International Conference on Rural Development and Entrepreneurship 2019: Enhancing Small Business and Rural Development Toward Industrial Revolution 4.0

Vol. 5 No.1
ISBN: 978-623-7144-28-1

Analysis and Implementation of Nodes Communication Between InterPlanetary File System (IPFS) in Smart Contract Ethereum

Achmad Muhaimin Aziz [1*], Adityas Widjajarto[2], Avon Budiyono [3]

[1] School of Industrial and System Engineering, Telkom University
[2] School of Industrial and System Engineering, Telkom University
[3] School of Industrial and System Engineering, Telkom University

Abstract. At present all business activities are bound to contracts or agreements. A written contract has several weaknesses, the contract can be lost and damaged, it is not cost effective and one party can commit fraud. The solution for that is to use the smart contract Ethereum. Smart contract Ethereum is a computer protocol that functions to facilitate, verify, or enforce digital negotiations written through the program code. Smart contract works without going through a third party and has a credible transaction process so that it cannot be tracked or changed. But Blockchain technology is not suitable for storing large amounts of data and expensive costs, the authors combine IPFS technology on Ethereum Blockchain. So the Ethereum Blockchain only stores the hash of the file, then the hash of this file can be connected to the file on IPFS to access it. In this study a web-based DApp (Decentralized applications) system was built that implemented IPFS on the smart contract Ethereum. The final result of this study is a discussion of data integrity and Quality of Service (QoS) communication between IPFS nodes on the smart contract Ethereum as a reference for implementation of the company. With the results of the implementation it was found that the data integrity possessed by IPFS was very good by fulfilling aspects of information security and having Quality of Service with average throughput values of

56.41 Kbps, 65.81 Kbps and 79.68 Kbps, for average packet loss values of 1.92%, 1.58% and 1.06%, while the average value of delay is 24.79 ms, 25.87 ms and 17.30 ms with the average value of the Quality of Service index which is 3 which meets the "Satisfying" category based on THIPON standards.

Keywords: Blockchain, Smart Contract Ethereum, IPFS, Data Integrity, Quality of Service (QoS)

* Corresponding Author, Email: achmadmuhaiminaziz8@gmail.com

Vol. 5 No.1
ISBN: 978-623-7144-28-1

1. INTRODUCTION

At present all business activities are bound to contracts or agreements, contracts or written data are not safe to use. The solution for that is to use the smart contract Ethereum. Smart contract Ethereum is a computer protocol that functions to facilitate, verify, or enforce digital negotiations written through the program code. Smart contract works without going through a third party and has a credible transaction process so that it cannot be tracked or changed. (Atzei, Bartoletti and Cimoli, 2017: 1-3)

Smart contracts are stored along with transactions on the Blockchain. The Blockchain uses distributed peer-to-peer network technology, so the Blockchain is the safest storage place for digital data such as cryptocurrency, smart contracts, property, shares, files, or valuable with the confidentiality, integrity and authenticity of dat1a. The Blockchain consists of several blocks that continue to grow and are connected by cryptographic algorithms. This is based on Distributed Ledger Technology (DLT), which is a system for recording digital transactions in distributed storage without having centralized storage. Distributed Ledger Technology (DLT). (Atzei, Bartoletti and Cimoli, 2017: 1-3).

To save data on Blockchain is quite expensive. Because the Blockchain is not suitable for storing large amounts of data, many developers create a DApp that integrates between IPFS and smart contract Ethereum. Thus, to store data on the Blockchain, it will be cheaper because the gas costs incurred for making contracts and making transactions are cheaper (Sinha and Kaul, 2018: 1209).

IPFS is a distributed file system that is peer-to-peer (P2P), serves to connect all computing devices with the same file system. IPFS combines successful ideas from previous peer-to-peer systems, namely DHT, BitTorrent, Git, and SFS. This combination aims to replace the current file distribution technique with more modern techniques that make it an interesting file system to learn (Benet, 2014: 03).

In this paper a web application system is built that is DApp (Decentralized Application) that integrates between IPFS and smart contract Ethereum. This application was created to replace the centralized file storage system such as Google Drive and Dropbox. This application will be analyzed based on Quality of service on IPFS networks and smart Contract Ethereum. The results of packet loss from IPFS will show how the data integrity of the upload process and file downloads. The research conducted is measuring the Quality of Service and the integrity of data from IPFS and smart contract ethereum as a reference for the implementation of a company.

