the image hash output with logs of the activities performed.

D. Implementation

This stage begins with system development based on the prepared model. The system is a GUI-based disk imaging application as an operational medium for disk imaging devices. The next application runs on a Raspberry Pi 4 Model

B with 2GB of RAM with the Raspbian operating system. The Raspberry Pi 4 device is supported by additional devices, namely a 3.5-inch screen, keyboard, and mouse. This stage gives a breakdown of the cost of building a disk imaging device. In the end, Applications go through the testing phase using several testing methods to determine the suitability of the resulting application with the design. Tests carried out include unit testing, integration testing, and system testing.

a. Unit tests ensure that each class in a disk imaging application conforms to the required system specifications. There are two types of unit testing called black-box testing and white-box testing [14]. The type of unit testing approach in this study is white-box testing.

b. Integration tests ensure that the integration of the use case diagram can work and produce the appropriate value. Integration tests consist of user interface, use case, interaction, and system interface tests [14]. The type of integration testing approach used in this study is user interface testing.

c. System testing ensures that changes from integration testing do not cause errors and are by the previously planned overview. There are five types of approaches to system testing, namely requirements testing, usability testing, documentation testing, performance testing, and security testing [14]. The system testing approach used in this research is requirement testing.

d. Performance tests ensure the reliability of the device in the imaging process. The variables used to measure device performance are speed and hash value. The device speed measurement refers to the Mali study [16]. The hash value of the source disk and the resulting image are compared to ensure the device copying process runs well and does not damage the disk source. These are the requirements for developing imaging devices stated in the CFTT standard, i.e., accurate duplicating, unaltered source disks, and verifying the integrity of image files.

IV. Results and Discussion

A. Requirements

The proposed disk imaging application builds on research conducted by Lyle [15]. The required specifications of an imaging application have specifications for the accuracy of duplicating, source disk does not change, verifying the integrity of image files, and logging.

TABLE I. FUNCTIONAL REQUIREMENTS

No	Functional Requirement
1	The disk imaging application has a bitstream copy function to copy evidence.
2	The disk imaging application has a logging function to log imaging activity.
3	The disk imaging application has a hashing function for image validation needs.
4	Disk imaging application has the function to read the flash drive connected to the device.

TABLE II. NON-FUNCTIONAL REQUIREMENTS

No	Non-Functional Requirements
1	The disk imaging application is built using Python with the Tkinter module.
2	The disk imaging application is built using Raspberry Pi 4 as the implementation environment.
3	The disk imaging application is built using MD5 and SHA512 algorithms as hashing algorithms.

Based on the functional and non-functional requirements obtained from the research of Lyle [15] and Mali [16] shown in TABLE I and TABLE II, the designed disk imaging application has three main features called bitstream copy, logging, and hashing. The application also has a supported feature, namely a grep disk display list of disks connected to the disk imaging device by retrieving information from the device's operating system. The image file is stored on another storage medium selected by the user with the resulting log file. Hash cryptographic algorithms used in this application, namely the MD5 and SHA512 algorithms used to ensure that the resulting image is identical to the original disk.

The application design model through three types of diagrams, called use case diagrams, activity diagrams, and sequence diagrams. This design is made based on the functional and non-functional requirements of the application as well as an overview of the applications made previously.

B. Use Case Diagrams

The disk imaging applications have one actor as the application user. Fig. 1 shows a use case diagram of a disk imaging application. The image shows that the user must first ensure that the device used as the imaging source is installed and correctly selected. Once a disc is selected, it continues to the imaging process. A list update process can occur when the application does not detect the desired disk.

