

## DAFTAR PUSTAKA

- [1] Rajarshi Mitra, "Complete Guide to Big Data and Blockchain", <https://blockgeeks.com/guides/big-data-and-blockchain/#What is a Blockchain>, 24 April 2020, Akses terakhir 10 September 2021.
  - [2] Deepa S. Kumar, M. Abdul Rahman, "Simplified HDFS Architecture with Blockchain Distribution of Metadata", 2017, *International Journal of Applied Engineering Research*, Vol. 12, PP. 11374-11382.
  - [3] Scott Ross Brisbane, "Decentralising Big Data Processing", *School of Computer Science and Engineering Faculty of Engineering The University of New South Wales*.
  - [4] Franklin John, Suji Gopinath dan Elizabeth Sherly, "A Decentralised Framework for Efficient Storage and Processing of Big Data Using HDFS and IPFS", 2021 *International Journal of Humanitarian Technology*, Vol 1, No.2, pp 131-143, doi: 10.1504/IJHT.2020.112451.
  - [5] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, Samir Belfkih, "Big Data Technologies : A Survey", 2018 *Journal of King Saud University - Computer and Information Sciences*, pp 1319-1578, doi: 10.1016/j.jksuci.2017.06.001.
  - [6] Seref Sagiroglu and Duygu Sinanc, "Big Data : A Review", 2013, *IEEE, Gazi University-Departement of Computer Engineering, Faculty of Engineering*.
  - [7] Roni Setiawan, "Apa Itu Big Data ? Mengapa Itu Penting", 2021 - <https://www.dicoding.com/blog/apa-itu-big-data>, Akses terakhir 15 Juni 2023 .
  - [8] Delis Permatasari, Mohammad Dani, M.T., Mia Rosmiati, S.Si., M.T., "Analisis Performansi Hadoop Cluster Multi Node pada Komputasi Dokumen", *Program Studi Teknik Komputer Peminatan Teknik Jaringan Politeknik Telkom Bandung*.
  - [9] Imam Cholissodin, Efi Riyandani, "Analisis Big Data (Teori & Aplikasi)", 2018 , Vol. 1. 01, *Fakultas Ilmu Komputer*.
  - [10] Qihong Zheng, Yi Li, Ping Chen, "An Innovative IPFS-Based Storage Model for Blockchain" 2018 - *IEEE ACM International Conference on Web Intelligence (WI)*, doi: 10.1109/WI.2018.000-8.
  - [11] Fabian Knirsch, Andreas Unterweger and Dominik Engel, "Implementing a Blockchain from Scratch: why, how, and what we learned", 2019 - *EURASIP Journal on Information Security*, doi: 10.1186/s13635-019-0085-3.
  - [12] Dylan Yag, Peter Mell, Nik Roby, Karen Scarfone, "Blockchain Technology Overview", 2018 *National of Standards and Technology*, doi: 10.6028/NIST.IR.8202.
  - [13] Satoshi Nakamoto, "Bitcoin: A Peer-toPeer Electronic Cash System", 2018, [aring] tersedia pada : <https://bitcoin.org/bitcoin.pdf>.
- ad Saleh, "Blockchain Without Waste: Proof-of-Stake", 2018 *SSRN Electronic Journal*, doi: 10.2139/ssrn.3183935.
- seno, Taufiq Rizky Darmawan, "Sistem E-Transkrip Berbasis Blockchain dan



- IPFS untuk Menjaga Integritas Data Akademik", 2022 *Universitas Komputer Indonesia*.
- [16] Juan Benet, "IPFS - Content Addressed, Versioned, P2P File System (Draft 3)", 2014 [Daring] tersedia pada : arXiv:1407.3561v1.
  - [17] IPFS, "How IPFS Works", [Daring] tersedi pada : <https://docs.ipfs.tech/concepts/how-ipfs-works/#subsystems-overview>, Akses terakhir 15 Juni 2023 .
  - [18] Weiqin Zou, David Lo, Pavneet Singh Kochhar, Xuan-Bach Dinh Le, Xin Cia, Yang Feng, Zhenyu Chen, Baowen Xu, "Smart Contract Development: Challenges and Opportunities", 2019 *IEEE*, doi: 10.1109/TSE.2019.2942301.
  - [19] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xianping Chen, Jian Weng, Muhammad Imran, "An Overview on Smart Contract: Challenges, Advances, and Platforms", 2019 *Futere Generation Computer System*, doi: 10.1016/j.future.2019.12.019
  - [20] MetaMasks Blockchain Mobile App Opens Doors For Next Level Web. <https://www.bloomberg.com/news/articles/2020-09-02/metamask-s-blockchain-mobile-app-opens-doors-for-next-level-web>. Akse terakhir Mei 2023.
  - [21] Patel Neha M, Patel Narendra, Mosin I Hasan, Shah Parth D, Patel Mayur M. "Improving HDFS Write Performance Using Efficient Replica Placement". 2014 *International Conference – Confluence The Next Generation Information Technology Summit*, doi: 10.1109/CONFLUWNCE.2014.6949234.
  - [22] Tian ma, Feng Tian, Bo Dong, "Ordniat Optimization-Based Performance Model Estimation Method for HDFS," 2020 *IEEE Access*, Vol. 8.
  - [23] Haiyang Yu, Qi Hu, Zhen Yang, Huan Liu, "Efficient Continuous Big Data Integrity Checking for Decentralitation Storage," 2021 *IEEE Trasaction on Network Science and Engineering*, vol. 8, no. 2, pp. 1658-1673, 1 April-June 2021, doi: 10.1109/TNSE.2021.3068261 .
  - [24] Wenjuang Cheng, Miaomiao Zhou, Bing Tong, Junhong Zhu, "Optimizing Small File Storage Process of the HDFS Which Based on the Indexing Mechanism," 2017 *the 2nd IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Chengdu, China, pp. 44-48, doi: 10.1109/ICCCBDA.2017.7951882.
  - [25] Gaurav Shrivastava, Sachin Patel, "Secure Storage and Data Sharing Scheme Using Private Blockchain-Based HDFS Data Storage for Cloud Computing" 2023 *International Journal of Computer Network and Applications (IJCNA)*, Volume 10, doi: 10.22247/ijcna/2023/218509.
  - [26] Jasmine Sethi, Shashank Srivastava, and Divya Upadhyay, "A Review on P2P File System Based on IPFS for Concurrency Control in Hadoop", 2021 *Emerging Technologies in Data Mining and Information Security*. doi: 10.1007/978-981-15-9927-9\_83