Vol. 5 No.1
ISBN: 978-623-7144-28-1

2. RELATED WORK

2.1. Blockchain

Fig. 1: Blockchain Logo

Blockchain is a chain of data structures, similar to a general ledger that documents and verifies all transactions carried out by users or users. The Blockchain uses a concencus distributed node algorithm to generate and update transaction data using a block chain to verify and store transaction data. All transactions carried out will be stored in the block list. Chains will grow when new blocks are added continuously. Each block has a timestamp, its own cryptographic hash and a hash reference from the previous block listed in the block header. The maximum number of transactions in one block depends on the block size and size of each transaction (Chen, 2017).

2.2. Smart Contract Ethereum

Fig. 2: Blockchain Logo

Ethereum is a platform with blocks that have a smart contract function. Ethereum has a function like a virtual machine that runs a smart contract in a peer-to-peer manner with cryptographic money, namely Ether (ETH). The contract on the smart contract Ethereum is written in the Turing- Complete byte code language, called EVM byte code(Wood, 2018).

Smart contracts can be implemented if there is a transaction and each transaction requires a fee or Ether (ETH) (Galal and Youssef, 2018: 2). There are several reasons for users to make transactions on the Ethereum network, namely: (i) making new contracts; (ii) carry out the contract function; (iii) Ether transfers to contracts or to other users (Atzei, Bartoletti and Cimoli, 2017: 1-3).

Vol. 5 No.1
ISBN: 978-623-7144-28-1

2.3. IPFS

IPFS is a distributed file system that is peer-to-peer (P2P), serves to connect all computing devices with the same file system. IPFS combines successful ideas from previous peer-to-peer systems, namely DHT, BitTorrent, Git, and SFS. This combination aims to replace the current file distribution technique with more modern techniques making it an interesting file system to learn (Benet, 2014: 03).

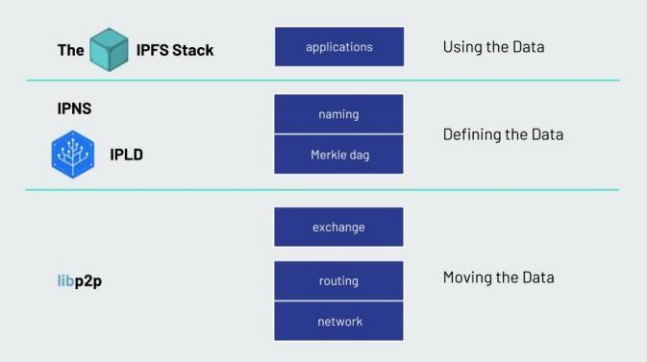

Fig. 3: IPFS Protocol Stack.

The IPFS node stores an IPFS object in a local storage. Nodes are connected to each other and transfer an object. These objects represent a file and other data structures. Fig. 1 shows the IPFS protocol stack which will be explained as follows.

1) Identities & Network

A node refers to the identity of a computing device that is in an IPFS network. The identity of nodes in an IPFS network identified by hash cryptography from a public key, created with S / Kademlia crypto is applied to ensure that each node is authenticated so that there are no "fake" nodes. The results of encryption are known as NodeID. To find the network address of all relevant nodes by using a routing system based on DHT (Distributed Hash Table). The DHT implementation used in IPFS is to save and secure NodeID and prevent Sybil's attacks (Wennergren et al., 2014: 05).

2) Routing

Routing in IPFS functions to manage information to be able to find peers and objects and be able to respond to local and remote queries. IPFS routing uses DSHT based on S / Kademlia and Coral. The size of the object and the IPFS usage pattern are similar to Coral and Mainline, so DHT IPFS can make a difference for stored values based on their size. For 1KB file size it is stored directly

Vol. 5 No.1
ISBN: 978-623-7144-28-1

on DHT. For large file size storage, DHT only stores references that are NodeID from peers that can serve a block (Benet, 2014).

3) Block Exchange - BitSwap Protocol

In IPFS, data distribution is done by exchanging blocks with peer using a protocol inspired by BitTorrent, namely Bitswap. Like BitTorrent, Bitswap peers looks for a set of blocks (want_list), and has another set of blocks to give (have_list). Unlike BitToreent, Bitswap has no block limits in one torrent. Bitswap operates like a marketplace where each node can get the blocks they need, regardless of what files are part of the block. Blocks can come from files that are completely unrelated to the file system (Benet, 2014).

