Design of Decentralized Application for Telemedicine Image Record System with Smart Contract on Ethereum

Darrell Yonathan[1], Diyanatul Husna[2], Fransiskus Astha Ekadiyanto[3], Anak Agung Putri Ratna[10]
Computer Engineering Department of Electrical Engineering, University of Indonesia, Depok, Indonesia

I Ketut Eddy Purnama[4], Mauridhi Hery Purnomo[6], Supeno Mardi Susiki Nugroho[7], Reza Fuad Rachmadi[8]
Faculty of Engineering Department of Computer Engineering, Sepuluh November Institute of Technology
Surabaya, Indonesia

Afif Nurul Hidayati[5]
Faculty of Medicine Department of Dermatology and Venereology, Airlangga University
Surabaya, Indonesia

Ingrid Nurtanio[9]
Faculty of Engineering Department of Informatics
Hassanudin University
Makassar, Indonesia

***Abstract*—This paper discusses the implementation of smart contracts on the Ethereum blockchain system for telemedicine data storage. Telemedicine is one of the currently developing digital technologies in the health and medical sectors. Telemedicine can be more efficient when seeking treatment because patients do not need to see a doctor face to face. When using blockchain technology, the stored data becomes more transparent for each node in the blockchain network but has verification on every transaction which takes time and gas costs. However, telemedicine has several risks and problems, one of which is long data storage process time because there must be a verification process first to ensure data security. Another problem faced is the issue of the gas fee of the blockchain telemedicine system which is billed in every data storage transaction. In this study, a blockchain system was introduced for managing and securing databases on telemedicine. The implementation of this blockchain system was carried out on a website page that can add data to and retrieve data from the blockchain system. The results of this study showed that blockchain was successfully implemented to store telemedicine data with Ethereum. The analysis in this paper refers to the set and gets functions. The set function is used to send data to the blockchain, and the get function is used to retrieve data from the blockchain. From testing, the Get function has a much faster execution time than the Set function because the Get function does not require verification to retrieve its data. In the iterations carried out—namely 1, 10, and 100—the longest time on average was at 100 iterations when compared to the other iterations. In the tests carried out, the more characters that were stored, the more gas costs must be paid. In the tests, the percentage increase in costs was 0.34% per character.**

Keywords*—Blockchain; Ethereum; smart contract; telemedicine

I. INTRODUCTION

Health is one of the most important components in life. By being healthy, one's productivity will be good and one can work optimally. The health of a person can be more easily analyzed using data. However, a person's health information is confidential. One example of confidential medical information is an image of a medical record from a hospital. For that, we need a system mechanism to protect confidential data or information because this confidential information should not be known to just anyone. Blockchain technology can protect confidential information [1].

The blockchain system will make data storage decentralized. In general, information data are stored in a server such that the data are in one place only. This makes information or data on the server easy to hack because the data are centralized. Therefore, blockchain technology is reliable because this system decentralizes information data [1]. Thus, the use of blockchain is not managed centrally but is instead managed by each user in the network.

Blockchain has several frameworks that can be implemented, such as Hyperledger Fabric and Ethereum. These two types of blockchain have different functionalities. The difference seen is in the type of consensus. This consensus is built to build user trust in the blockchain network. A certain algorithm will be created for approval on all nodes in the network. All approved algorithms are code-based. Consensus has several types, such as Proof of Work (PoW), Proof of Authority (PoA), and Proof of Stake (PoS) [2]. Ethereum is a decentralized blockchain system. Ethereum is open source so that anyone can use it. On the Ethereum platform, the cryptocurrency unit used is Ether (ETH).

Smart contracts on Ethereum can manage transactions from every node on the blockchain network [1]. This managed transaction will become a record that will be propagated to all nodes. In telemedicine, patient data collected by the telemedicine center are a node because of the party who has the disease data or complaints at the time of treatment. The data from these patients will be decentralized with the blockchain network so that the data are much more secure and are in hashed form.

An image data management system for telemedicine transactions was developed by implementing blockchain technology to create transparency in the transaction. This transparency can improve the data integrity of telemedicine transactions through a decentralized system so that transaction activities in telemedicine can gain the trust of consumers. In the system created, image data on patients can be uploaded to the blockchain network, and medical parties can see these data in the network [1]. This makes the data much more secure because in the blockchain network there is a separate authentication and verification process. Image data can be accessed on nodes that have been confirmed and registered as valid. Therefore, the researcher designed a medical image recording application with blockchain using Ethereum.

There are various frameworks on the blockchain that can be used to develop a system, such as Hyperledger and Ethereum. The Hyperledger framework is described in the paper "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains" [3]. From paper "Design and Implementation of Storage System for Real-time Blockchain Network Monitoring System" This paper describes how to monitor the blockchain network using an apache engine that analyzes the block size of each block on the blockchain [4]. For data management, the paper "Analysis of Data Management in Blockchain-based Systems" describes how data management can be implemented on a blockchain network and the quality of the data [5]. The implementation of Blockchain technology using Ethereum in healthcare has been described in the paper "Decentralized Telemedicine Framework for a Smart Healthcare Ecosystem" but in this paper there is no data other than String data that can be stored (images that are converted to other forms also do not exist) [1]. The healthcare flow mechanism that uses Blockchain is discussed in the paper "A Blockchain-Based Smart Contract System for Healthcare Management", the analysis of the costs required for the analysis of each parameter is mentioned in the paper [6]. To develop a user interface for interaction with users the paper "The Implementation of Blockchain in Banking System using Ethereum" discusses using react.js with development using Ethereum [7].