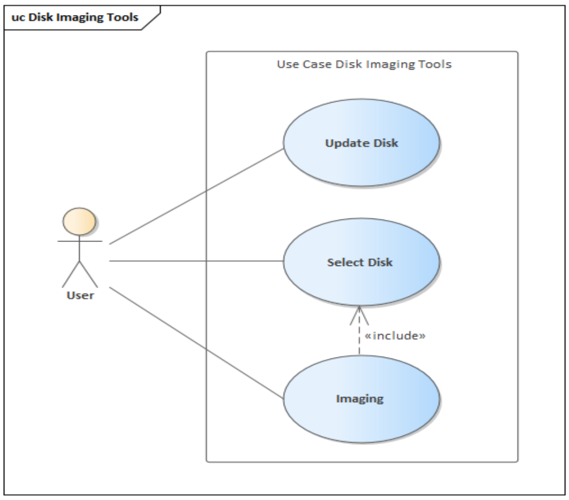


Fig. 1. Use case diagram

The image and log files are stored in the directory and file name according to user input. To use the disk imaging application again, the user can select the back button so that the application will delete the entry and return to the disk selection page.

C. Implementation

This section contains the specifications of the devices used in the implementation phase, both hardware and software. The specifications of the used equipment are presented in TABLE III. Before implementing the application, install the operating system on the Raspberry Pi device. In addition, the installation and configuration of additional devices such as LCD, keyboard, and mouse also need to be done. Fig. 2 shows the condition of the hardware used at the implementation stage. TABLE IV details the costs involved in the manufacture of disk imaging devices.

TABLE III. IMPLEMENTATION DEVICE SPECIFICATION

Device Name	Description
Hardware	
Raspberry Pi 4	Raspberry Pi 4 Type B RAM 2GB
LCD	3.5 Inch Raspberry Pi LCD
Keyboard	keyboard robot
Mouse	Robot Mouse
SD Card 16GB	SanDisk Ultra 16GB
Flash disk	SanDisk Cruzer Blade 16GB, Toshiba 8GB V-Gen 8GB, SanDisk Cruzer Blade 8GB
Software	
Operating system	Raspbian
Application Development	PyCharm 2021.1 (Community Edition)
Python Library	tkinter: GUI libraries hashlib: Library Hash logging: Library Log datetime: Time library OS: Libraries controlling the operating system multiprocessing: Library breaks parallel running processes

TABLE IV. DETAILS OF DEVICE DEVELOPMENT COST

Device Name	Price
Raspberry Pi 4	IDR 506.800
Mouse and Keyboard	IDR 154.000
Adapter	IDR 59.000
LCD	IDR 189.000
SD Card	IDR 56.000

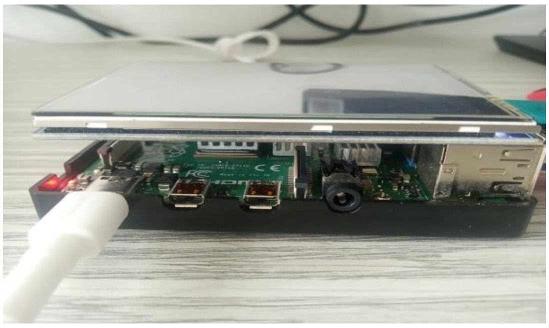
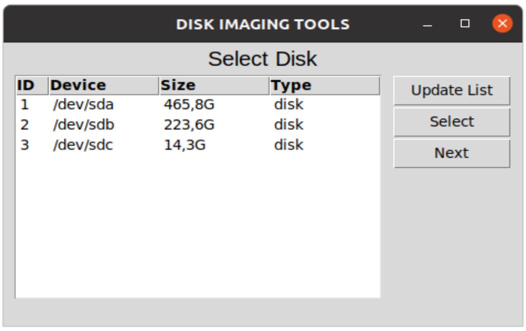


Fig. 2. Implementation Tool

It is known that the total cost required to manufacture disk imaging devices in this study is IDR 964.000,00. Based on his research [16] the price for constructing the disk imaging device using a Raspberry Pi 3 and a complete screen and protective case was IDR 838.336,18 in 2018. This shows a cost difference of IDR 125.663,82, where the device built in this study has a higher total cost. The difference condition is because the type of device used is a newer version of the Raspberry Pi, and there are several additional components, such as a keyboard and mouse.

The first step in running the disk imaging application is to connect the device to a power source. After the device is powered on, the disk list column, update list, select, and following buttons will appear, as shown in Fig. 3. This page has provided a list of disks connected to the disk imaging device. However, if the disk to be imaged is connected after the application is running, the user can select the update list button to display a list of the latest disks.