4) Object Merkle DAG

Essentially IPFS is a peer-to-peer system for retrieving and sharing IPFS objects. IPFS object has a structure with two fields as follows (Benet, 2014):

a. Data, is BLOB from unstructured binary data with a size of less than 256 kb.

b. Links, is an array containing the link structure that connects IPFS objects with other IPFS objects. The link structure has the following fields:

- Name, the name of a link.
- Hash, is the result of an IPFS object that is connected.
- Size, the size of an IPFS object that is connected.

5) Files Transfer

File distribution occurs as follows. First, Bitswap will search the blocks needed by a peer by broadcasting "want_list" to all connected peers. After receiving a want_list, the node will save it, then check availability and send the desired block to the recipient. The recipient will verify the suitability of the file with a multihash checksum. If the file does not match the recipient has the right to reject the file and the file sharing process is stopped. After completing the correct block transmission the recipient sends confirmation of acceptance and moves the block from want_list to have_list. Some nodes are allowed to accept various blocks of files or all files at once.

Bitswap will save the file transfer history to Bitswap Ledger. After file sharing, both recipients and senders update their Ledger Bitswap to reflect the additional bytes transmitted. The more bytes recorded in Bitswap Legder, the node is increasingly trusted by Bitswap (Benet, 2014).

Vol. 5 No.1
ISBN: 978-623-7144-28-1

6) IPNS: Naming and Mutable State

The Interplanetary Name System (IPNS) is a system for creating and updating mutable links to IPFS content. Because objects in IPFS are content-addressed, addresses will change every time the content changes. It is useful for many things, but it is difficult to get the latest version. The name in IPNS is the hash of the public key. Regarding records containing information about hashes related to links that have been signed by private key records, they can be signed and published at any time.

IPNS is not the only way to create addresses that can be changed on IPFS. Can use DNSLink which is currently much faster than IPNS and uses a name that is easier to read. Because files in IPFS are aimed at content that cannot be changed, it becomes difficult to edit. Mutable File System (MFS) is a tool built into IPFS that allows to use name-based file systems that can add, delete, move, and change MFS files and make it possible to update previous links and hashes (Benet, 2014: 09).

2.4. CIA Triad Model

Data that becomes a source of information can be said to be safe if it meets three parameters, namely Confidentiality, Integrity, and Availability. The following is an explanation of the three parameters (CompTIA +, 2014).

1) Confidentiality, is information that is confidential and can only be accessed by certain parties.

2) Integrity or integrity, which is stored data, transmitted has integrity and has not changed.

3) Availability is the information that can be accessed by parties who have access rights, whether to view or modify it.

2.5. Quality of Service

Quality of Service (QoS) is a method that can be used to measure the quality of a network and is an attempt to define the characteristics and properties of a service. QoS is used to measure a set of attributes from performance that are specified and associated with a service (Cisco Internetworking, 2018).

1) Throughput

Vol. 5 No.1
ISBN: 978-623-7144-28-1

Throughput is a parameter that measures the speed (rate) of data transfer measured in bps (bits per second). The following are standards or recommendations from Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) regarding the value of a throughput:

Table. 1: Throughput Category.

Throughput Category	Throughput Value	Index
Very Good	>2.1 Mbps	4
Good	1200 kbps s/d 2.1 Mbps	3
Normal	700 Kbps s/d 1200 Kbps	2
Bad	< 700 Kbps	1

To get a value from a throughput can use the following formula

$$\text{Throughput} = \frac{\text{Total Packet (Kb)}}{\text{Duration (s)}}$$

2) Packet Loss

Packet Loss is a QoS parameter that shows a condition for the total number of packets lost due to collision and congestion on the network. The following are the Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) standards or recommendations regarding the value of a packet loss:

Table. 2: Packet Loss Category.