The purpose of this research was to design and develop a blockchain system for use in telemedicine image data storage. A smart contract function on Ethereum was designed to send and retrieve telemedicine image data to and from the blockchain network. The flow of data on the implemented blockchain system in telemedicine was examined, and the performance of this blockchain system was analyzed. In this study, the implementation is limited to analyzing the system from the blockchain only. For the telemedicine process, this research only implements uploading and downloading images through a simple website page. So the analysis used is to analyze the upload time, download images, and analyze the cost of each transaction.

II. Basic Concept of Telemedicine System using Smart Contract Ethereum

A. Blockchain

Blockchain is a digital data storage system that contains data and records that are connected to a cryptographic system [8]. Currently, the best-known blockchain technology is cryptocurrency transactions, one of which is Bitcoin. Blockchain technology takes the form of recording digital transactions that exist on many servers.

Blockchain has many sets of blocks that contain information. Each of these blocks will have a hash component. This hash is the character set that composes the information in the block. The character hash is entered sequentially for each piece of new information. Thus, the information from each block from blockchain will continue to add the new hash value to be recorded so that all data are not lost.

Each block on the blockchain will form a blockchain network. Any data will be replicated to all networks. Every computer that exists and is connected to this network will execute the program at the same time. Therefore, blockchain is arguably a computer on a large scale formed from communication between several computers. If it is implemented in a database, then blockchain makes the database a decentralized system.

Blockchain has characteristics that can benefit some systems, including a chain-like structure that stores and updates a chain of transaction data. Each block will store transaction data through one consensus rule to validate. Other characteristics of the blockchain system include:

- Decentralization: The main advantage of blockchain systems is the decentralization of data. As data centralization has many vulnerabilities, blockchain can be a solution. The consensus mechanism will validate the transaction. The decentralization of data on the blockchain will make the data in the network well verified so that security is much better than that in centralized data.
- Transparency: The blockchain network is data decentralized such that every node on the network can see the transactions. The principle of the blockchain network will also record every transaction and distribute it to all nodes on the blockchain network. Private and public keys in blockchain can also help data security such that even though they are transparent, blockchain still maintains data security.

Blockchain technology can be implemented in several ways in the expected conditions. There are three types of blockchain:

- Public Blockchain: A public blockchain is a blockchain that anyone can access and use. Public blockchains are not controlled by any individual or organization. The ledger on the blockchain is open and transparent. However, there are drawbacks to public blockchains, namely high operating and maintenance costs, and slow transaction speeds. Examples of its use are in Bitcoin, Ethereum, and Hyperledger.
- Private Blockchain or Permissioned Blockchain: Private blockchains are formed to facilitate the private exchange of data among a group of individuals or organizations. Unknown users cannot access this blockchain network without a special invitation. An example of its use is on the R3 Corda.

- Blockchain Consortium: The blockchain consortium is a combination of public and private blockchains where there is no single organization that is responsible for controlling the network; instead, the network is controlled by several predetermined nodes. These nodes can decide who can be part of the network and who can be miners. For block validation, a multi-signature scheme is used in which a block is considered valid only if it is signed by some of these nodes. An example of its use is on Fabric.

A consensus algorithm is an algorithm used to validate the data. The algorithm have hash value for each block that has been formed in the blockchain. This hash value can be formed by converting inputs, reference hashes, and random numbers using the SHA-256 hash algorithm, which produces a hash value with a certain pattern [9]. However, on Ethereum, the hash used in general is Keccak-256. This hash is used on Ethereum addresses derived from public keys or contracts. Ethereum addresses are hexadecimal numbers, with the identifier derived from the last 20 bytes of the Keccak-256 hash of the public key. Unlike Bitcoin addresses, which are coded in the client's user interface or displayed to include a built-in checksum to protect against typos, Ethereum addresses are written as raw hexadecimal without any checksums.

The PoW mining process is carried out with dependence on computing power; while in PoS, the validation capacity depends on the stakes on the network. Prizes in the form of cryptographic money are given to miners when they can solve cryptographic puzzles on PoW and transaction fees on PoS. Attacks that may occur in each consensus are also different. In PoW, it takes a large computation about 51% larger than the existing blockchain network. On the other hand, PoS can be attack if 51% of cryptocurrencies are already in the hands of hackers, but this is impossible.

Another consensus that can be used is the Proof-of-Authority (PoA) consensus. The PoA usually used in the Ethereum private network. This consensus has good performance as well [10]. The PoA consensus have different system because everyone can be a node in blockchain network. As such, it is different from the type of blockchain that uses permissions on each node. The PoA consensus is used in test networks, one of which is Rinkeby, which can be used to develop a system.

In PoA, to generate a new block is granted to a node that has proven its authority operate. Such nodes are referred to as "Validators," and they run software that allows them to place transactions in blocks. The process is automated and does not require validators to continuously monitor their computers but require good computer maintenance. PoA is suitable for both private and public networks.

There are several advantages to using PoA. It has a fairly large risk tolerance, the block creation time is predictable, and it can be used more sustainably because it does not require large computations such as PoW. But, on the other hand, PoA has a drawback, namely that the validator can be seen so that there can be manipulation from third parties.

B. Telemedicine

Telemedicine is an information technology-based health service that allows patients to consult with doctors or other health experts without having to meet [11]. This innovation in health services with the internet can help patients use their time more efficiently because they do not have to come to a hospital or health facility for a consultation. During the remote consultation, the doctor helps the patient to get information regarding the suspected diagnosis, treatment, or first treatment for illness or injury, as well as tips to improve body health.

However, telemedicine also has a limitation in that doctors cannot detect where the patient is sick. This makes the diagnosis less accurate [12]. But if it is developed further, then telemedicine can be a breakthrough. Doctors and patients alike will be able to rest more and maintain health. Some implementations of telemedicine are being developed, one of which is the detection of skin diseases through images and disease consultations through doctors. This innovation in health services with the internet can help patients use their time more efficiently because they do not have to come to a hospital or health facility for a consultation. During the remote consultation, the doctor helps the patient to get information regarding the suspected diagnosis, treatment, or first treatment for illness or injury, as well as tips to improve body health.