After the desired disk appears on the list, the user can select the disk by pressing the disk name and the select button. After the disk is selected, the application will save the selection and display the name of the chosen disk on the entry device on the imaging page. The user presses the next button in the following step, and the application will open the imaging page.

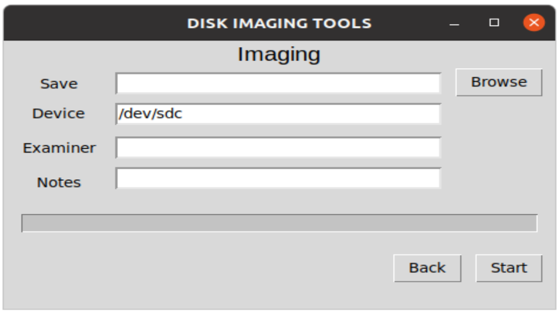


ID	Device	Size	Type
1	/dev/sda	465,8G	disk
2	/dev/sdb	223,6G	disk
3	/dev/sdc	14,3G	disk

Fig. 3. Disc Select Page

When the user has not selected a disk but pressed the following button, an error notification will appear, directing the user to select the disk first. The imaging process is performed after the selected disk choose through the disk selection process. The application displays an imaging page consisting of save entry, device, examiner, and note, as shown in Fig. 4.

After entries are filled, the user can start the imaging process by pressing the start button. After pressing the start button, a notification will appear that "the process cannot be interrupted" and if the user wants to stop the process, it can be done through the terminal by pressing “Ctrl + C” on the keyboard. As shown in Fig. 5, when the process is running the progress bar will show the progress of the imaging process.

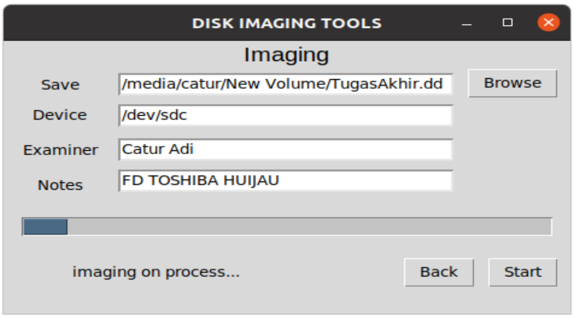


Save
Device /dev/sdc
Examiner
Notes

Fig. 4. Imaging Page

After the imaging process, a notification will be displayed to the user. If the user is going to perform the imaging process again, the user can press the back button, and the application will delete all filled entries and display the disk selection page again.

When an error is found because of the total capacity of storage or disk condition detached from the disk imaging device, the process will stop, and the application will display an error notification.

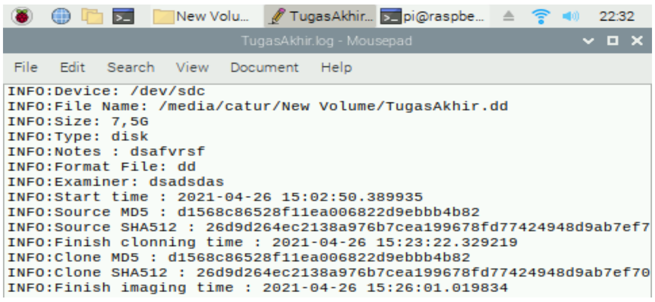


Save: /media/catur/New Volume/TugasAkhir.dd
Device: /dev/sdc
Examiner: Catur Adi
Notes: FD TOSHIBA HUIJAU

imaging on process...