Packet Loss Category	Packet Loss (%)	Index
Very Good	0	4
Good	3	3
Normal	15	2
Bad	25	1

To get the packet loss value can use the following formula:

$$\text{Packet Loss} = \frac{(\text{Packet Sent} - \text{Packet Received})}{\text{Packet Sent}} \times 100\%$$

3) Delay

Delay is the time it takes for a package to reach its destination. Delay can be affected by distance, media, congestion or long processing times. The following are the standards or recommendations of Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) regarding the value of a Delay:

Vol. 5 No.1
ISBN: 978-623-7144-28-1

Table. 3: Delay Category.

Delay Category	Delay Value (ms)	Index
Very Good	< 150 ms	4
Good	150 ms s/d 300 ms	3
Normal	300 ms s/d 450 ms	2
Bad	> 450 ms	1

To get the value of delay can use the formula as follows:

$$\text{Delay} = \frac{\text{Duration (s)}}{\text{Total Packet}}$$

3. PROPOSED ARCHITECHTURE

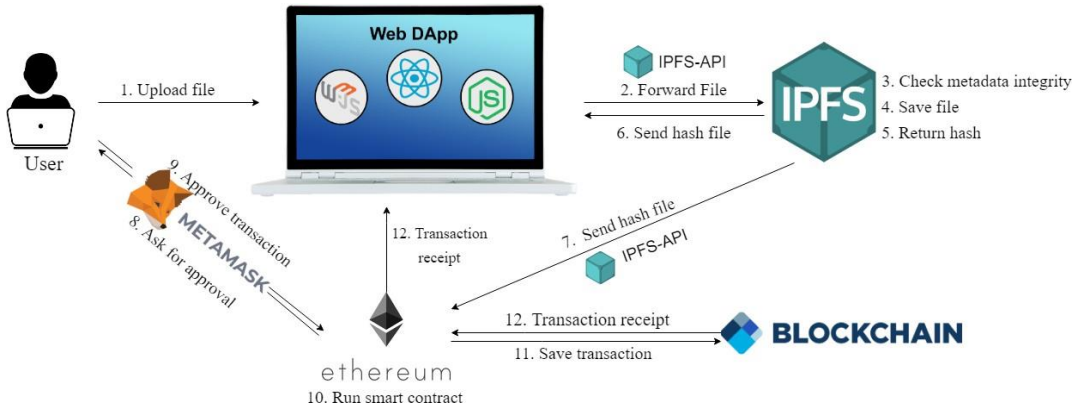

Fig. 4: Proposed Architecture.

A In this research a system or application was built, namely web-based DApp (Decentralized Application) using NPM. In the DApp web system, the integration of IPFS and smart contract is carried out using Ethereum using ipfs-api.

During the process of building the DApp web system, contracts are made using the program code written into Solidity through the web remix.ethereum.org The DApp web system functions as an application that runs Node.js and NPM programs that have several dependencies.

Users upload files through the DApp web system, then the browser will convert the file into a buffer. After the process of converting files to buffer has been completed, it is continued with the

Vol. 5 No.1
ISBN: 978-623-7144-28-1

process of uploading files using ipfs-api. Ipfs-api will process file uploads to IPFS. During the process of sending data via IPFS, configuration uses port 4001 during the data transmission process.

IPFS will check the content file based on the metadata of the file. Furthermore, IPFS will return in the form of an IPFS hash ("Qm ..."). IPFS will send several blocks to a number of connected nodes. After the file distribution process is complete, IPFS will send hash data from the file into the DApp web system and send raw data in the form of IPFS hashes to the Ethereum network written into the smart contract. Sending raw data in the form of an IPFS hash is done using ipfs-api and Metamask will ask for confirmation from the transaction to the user. Then the user will give approval of the transaction.

After the user gives approval, sending raw data in the form of an IPFS hash will be saved to the Blockchain storage. As proof of the transaction carried out the user will request a receipt transaction from Ethereum. Ethereum will send data in the form of transaction hash, block, and gas used.

4. ANALYSIS

4.1. Analysis Quality of Service

After the DApp web application system is successfully built, testing will be carried out to determine the quality of service and data integrity. In this test using 3 nodes that are interconnected using the IPFS Cluster Private Network. To do the test, the test scenarios performed are as follows.

a. Quality of Service-1 testing scenario

This test scenario is carried out using 3 nodes connected via IPFS Cluster private network. 1 node performs the process of uploading data using the DApp web system that integrates

IPFS on the smart contract Ethereum, while the other 2 nodes carry out the process of receiving files.

b. Quality of Service-2 testing scenario

This test scenario is carried out using 3 nodes connected via IPFS Cluster private network. 2 nodes make the process of uploading data using the DApp web system that integrates IPFS on the smart contract Ethereum, while the other 3 nodes carry out the process of receiving files.

c. Quality of Service-3 testing scenario

Vol. 5 No.1
ISBN: 978-623-7144-28-1

This test scenario is carried out using 3 nodes connected via IPFS Cluster private network. 3 nodes carry out the process of uploading data and receiving using the DApp web system that integrates IPFS on the smart contract Ethereum.