Telemedicine has various types, three of which are as follows [13]:

- Real-time Interactive Medicine: This type allows communication between patients and doctors in realtime when there is a complaint from the patient.
- Store and Forward: This type allows owners of patient data to share their data with other practitioners.
- Remote Patient Monitoring: This type allows a medical professional to monitor patients remotely using a mobile medical device to collect data, such as blood pressure and blood sugar levels.

The use of blockchain in telemedicine has many benefits. It was mentioned in another paper that there are 6 properties of blockchain that can be useful in implementing telemedicine[14]. The first is its decentralized nature which allows data records to be accessible and managed in multiple locations. Then there is the immutability nature of the blockchain which makes patient data records cannot be changed once the data has been entered into the blockchain system, then there is asymmetric key cryptography that supports immutability. Next is data transparency which makes data exchange always traceable. Furthermore, there is also an important feature, which is open source so that patients can see the doctor's profile first. The nature of auditability and anonymity is also important to trace and hide the identity of user data.

The implementation of a simple application for telemedicine interaction using blockchain can also be applied to remote monitoring. For the selected storage, using several tools such as Filecoin, Storj.io, Napster, and IPFS [14]. The storage is usually intended to store documents that have a large enough size such as images or videos. In this section,

blockchain will play an important role in every data storage process that will be connected to the main system, users, and doctors.

C. Ethereum

Ethereum is an open-source blockchain platform [15]. Ethereum uses PoS as its consensus algorithm, but several test networks use PoA as well. To run transactions, Ethereum uses smart contracts with the Solidity programming language. The type of cryptocurrency used in Ethereum is ether (ETH).

Solidity is a high-level programming language that is better known as being contract-based. The syntax of this programming language is similar to the Javascript programming language. This programming language is used to improve the performance of virtual machines on Ethereum. Solidity is a scripting language that is statically created for verification and compile-time constraints. In addition to the time of compilation, this programming language will also help check at runtime. The Solidity programming language also supports object-oriented programming, such as object inheritance.

Solidity is a programming language created specifically for smart contracts on Ethereum [16]. Solidity is written in the .sol storage format. Solidity is not an executable language on a blockchain virtual machine but rather a language that aims to make it easier for developers to create smart contracts. When compiling or deploying smart contracts, a Solidity Compiler is needed. Through the Solidity Compiler, the smart contract is compiled into bytecode, which is then executed by the virtual machine.

In the Solidity programming language, several data types can be used, namely strings, integers, and Booleans, and others. Strings are generally a variable from a set of characters. Integers are used for data types in the form of positive or negative numbers. Boolean is a data type with a Boolean variable (true or false). Solidity includes new resources or new languages so that the code documentation of this programming language is not too excessive.

A smart contract is a script that is stored in a blockchain system. This smart contract has a unique address and consists of executable functions and state variables [17]. The user launches the smart contract by addressing the transaction to the contract. Once launched, the code of the smart contract cannot be changed.

The workings of this smart contract correspond to the basic function of Ethereum. Because Ethereum will work based on the exchange of information on an account owned by its users, when a smart contract is created, there must be an identification for each actor entity involved in the network. Then, in the existing smart contract, certain functions are added according to the design. This function will manage the exchange of information on the existing blockchain network.

To execute this smart contract, a function is triggered. This function depends on the code used along with the programming logic that has been made previously. There are two types of accounts in Ethereum; the first is Externally Owned Accounts (EOA) and the second is contract accounts [17]. EOA is controlled by the private key assigned to the user and the public address used to send and receive ETH from other accounts and send transactions to smart contracts. The contract account, on the other hand, does not have a private key and can only be activated by EOA. The contract account is where the smart contract resides in Ethereum.

Smart contracts are written in the Solidity programming language, which is a high-level programming language [18]. Users use EOA to make transactions on a contract account. This account is encrypted with a private key to be able to send transactions to other nodes. After that, other users verify the integrity of the transaction. This happens until a consensus is reached that has been agreed upon previously. The transaction will then be added to the block and will be recorded. The status of the network will also be updated as the smart contract has been successfully executed. The presence of this smart contract allows developers to send transactions with specific data and can check transactions between servers and inputs entered [19]. The checks performed are almost the same as the checks performed on the database. However, the specific difference is that there is an address for each data stored in the blockchain network.

In implementing the blockchain system for telemedicine, the Ethereum framework is used. Ethereum is a cryptocurrency-based blockchain like Bitcoin and is based on a public network, but it can also be used to implement a permissioned blockchain. Like Bitcoin, Ethereum also implements the PoW protocol [20]. In this implementation, a blockchain network is used which is a test network and coins from fees for each transaction that are executed can be obtained from a faucet on a website. So for every transaction that will be executed, a fee will be obtained from a faucet. Because it uses a test network, the consensus of the blockchain will change to Proof of Authority (PoA). The most important feature of Ethereum is that it supports the execution of a smart contract, which allows decentralized applications to build on top of it [20]. This is one of the considerations for choosing a framework. With the smart contract, the data that will be stored in the blockchain can be structured. So the implementation is generally not too different from the database implementation. When compared to other frameworks such as bitcoin, Ethereum has advantages such as private transactions and has higher transactions per second. When compared to Hyperledger Fabric, Ethereum has limitations, one of which is that it has a smaller throughput compared to other papers [20]. From the implementation using Ethereum, it can be investigated whether Ethereum can still be implemented to store image data with low transaction throughput.