Fig. 5. Progress bar Imaging

The imaging process results are an image file and a log file stored in the selected directory by the entered file name. The contents of the log file are shown in Fig. 6.



```
INFO:Device: /dev/sdc
INFO:File Name: /media/catur/New Volume/TugasAkhir.dd
INFO:Size: 7,5G
INFO:Type: disk
INFO:Notes : dsafvrsf
INFO:Format File: dd
INFO:Examiner: dsadsdas
INFO:Start time : 2021-04-26 15:02:50.389935
INFO:Source MD5 : d1568c86528f11ea006822d9ebbb4b82
INFO:Source SHA512 : 26d9d264ec2138a976b7cea199678fd77424948d9ab7ef7
INFO:Finish clonning time : 2021-04-26 15:23:22.329219
INFO:Clone MD5 : d1568c86528f11ea006822d9ebbb4b82
INFO:Clone SHA512 : 26d9d264ec2138a976b7cea199678fd77424948d9ab7ef70
INFO:Finish imaging time : 2021-04-26 15:26:01.019834
```

Fig. 6. Log File Contents

D. Testing and Analysis

Unit testing was conducted using white-box testing on two program classes with 13 functions. Based on the test results, the inputs and outputs tested for each part showed the expected results. Therefore, it can be stated that all the units in the application have been running correctly and in accordance. Integration testing uses the user interface testing method, provided that the test emphasizes the conformity of the interface on each page in the application. Based on the tests conducted, it is known that all the tested interfaces show outputs that are in line with expectations. Therefore, all interfaces and relationships between units in the application are appropriate, and the application can run according to the expected conditions.

The system testing used requirements testing because the test aims to see whether the application needs have been met. The test results showed that the application successfully fulfilled the three functional and non-functional requirements previously determined at the design stage. Therefore, the functional and non-functional requirements in the application are the results of the requirements analysis. In addition, the application complies with the CFTT standard because it provides a logging feature. Performance testing refers to the testing conducted in [16]. A device with a size of 8 GB was used in this test. This test aims to determine the average speed of the resulting disk-imaging machine and ensure that the imaging process does not change the contents of the disk source, as described by the CFTT standard.

a. Speed Test

The first variable tested is device speed. In this test, three flash disks with a storage capacity of 8 GB were used under the Toshiba, SanDisk, and V-Gen brands. The resulting image is stored on a SanDisk flash disk with a size of 16 GB. Speed testing is done by running a disk imaging application on the prepared device. Data retrieval in this test was carried out three times on each disk brand. TABLE V shows the results of the disk imaging device speed test. The speed value is obtained from the quotient between the actual size of the disk and the time duration generated by the device in seconds.

TABLE V. SPEED TEST RESULT

Device Name	Actual Capa-city	Imaging Duration 1	Imaging Duration 2	Imaging Duration 3	Speed
Flash disk Toshiba 8GB	7,577 MB	68 minutes	69 minutes	69 minutes	1.84 MB/s
Flash disk V-Gen 8GB	7,802 MB	67 minutes	66 minutes	67 minutes	1.95 MB/s
Flash disk SanDisk 8GB	7,616 MB	65 minutes	66 minutes	65 minutes	1.94 MB/s

b. Hash Value Test

The second variable is the hash value. In this test, a comparison of the hash value on the source disk and image results was performed. Based on the results of the comparison of hash values between the source disk and the resulting image, it is known that all hash values match. Thus, the disk imaging device has met the requirements for imaging application development: accuracy of duplicate creation, source disk is unchanged, and verifying image file integrity [15].

V. CONCLUSION

The design and development of disk imaging applications are based on the Computer Forensic Tool Testing (CFTT) project developed by the National Institute of Standards and Technology (NIST), which explains that a disk imaging application must meet several specifications, namely accuracy of duplicate creation, unchanged source disk, verified image file integrity, and logging. The features in disk imaging applications consist of the main features, namely bitstream copy, logging, and hashing, as well as supporting elements, namely, a grep disk. The application was built with four functional requirements and three nonfunctional requirements. There are three types of diagrams for designing disk imaging applications: two use-case diagrams, three activity diagrams, and three sequence diagrams.

There are four types of tests carried out in this research, unit testing, which is carried out in two classes, integration testing, which is carried out on two interfaces with the user interface testing method, system testing using the requirements testing method, which is carried out on four functional requirements and three non-functional requirements, and performance testing to measure the suitability of the hash value and speed of the disk imaging device. Unit, integration, and system testing results showed the usefulness of all tested variables. The performance testing results show that the disk imaging device has an average speed of 1.91 MB/s, and all the hash values generated are by the expected output.