The following are the results of testing of quality of service.

Table. 4: Quality of Service Testing Result.

Files Size	Quality of Service-1			Quality of Service-2			Quality of Service-3		
	Throughput (Kbps)	Packet Loss (%)	Delay (ms)	Throughput	Packet Loss	Delay (ms)	Throughput (Kbps)	Packet Loss (%)	Delay (ms)
10MB	19.61	0.00	56.79	29.84	0.13	39.85	20.19	1.08	60.00
50MB	36.36	0.78	30.09	130.71	0.91	10.37	68.16	0.08	17.60
100MB	93.04	0.22	12.08	112.15	0.46	11.66	90.13	0.48	13.22
200MB	68.35	2.75	16.78	68.00	0.83	18.70	78.70	0.43	15.04
300MB	82.34	1.76	13.76	121.27	2.03	9.64	96.87	0.25	11.67
400MB	108.21	2.72	10.62	66.76	1.34	20.73	82.78	0.57	14.33
500MB	88.53	2.27	12.42	53.95	2.54	24.20	80.06	0.65	14.53
600MB	65.40	2.10	17.58	78.76	1.97	15.89	84.89	0.29	13.39
700MB	55.49	2.70	21.04	64.72	1.97	17.71	83.42	0.28	14.09
800MB	49.12	2.35	23.51	59.25	1.56	22.96	67.96	2.21	16.98
900MB	43.50	2.86	26.72	16.75	1.75	76.36	89.89	0.90	12.98
1GB	22.36	2.11	50.11	60.02	2.96	23.32	79.86	1.16	14.45
1.1GB	57.35	2.01	20.14	56.91	1.22	23.38	82.84	1.69	16.19
1.2GB	44.32	2.11	25.98	55.37	1.84	21.39	77.13	2.54	18.45
1.3GB	44.64	2.04	25.84	77.80	2.68	17.91	103.17	1.33	11.34
1.4GB	34.58	2.25	32.80	47.23	1.75	24.83	102.04	1.78	11.98
1.5GB	45.69	1.70	25.07	19.23	0.92	60.97	66.37	2.31	17.92
Total	958.91	32.72	421.35	1118.72	26.85	439.87	1354.48	18.04	294.17
Average	56.41	1.92	24.79	65.81	1.58	25.87	79.68	1.06	17.30

This result of the Quality of Service-1 test scenario file upload is only done by 1 node and data is received by 2 nodes with a throughput value of 56.41 Kbps. Whereas in the Quality of Service-2 testing scenario file uploads are carried out 2 nodes and file receipts by 3 nodes, thus causing an increase in the throughput value of the Quality of Service-1 test scenario of 65.81 Kbps. Furthermore, in the Quality of Service-3 testing scenario file uploads and file receipts are carried

Vol. 5 No.1
ISBN: 978-623-7144-28-1

out by 3 nodes, causing an increase in the throughput value of the Quality of Service-1 and Quality of Service-2 scenarios of 79.68 Kbps. Based on the standard or recommendations of Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON), the average results of recapitulation of calculation of throughput value get results with index 1 in the category "Bad".

The results of the Quality of Service-1 test scenario file upload is only done by 1 node and data is received by 2 nodes with a packet loss value of 1.92%. Whereas in the Quality of Service-2 testing scenario file uploads are carried out 2 nodes and file receipts by 3 nodes, thus causing a decrease in the value of packet loss from the Quality of Service-1 test scenario of 1.58%.

Furthermore, in the Quality of Service-3 testing scenario, file uploads and file receipts are carried out by 3 nodes, causing a decrease in the value of packet loss from the Quality of Service-1 and Quality of Service-2 test scenarios of 1.06%. Based on the standards or recommendations of Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) the average results of the recapitulation of the calculation of packet loss value get results with index 4 with the category "Very Good".