D. Node.js

Node.js is an open-source, cross-platform platform for developing server- and network-side applications [21]. Node.js applications are written in JavaScript and can be run in the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js is a platform built on top of the Chrome JavaScript runtime to easily build fast and scalable network applications. Node.js uses an event-driven and non-blocking I/O model that makes it lightweight and efficient. The use of this platform is

suitable for data-intensive real-time applications running across distributed devices.

The features of Node.js are as follows:

- It is asynchronous; all API libraries in Node.js are asynchronous, which means that Node.js-based servers never wait for the API to return data.
- The speed is good; the execution speed is quite fast because it uses the Google Chrome V8 JavaScript Engine.
- It is a single thread but still scalable; Node.js uses a single-threaded program, and the same program can provide services to a much larger number of requests than traditional servers like the Apache HTTP Server.

In Node.js there is Web3, which is a collection of JavaScript Libraries (GNU Lesser General Public License version 3) that possess functionality for interacting with the Ethereum ecosystem. Web3 is built by the Ethereum Foundation and includes functionality to communicate with Ethereum nodes via Object Notation - Remote Procedure Call (JSON-RPC). Web3 allows users to interact with other Ethereum nodes using the HTTP, IPC, or WebSocket protocols [21]. Communication involves reading and writing data from the blockchain via smart contracts. Web3 providers can be set up on the frontend and backend to send transactions and listen for events happening on the blockchain network. With this, Node.js can be used because it has a library.

III. Telemedicine System Design using Blockchain

In this study, the design of a blockchain model for a telemedicine system used hardware and software with the following specifications. The hardware specifications used in this study are as follows:

- Manufacturer
- Asus ROG GL553VD
- Processor
- Intel® Core™ i7-7700HQ
- Graphics adapter
- NVIDIA GeForce GTX 1050 (Laptop) - 4096 MB; Core: 1354 MHz; Memory: 7000 MHz, NVIDIA GeForce GTX 1050 21.21.13.7654, Optimus
- Memory
- 16384 MB, DDR4-2400
- Storage
- Travelstar 7K1000 HGST HTS721010A9E630, 1000 GB, 7200 rpm, 1 TB HDD, 7200 RPM, 930 GB free
- Operating System
- Windows 10 (64-bit)

The following software specifications were used in this study:

- Node.js : version 14.15.4
- Solidity : version 0.4.25 - 0.7.0
- Metamask : version 9.5.4
- Truffle : version 5.3.1
- Git : version 2.24.1.windows.2
- NPM : version 6.14.10
- Solidity compiler : version 0.6.12

A. System Design

Smart contracts on Ethereum can be leveraged to ensure the integrity of transactions from patient care to the doctors involved. This can reduce the risk of scattered data and increase the scalability and transparency of a transaction. More specifically, this smart contract permits participating entities, especially doctors, to monitor and track all transactions on the network. In this way, both doctors and patients can feel more secure because their personal health data details are stored securely.

The blockchain technology design implemented in this telemedicine system has the advantage of being immutable. This makes the data that has been stored on the blockchain immutable as well, i.e., the data can no longer be changed. This feature can support data security from telemedicine data in the form of images represented as hash values. Access to stored data is also secured by smart contracts so that only interested parties can access the data.

The system applied in this study used a simple web page to upload the required medical images. There are two entities, namely doctors and specialists. Doctors upload images that are analyzed by specialists. The implementation is done using Node.js to run Web3, a metamask for the wallet system on transaction fees, and Infura.io to run smart contracts.

To ensure secure telemedicine and blockchain-based healthcare data transactions, in this scenario, the researcher designed a framework for patients to provide data to doctors for analysis. This agreement was carried out at the telemedicine center by approving health care data with the health organization as shown in Fig. 1. After that, the patient made an appointment with a doctor for consultation, after which the doctor gave a consultation and medication, and then updated the patient's data. The hash value was stored as a transaction when the patient was consulted. The physician could then provide access to the patient's health care data to medical research organizations to conduct trials and clinical trials. After providing a diagnosis according to the patient's illness, the data were sent back to the puskesmas through the same process, after which the puskesmas sent the data to the telemedicine center.

The problem faced today in a telemedicine system is the use of a database system that uses one server. The idea of this paper is to improve data security using a storage system on a blockchain network. But, the difference in the network used for data storage can make a difference in the speed of data storage. In this paper, we discuss how the speed of data storage to the blockchain network is discussed. A variation of the

implemented implementation is the amount of data sent to the blockchain (String). This paper involves image data that has been converted into a hash because it has a more efficient value compared to uploading the full image to the blockchain system, and the blockchain system also has a limitation on the size of the uploaded file to produce data in the form of a String which is sent to save gas usage on Ethereum.

In the sequence diagram in Fig. 2, we can see the sequence of the blockchain system for telemedicine. First, the doctor uploads pictures to send information. If the patient agrees, then the treatment process is complete. After that, the central telemedicine system in the form of a website sends image data to the blockchain system in the form of a hash value of the IPFS value. Next, the specialist who analyzes the image data continues to the center so that the hash can be known, and the image can be retrieved for analysis.

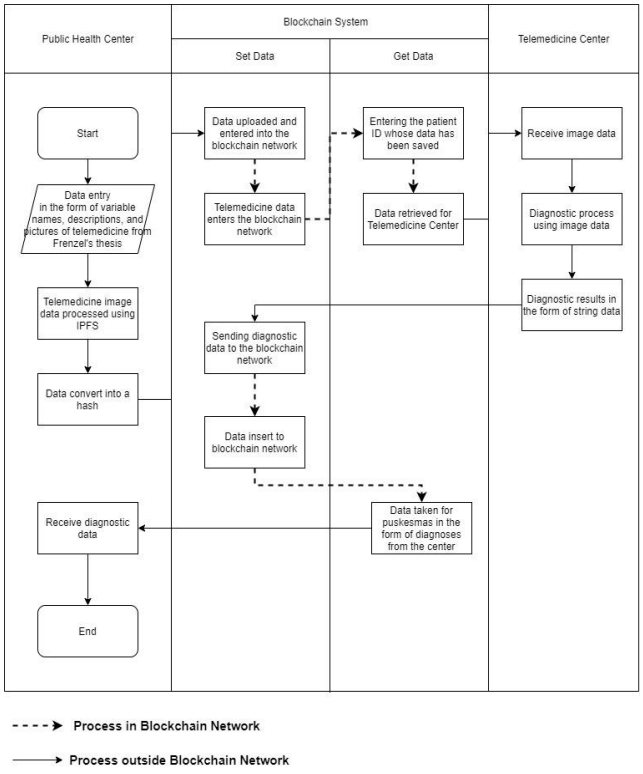

Fig. 1. Activity Diagram for Blockchain Implementation.