REFERENCES

[1] M. Pomffyová and L. Bartková, "Take Advantage of Information Systems to Increase Competitiveness in SMEs," *Procedia Soc Behav Sci*, vol. 220, pp. 346–354, 2016, doi: 10.1016/j.sbspro.2016.05.508.

[2] R. Sabillon, J. Cano, V. Cavaller Reyes, J. Serra Ruiz, V. Cavaller, and J. Serra, "Cybercrime and cybercriminals: A comprehensive study," *International Journal of Computer Networks and Communications Security*, vol. 4, no. June, pp. 165–176, 2016, [Online]. Available: www.ijcncs.org

[3] L. Kertopati, "Dua Rumah Sakit di Jakarta Kena Serangan Ransomware WannaCry," *CNN Indonesia*, 2017. https://www.cnnindonesia.com/teknologi/20170513191519-192-214642/dua-rumah-sakit-di-jakarta-kena-serangan-ransomware-wannacry

[4] M. Kaur, N. Kaur, and S. Khurana, "A Literature review on Cyber Forensic and its Analysis tools," *Ijarcce*, vol. 5, no. 1, pp. 23–28, 2016, doi: 10.17148/ijarcce.2016.5106.

[5] R. Altschaffel, K. Lamshöft, S. Kiltz, and J. Dittmann, "A Survey on Open Automotive Forensics," in *International Conference on Emerging Security Information, Systems and Technologies*, 2017, pp. 65–70.

[6] E. Akbal and S. Dogan, "Forensics Image Acquisition Process of Digital Evidence," *International Journal of Computer Network and Information Security*, vol. 10, no. 5, pp. 1–8, 2018, doi: 10.5815/ijcnis.2018.05.01.

[7] M. Saudi, "An Overview of Disk Imaging Tool in Computer Forensics," 2020 [Online]. Available: https://www.sans.org/reading-room/whitepapers/authentication/overview-authentication-methods-protocols-118

[8] P. Podrzaj, "A brief demonstration of some Python GUI libraries," 2019, pp. 1–6.

[9] V. Kapoor, S. Taneja, and K. A. Kumar, "Digital Forensics Tools," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 2, pp. 3065–3969, 2019, doi: 10.35940/ijeat.B3980.129219.

[10] S. Yadav, K. Ahmad, and J. Shekhar, "Analysis of digital forensic tools and investigation process," *Communications in Computer and Information Science*, vol. 169 CCIS, pp. 435–441, 2011, doi: 10.1007/978-3-642-22577-2_59.

[11] R. F. M. Román, N. M. L. Mora, J. P. N. Vicuña, and J. I. P. Orozco, "Digital forensics tools," *International Journal of Applied Engineering Research*, vol. 11, no. 19, pp. 9754–9762, 2016, doi: 10.35940/ijeat.b3980.129219.

[12] A. Nayyar and V. Puri, "Raspberry Pi-A Small , Powerful , Cost Effective and Efficient Form Factor Computer : A Review International Journal of Advanced Research in Raspberry Pi- A Small , Powerful , Cost Effective and Efficient Form Factor Computer : A Review," no. December, 2015, [Online]. Available: https://www.researchgate.net/profile/Anand_Nayyar/publication/305668622_Raspberry_Pi-A_Small_Powerful_Cost_Effective_and_Efficient_Form_Factor_Computer_A_Review/links/5798c41908aeb0ffcd08b80f/Raspberry-Pi-A-Small-Powerful-Cost-Effective-and-Efficient-Form

[13] R. Sobti and G. Geetha, "Cryptographic Hash functions - a review," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 461–479, 2012.

[14] A. Dennis, B. H. Wixom, and R. M. Roth, *Systems Analysis and Design with UML 5th Eedition*, 5th ed. John Wiley & Sons, Inc., 2015.

[15] J. R. Lyle, "NIST CFTT: Testing disk imaging tools," in *Proceedings of the Digital Forensic Research Conference, DFRWS 2002 USA*, 2002, pp. 1–10.

[16] P. Mali, "Low Cost And Ultra Low Cost Digital Forensic," vol. 4, no. 1, pp. 156–160, 2018.