The results of the Quality of Service-1 test scenario file upload is only done by 1 node and data is received by 2 nodes with a delay value of 24.79 ms. Whereas in the Quality of Service-2 testing scenario file uploads are carried out 2 nodes and file receipts by 3 nodes, thus causing an increase in the delay value of the Quality of Service-1 test scenario of 25.87 ms. Furthermore, in the Quality of Service-3 test scenario the file upload and file reception is done by 3 nodes, causing a decrease in the delay value of the Quality of Service-1 and Quality of Service-2 test scenarios by 17.30 ms. Based on the standards or recommendations of Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON), the average results of the recapitulation of the calculation of delay values obtain results with index 4 in the category of "Very Good".

4.2. Analysis Data Integrity

In this test using 3 nodes that are interconnected using the IPFS Cluster Private Network. The following are the results of data integrity.

Table. 5: Data Integrity Testing Result

File Name	File Content	Original IPFS Hash	IPFS Hash Download	File Content After Downloaded
*Integritas Data*1	*Ini Ibu* Dicky	QmargdpNa7wafb7pSgUvG6c8Y7	QmargdpNa7wafb7pSgUvG6c8Y7	*Ini Ibu* Dicky

Vol. 5 No.1
ISBN: 978-623-7144-28-1

		G2Wuf2SGbsGUJAjcAgfm	G2Wuf2SGbsGUJAjcAgfm	
*Integritas Data*1	*Ini Bapak* Dicky	QmNYwgt4WViw2tUBYqDM G3v jXWRyFqX1SiQDXuCMdb81n s	QmNYwgt4WViw2tUBYqDM G3v jXWRyFqX1SiQDXuCMdb81n s	*Ini Bapak* Dicky
*Integritas Data*2	*Ini Bapak* Zakky	QmXcnThoK4VykkZh7MFnzW N MJNPRDmSrQEbPnf5EQ8VW B8	QmXcnThoK4VykkZh7MFnzW N MJNPRDmSrQEbPnf5EQ8VW B8	*Ini Bapak* Zakky
*Integritas Data*2	*Ini Ibu* Zakky	QmVttcCDNMCQSjToWgBWV Gi DnwL8XuUNqLJqL9WnWueAt c	QmVttcCDNMCQSjToWgBWV Gi DnwL8XuUNqLJqL9WnWueAt c	*Ini Ibu* Zakky
*Integritas Data*3	*Ini Ibu* Jafar	QmXRpsF9sdEdy4WPnWXMgb M DFG5M2MCVxVnocyPgXZC1H b	QmXRpsF9sdEdy4WPnWXMgb M DFG5M2MCVxVnocyPgXZC1H b	*Ini Ibu* Jafar
*Integritas Data*3	*Ini Bapak* Jafar	QmXgP5fjaS48KC682pUu6e34 N MjeWTsyY2TjhDNxD3FTy4	QmXgP5fjaS48KC682pUu6e34 NM jeWTsyY2TjhDNxD3FTy4	*Ini Bapak* Jafar
Integritas Data A	*Ini Bapak* Prabowo	QmVLshXRbwnGYJYQFU2gA Zci tFP2YymaXcciCjgtzqZmM3	QmVLshXRbwnGYJYQFU2gA Zci tFP2YymaXcciCjgtzqZmM3	*Ini Bapak* Prabowo
Integritas Data B	*Ini Bapak* Prabowo	QmVLshXRbwnGYJYQFU2gAZ ci tFP2YymaXcciCjgtzqZmM3	QmVLshXRbwnGYJYQFU2gAZ ci tFP2YymaXcciCjgtzqZmM3	*Ini Bapak* Prabowo
Integritas Data C	*Ini Bapak* Jokowi	Qmbu3e5W9nHvktwrCiDs86w DN YX7DDoDS4sWZyytWWazD4	Qmbu3e5W9nHvktwrCiDs86w DN YX7DDoDS4sWZyytWWazD4	*Ini Bapak* Jokowi
Integritas Data D	*Ini Bapak* Jokowi	Qmbu3e5W9nHvktwrCiDs86 wDN YX7DDoDS4sWZyytWWazD4	Qmbu3e5W9nHvktwrCiDs86 wDN YX7DDoDS4sWZyytWWazD4	*Ini Bapak* Jokowi
Integritas Data E	*Ini Bapak* Sandiaga	QmZ1kmpJURcgibhWZ7AoaJ5 67n TvY8S5a6MovncL5pFLuM	QmZ1kmpJURcgibhWZ7AoaJ56 7n TvY8S5a6MovncL5pFLuM	*Ini Bapak* Sandiaga
Integritas Data F	*Ini Bapak* Sandiaga	QmZ1kmpJURcgibhWZ7AoaJ5 67n TvY8S5a6MovncL5pFLuM	QmZ1kmpJURcgibhWZ7AoaJ56 7n TvY8S5a6MovncL5pFLuM	*Ini Bapak* Sandiaga