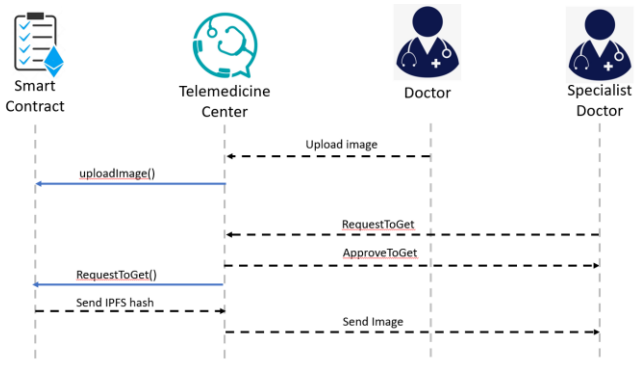

Fig. 2. Sequence Diagram for Blockchain Implementation.

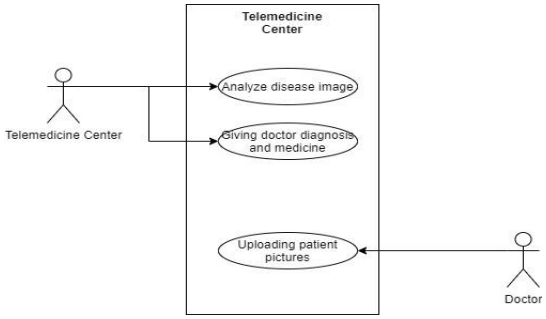

Fig. 3. Use Case Diagram for Blockchain Implementation.

In the use case diagram in Fig. 3, the telemedicine center analyzes the existing images and also provides a diagnosis of the disease from which the patient may be suffering. Meanwhile, the doctor only uploads pictures of the patient, which are then analyzed by specialist doctors at the telemedicine center.

B. System Design

The implementation scenario on this system was programmed in the Solidity programming language with web browser-based Remix. Entities participating in the framework were identified using their Ethereum addresses within the blockchain network. Communication between entities was possible by calling functions in smart contracts. Doctors uploaded notes in the form of images in an entity that recorded all patient care documents stored in smart contracts. Consequently, there was a flow-on of each doctor entity until the data were stored in the blockchain network using smart contracts. These stored data were in the form of a hash value generated by the IPFS system.

In Fig. 4, the implementation of scenarios that can be combined into steps can be seen.

- The doctor opens a website to upload patient data.
- Image data are telemedicine data that have been encrypted using Frenzel's thesis.
- The smart contract contains input data in the form of ID, name, description, and IPFS hash, which is generated when the image is uploaded to the IPFS system.
- The state contract is idle until there is a transaction.
- The state contract is ready to be submitted after a transaction occurs at the telemedicine center in the form of parameters set by the smart contract in the form of strings and image uploads that are converted into hash values.
- After being successfully verified, the ID appears automatically through an alert.
- Image (IPFS hash) and data are entered into the blockchain database using IPFS and smart contracts.
- To retrieve the data, the ID is needed. This Get process returns the value of the parameter that was sent previously on the blockchain network.

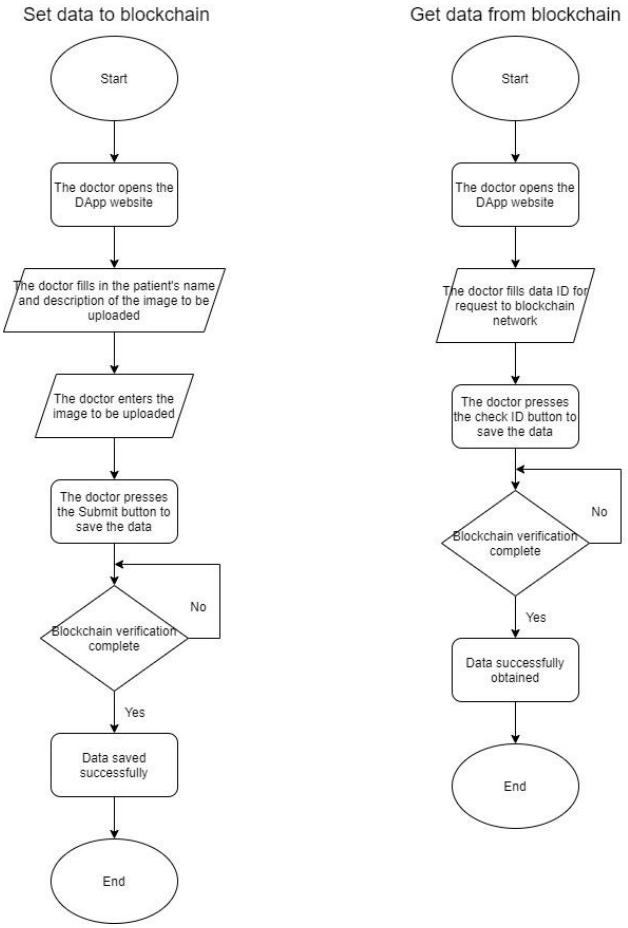

Fig. 4. Get and Set Function Scenario.

C. Collecting Data Scenario

In Fig. 5, data retrieval is done by taking the execution time of the Get and Set functions on the blockchain system. This data retrieval uses a testing function from Truffle with the Chai library. The program that runs this test is Upload.test.js, which contains tests for the Get and Set methods on the blockchain network. The test is carried out with the same data declared at the beginning of the code, which is iterated for 1, 10, and 100 transactions, respectively. The following parameters are measured in this scenario:

- The time of each Get transaction and Set transaction in milliseconds is taken on average for each iteration.
- The difference in the amount of gas fee used by the transaction is based on the number of strings entered in the description.

The system test scenario, especially the Get and Set functions, is executed with a Javascript file called Upload-test.js. This file is in the test folder in the project documents using the React framework. This document will integrate with the scripts installed in the module package from Node.js that are run using Truffle.

Fig. 6 is a flowchart of the Upload-test.js. This document contains instructions that one wants to perform in the test. There are two important functions in this document, namely the describe() function and the assert() function. The describe() function is tested using the Mocha library. Another library that can be used is from Truffle using the contract() function; but in this test, the Mocha function was used. The assert() function is used to create a condition that must be met—for example, there is a condition for sending data. However, in this test, the testing scenario was directed to focus more on the describe() function to get the test results in the form of time.

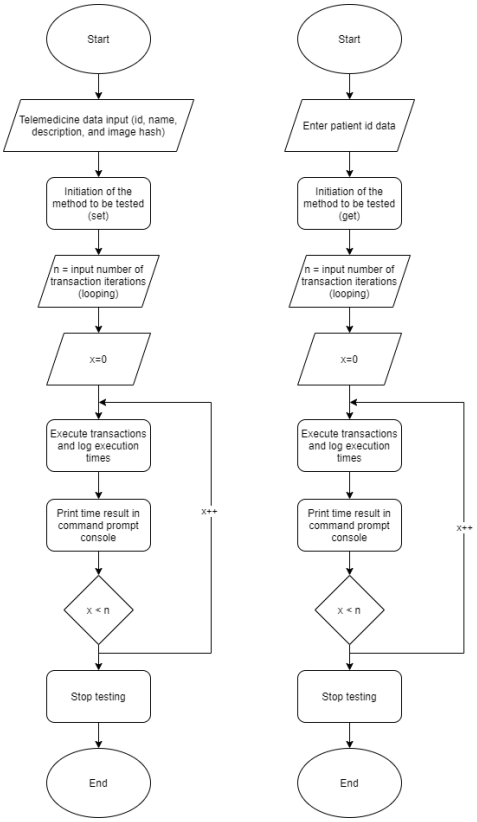

Fig. 5. Collecting Data Diagram Scenario.

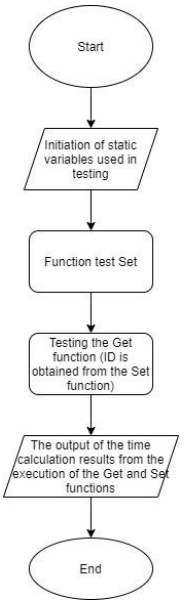

Fig. 6. Process Testing Set and Get Function Diagram.

IV. Test Results and Analysis

The test was carried out to determine system performance based on the duration needed to send and retrieve data on the blockchain system. This time was measured using a Javascript document that was run using Truffle. The measurement time was based on the execution time of the Set and Get functions on the telemedicine data. This time measurement was performed because the blockchain network system was run on Infura.io, which uses the Rinkeby test network such that the performance of the blockchain can be known to add new blocks to the network. This performance test was performed using a Javascript document that was executed using Truffle.

The test was distinguished by iterations and the number of strings that filled the description variable. Iterations or repetitions were varied into three, namely 1, 10, and 100 iterations. This iteration was performed as a burst value that was used to load which is referred to this test performed on the blockchain test network. Iteration variations were carried out to determine the rate of three repetition variations with the previously mentioned number. As for the number of strings, these were divided into three as well, namely:

- string1 (contains 39 characters, 6 words, and 5 spaces)
- string2 (contains 96 characters, 16 words, and 15 spaces)
- string3 (contains 161 characters, 28 words, and 27 spaces)

Test implementation is using only string not image from telemedicine data to reduce gas cost from a transaction. It is possible to store image data directly to blockchain but small image. Ethereum network has an 8M gas block limit. Every new 32 bytes of storage uses 20k gas, because the cost of data storage is 640K gas per kilobyte of data and the current gas price is approximately 15 GWEI [22]. So the system can't store data that sum to more than 12.8 kb. The solution is to use IPFS to convert images as IPFS hash so only string data will store at blockchain network and will also reduce gas costs as a whole. Due to the limitations of sending data on each transaction on the blockchain network, the variation in the number of strings is not too large. This is done because the test does not test limitations and only examines the effect of variations on the time and gas costs charged for each transaction.

From each of these tests, the average time was taken to be able to represent the calculation of the time of each iteration and each string difference. This time corresponded to the time needed to make transactions, apart from the time needed for miners to mine because the time for mining blocks was 15 seconds, which, in all transactions, was the same value because it used the Rinkeby test network.

A. Comparison and Analysis of Time in the Set Function and Get Function

In Table I, a comparison between the average times based on the number of iterations performed on each string can be seen. Comparisons were made by comparing the average time obtained in each iteration. For this reason, in this comparison, the 1x iteration as compared to the average time of the Set and Get functions that applied to all iterations. Thus, the results of the comparison of the amount of time taken to perform the two functions are as follows.