The data integrity test the same results have the IPFS hash and IPFS hash downloaded, this proves that the authentication process is successful because the data integrity in each file does not change from the beginning of the data transmission process to completion. The results of the IPFS hash and IPFS hash downloads are different even though they have the same file name, this proves that the data obtained by the IPFS hash only reads based on the metadata of a file. In the

Vol. 5 No.1
ISBN: 978-623-7144-28-1

authentication process it works because the data integrity in each file does not change from the beginning of the data transmission process to completion.

Because IPFS reads files during the process of sending data using the metadata of the file to be sent, so the possibility of redundant data is very small. Changes in data integrity cannot be changed because IPFS sends data through content hash that has been very well encrypted during the data transmission process, so that intruders or intruders have difficulty making data changes because they have a public key and private key that must be owned by intruders. The analysis of this test is to use IPFS to maintain data integrity more reliably, because IPFS is able to maintain content files and IPFS hashes during the process of uploading and downloading data. The test fulfills aspects of information security based on Confidentiality, Integrity, and Availability

5. CONCLUSION

The average value of throughput on the quality of service-1, quality of service-2, quality of service-3, testing results is 56.41 Kbps, 65.81 Kbps and 79.68 Kbps, for average packet loss values of 1.92%, 1.58% and 1.06%, while the average value of delay is 24.79 ms, 25.87 ms and 17.30 ms with the average value of the Quality of Service index which is 3 which meets the "Satisfying" category based on THIPON standards. The average value of throughput is not absolute and can be influenced by many factors and conditions of networks and devices during the process of sending and receiving data. The results of all index averages with parameters throughput, packet loss, delay, on the Quality of Service test get results with index value 3 with the category "Satisfying". Based on the results of the tests carried out, data integrity with hash parameters on IPFS has been tested if the file name is changed in such a way that IPFS maintains the authenticity of the file content. if the content file changes, the hash obtained from IPFS is different even though the file name is the same. Thus it can be concluded that using IPFS to maintain data integrity can be relied upon because there is no change in the changes during the testing, so that the results have met the aspects of information security based on Confidentiality, Integrity, and Availability.

REFERENCE

Atzei N., Bartoletti M., & Cimoli, T. (2017) : A Survey of Attacks on Ethereum Smart contracts. International Conference on Principles of Security and Trust hal. 164-186.

Bennet, J. (2014) : IPFS - Content Addressed, Versioned, P2P File System(DRAFT 3). ArXiv.

Vol. 5 No.1
ISBN: 978-623-7144-28-1

Chen, Yongle, dkk. (2017) : An Improved P2P File System Scheme based on IPFS and Blockchain. IEEE International Conference on Big Data.

Cisco, Internetworking. (2016): Internetworking Technology Handbook, http://docwiki.cisco.com/wiki/Internetworking_Technology_Handbook. Accessed 1 December 2018.

CompTIA+. (2014) : CompTIA Security+ SY0-40I Official Study Guide Student Editon. London: gtslearning.

Galal, H.S dan Youssef, A.M. (2018,3) : Verifiable Sealed-Bid Auction on the Ethereum Blockchain. Conference of Finanacial Cryptography 2018, Curacao.

Sinha, P. dan Kaul, A. (2018) : Decentralized KYC System. International Research Journal of Engineering and Technology (IRJET), 5(8), 1209-1210.

Wennergen, O., dkk. (2018) : Tranparency Analysis Of Distributed File System With a Focus on Interplanetary File System, Swedia: University of Skovde.

Wood, G. (2017) : Ethereum: A Secure Decentralised Generalised Transaction Ledger EIP-150 Revision.