- 1x iteration = 12663.3 / 576.3 = 21.97 times faster
- 10x iteration = 15180.2 / 589.5 = 25.75 times faster
- 100x iteration = 15321.3 / 594.2 = 25.78 times faster

From the average amount of time taken to obtain each iteration, it can be seen that in the 1x iteration the execution speed was 21.97 times faster in the Get function than in the Set function. For speed, the biggest difference between the Set and Get functions was in 100x iterations, which was 25.78 times faster than the Set function than in the Get function. The significant difference was in the 1x iteration, where the average iteration obtained tended to be small. This happened because the initial initiation and lack of repetition made the blockchain network not load transactions at almost the same time. The absence of this transaction burden made the time required for verification of the Set function faster, in contrast to string data that have repetitions or iterations greater than one time.

Time parameter in this implementation is important. The time parameter in this implementation will compare the time it takes to send and retrieve data because there are verification differences. At the time of sending data, a new transaction will be made that requires gas fees and will take time to verify. While in the data collection process there is no verification process at all. Verification on this test network sometimes takes a long time, resulting in data that is much longer than the average time.

B. Comparison and Analysis of Gas Fee Results with String Variations

This test was intended to determine how much it costs to store data on a blockchain network. Keep in mind that the blockchain network used was the Rinkeby testing network. However, this test can be used as a reference for the number of fees that must be paid when storing data in the blockchain network. The test was varied using the number of letters, words, and spaces in the "description" variable in the form of a string. In detail, the cost of each string was as follows:

- string1 : 0.000205 ETH
- string2 : 0.00025 ETH
- string3 : 0.000296 ETH

TABLE I. Average Time Comparison between Set and Get Functions

String / Iteration	Function					
	Set			Get		
	1x	10x	100x	1x	10x	100x
String1	12027	14857.9	14881.3	571	546.7	593.3
String2	12962	15687.4	15465.7	552	558.6	592.1
String3	13001	14995.3	15616.8	606	663.3	597.3
Average	12663.3	15180.2	15321.3	576.3	589.5	594.2

When string1 was compared to string2, an additional cost of 0.000045 ETH, or 21.95%, occurred for storing string2 when compared to the cost required to store string1. This means that there were additional costs when data were added. It can be concluded that the addition of special characters from string1 to string2 involved 57 characters. The addition of these 57 characters required an additional fee of 0.00045 ETH. In this test, spaces were ignored because they did not affect the amount of data stored. From the comparison test of string1 and string2, it can be concluded that the average additional gas cost is 0.38% per character.

In the next test, we compared the transaction gas costs using string2 and string3. The difference in gas costs between the two was 0.000046 ETH. Then, concerning the number of characters, string2 had 96 characters and string3 had 161 characters. If these two variables are separated, it produces 65 characters. For the cost of gas required from string2 to string3, there was an increase of 18.4%. If the average value per character is taken, an increase of 0.28% is obtained for each character.

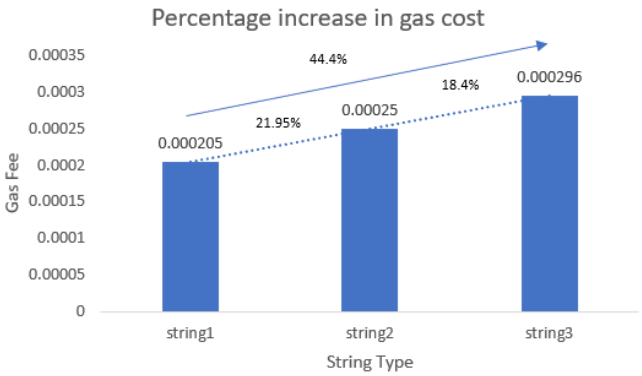

Fig. 7. Average Time Comparison between Set and Get Functions.

Fig. 7 depicts a graph of the percentage increase in the number of gas costs. The rising line has a linear tendency. Therefore, it can be concluded that the amount of string data is directly proportional to the number of gas costs that must be paid. However, in the tests carried out, the percentage increase in gas costs was slightly inappropriate. This discrepancy can be seen in Fig. 9, where the percentage increase from string1 to string3 is not the same as the sum of the percentage increase from string1 to string2 plus string2 to string3. This could be because the number of spaces was not taken into account or was ignored in this test. Thus, the percentage increase in costs had an error of 4.05% in this test.

V. Conclusion

In the tests carried out in this work, several conclusions from the implementation can be made. First, we can conclude that the implementation of smart contracts on Ethereum as a blockchain system for application to telemedicine data repositories was successfully achieved. Thus, telemedicine treatment can improve the effectiveness of a person's time and can secure data. In addition, the performance of the blockchain was determined by the cost of the gas used. Using PoA, the validator block was limited in number to make the network more scalable and faster in the transaction process. From the tests carried out, the performance for retrieving data from the blockchain was 24.4 times faster than adding data to the blockchain system. This is because the addition of data first requires a transaction verification such that the number of required time increases. Additionally, the more data one wants to store (in this study, in the form of string data), the more average time is required to make a transaction. The additional fee is as previously mentioned, the cost of each verified transaction will be following the amount of data that will be stored in the blockchain network. At the time of the experiment, the success rate of data transmission and retrieval was 100%, but there were some time spikes from time testing, especially in the Get function at 100 iterations. This was due to pending transactions at the time of verification. The gas fee that must be paid to add data to the blockchain system or for transaction validation fees to the validator was directly proportional to the amount of data stored in the blockchain. Therefore, from the experiment, it was shown that the larger the amount of data stored in the system, the higher the gas costs incurred. In this test, an additional cost of 0.34% per character string was added. The error obtained from the gas cost comparison experiment was 4.05%.

Acknowledgment

This research is supported and funded by the Directorate of Research and Community Service, Deputy for Strengthening Research and Development, Ministry of Research, Technology / National Research and Innovation Agency of the Republic of Indonesia under the grant of Penelitian Konsorsium Riset Unggulan Perguruan Tinggi, contract number: 1056/PKS/ITS/2021 between researchers and Direktorat Riset dan Pengabdian kepada Masyarakat, Institut Teknologi Sepuluh Nopember.

References

[1] A. Abugabah, N. Nizamuddin and A. A. Alzubi, "Decentralized Telemedicine Framework for a Smart Healthcare Ecosystem," in IEEE Access, vol. 8, pp. 166575-166588, 2020, doi: 10.1109/ACCESS.2020.3021823.

[2] "Analysis of PoW, PoS and PoA," [Online]. Available: https://www.programmersought.com/article/2017130150/. [Accessed 21 July 2021].

[3] Androulaki, Elli & Barger, Artem & Bortnikov, Vita & Cachin, Christian & Christidis, Konstantinos & Caro, Angelo & Enyeart, David & Ferris, Christopher & Laventman, Gennady & Manevich, Yacov & Muralidharan, Srinivasan & Murthy, Chet & Nguyen, Binh & Sethi, Manish & Singh, Gari & Smith, Keith & Sorniotti, Alessandro & Stathakopoulou, Chrysoula & Vukolic, Marko & Yellick, Jason. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains.

[4] J. Bang and M. Choi, "Design and Implementation of Storage System for Real-time Blockchain Network Monitoring System," 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2019, pp. 1-4, doi: 10.23919/APNOMS.2019.8892967.

[5] H. Paik, X. Xu, H. M. N. D. Bandara, S. U. Lee and S. K. Lo, "Analysis of Data Management in Blockchain-Based Systems: From Architecture to Governance," in IEEE Access, vol. 7, pp. 186091-186107, 2019, doi: 10.1109/ACCESS.2019.2961404.

[6] Khatoon, A. A Blockchain-Based Smart Contract System for Healthcare Management. Electronics 2020, 9, 94. https://doi.org/10.3390/electronics9010094.

[7] Bakaul, Masum & Das, Nipa & Moni, Madhabi Akter. (2020). The Implementation of Blockchain in Banking System using Ethereum.

International Journal of Computer Applications. 177. 50-54. 10.5120/ijca2020919895.

[8] T. Annisa, "Mudah, ini penjelasan dasar blockchain untuk pemula," Ekrut Media, [Online]. Available: https://www.ekrut.com/media/blockchain-adalah. [Accessed 20 December 2020].

[9] J. Bang and M. Choi, "Design and Implementation of Storage System for Real-time Blockchain Network Monitoring System," 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2019, pp. 1-4, doi: 10.23919/APNOMS.2019.8892967.

[10] "Proof of authority consensus," [Online]. Available: https://www.geeksforgeeks.org/proof-of-authority-consensus/. [Accessed 25 May 2021].

[11] R. W. Ahmad, S. Khaled, R. Jayaraman, I. Yaqoob, S. Ellahham and M. Omar, "The Role of Blockchain Technology in Telehealth and Telemedicine," International Journal of Medical Informatics, vol. 148, no. 18:104399, p. 1, 2020.

[12] A. Efendi, "Mengenal Telemedicine Beserta Kelebihan dan Kekurangannya," 13 May 2020. [Online]. Available: https://tirto.id/mengenal-telemedicine-beserta-kelebihan-dan-kekurangannya-fsnL. [Accessed 21 December 2020].

[13] V. Kamani, "3 Telemedicine Types for Every Healthcare Organization," arkenea, [Online]. Available: https://arkenea.com/blog/types-of-telemedicine/. [Accessed 21 December 2020].

[14] Ahmad, Raja & Salah, Khaled & Jayaraman, Raja & Yaqoob, Ibrar & Ellahham, Samer & Omar, Mohammed. (2021). The Role of Blockchain Technology in Telehealth and Telemedicine. International Journal of Medical Informatics. 148. 104399. 10.1016/j.ijmedinf.2021.104399.

[15] J. Golosova and A. Romanovs, "The Advantages and Disadvantages of the Blockchain Technology," IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), pp. 1-6, 2018.

[16] "What is Solidity Programming, its Data Types, Smart Contracts, and EVM in Ethereum?" Simplilearn, [Online]. Available: https://www.simplilearn.com/tutorials/blockchain-tutorial/what-is-solidity-programming. [Accessed 23 October 2021].

[17] A. Pinna, S. Ibba, G. Baralla, R. Tonelli and M. Marchesi, "A Massive Analysis of Ethereum Smart Contracts Empirical Study and Code Metrics," in IEEE Access, vol. 7, pp. 78194-78213, 2019, doi: 10.1109/ACCESS.2019.2921936.

[18] R. Ghods, "web3.js - Ethereum JavaScript API," 24 June 2020. [Online]. Available: https://web3js.readthedocs.io/en/v1.3.4/. [Accessed 18 April 2021].

[19] Yu, Hongru & Sun, Haiyang & Wu, Danyi & Kuo, Tsung-Ting. (2020). Comparison of Smart Contract Blockchains for Healthcare Applications. AMIA ... Annual Symposium proceedings. AMIA Symposium. 2019. 1266-1275.

[20] Agbo, Cc & Mahmoud, Qusay. (2019). Comparison of Blockchain Frameworks for Healthcare Applications. Internet Technology Letters. 2. e122. 10.1002/itl2.122.

[21] "About Node.js," OpenJS Foundation, [Online]. Available: https://nodejs.org/en/about/. [Accessed 13 April 2021].

[22] “Is it possible to store images on the Ethereum blockchain?, [Online], Available:”https://stackoverflow.com/questions/52994467/is-it-possible-to-store-images-on-the-ethereum-blockchain/52994971 [Accessed 21 October 2021].