ut Viorel Gingu, Daniel Hursan, "IPFS: Dencetralized Storage in a Centralized orld", 2022 *IEEE Blockchain Technical Briefs - 2022*.

- [28] Omar Abdullah Lajam, Tarek Ahmed Helmy, "Performance Evaluation of IPFS in Private Networks", *2021 4<sup>th</sup> International Conference on Data Storage and Data Engineering (DSDE '21) - 2022*. doi: 10.1145/3456146.3456159 .
- [29] N. Ahmed, A. L. C. Barczak, S. U. Bazai, T. Susnjak and M. A. Rashid, "Performance Analysis of Multi-Node Hadoop Cluster Based on Large Data Sets," *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, Gold Coast, Australia, 2020, pp. 1-6, doi: 10.1109/CSDE50874.2020.9411587.
- [29] Wang Junmei, Wu Jihong, "Research on Performance Automation Testing Technologi Based on Jmeter" *2019 IEEE International Conference on Robots & Intelligent System (ICRIS)*, Haikou, China, 2019, doi: 10.1109/ICRIS.2019.00023.
- [30] MySkill.id, 2023 "Correlation Analysis" *IS*), Online at <https://medium.com/@myskill.id/correlation-analysis-b39b9ff96c86>, accessed 04 Mei 2024



## LAMPIRAN

### Lampiran 1 Source code Smart Contract (storehash.sol)

```

1. pragma solidity >=0.4.21 <8.10.0;
2.
3. contract Contract {
4.     string ipfsHash;
5.
6.     function sendHash(string x) public {
7.         ipfsHash = x;
8.     }
9.
10.    function getHash() public view returns (string x) {
11.        return ipfsHash;
12.    }
13. }
14.

```

### Lampiran 2 Source code DApps (App.js)

```

1. import { Table, Grid, Button, Form } from 'react-bootstrap';
2. import React, { Component } from 'react';
3. //import logo from './logo.svg';
4. import './App.css';
5. import web3 from './web3';
6. import ipfs from './ipfs';

```

```

: storehash from './storehash';

```

```

App extends Component {

```



```

11. state = {
12.   ipfsHash:null,
13.   buffer:",
14.   ethAddress:",
15.   blockNumber:",
16.   transactionHash:",
17.   gasUsed:",
18.   txReceipt: "
19. };
20.
21. captureFile =(event) => {
22.   event.stopPropagation()
23.   event.preventDefault()
24.   const file = event.target.files[0]
25.   let reader = new window.FileReader()
26.   reader.readAsArrayBuffer(file)
27.   reader.onloadend = () => this.convertToBuffer(reader)
28. };
29.
30. convertToBuffer = async(reader) => {
31.   //Konversi file ke Buffer
32.   const buffer = await Buffer.from(reader.result);
33.   //set buffer-menggunakan es6
34.   this.setState({buffer});
35.   console.log({buffer});
36. };
37.
38. onClick = async () => {
39.
40.   try{
41.     this.setState({blockNumber:"waiting.."});
42.     this.setState({gasUsed:"waiting..."});

```

/ Mendapatkan Transaction Receipt pada konsol



```

46.     await web3.eth.getTransactionReceipt(this.state.transactionHash, (err,
txReceipt)=>{
47.         console.log(err,txReceipt);
48.         this.setState({txReceipt});
49.     }); //menunggu getTransactionReceipt
50.
51.     await this.setState({blockNumber: this.state.txReceipt.blockNumber});
52.     await this.setState({gasUsed: this.state.txReceipt.gasUsed});
53. } //try
54. catch(error){
55.     console.log(error);
56. } //catch
57. } //onClick
58.
59. onSubmit = async (event) => {
60.     event.preventDefault();
61.
62.     //mendapatkan akun metamask
63.     const accounts = await web3.eth.getAccounts();
64.
65.     console.log('Sending from Metamask account: ' + accounts[0]);
66.
67.     //mendapatkan contract address dari storehash.js
68.     const ethAddress= await storehash.options.address;
69.     this.setState({ethAddress});
70.
71.     //menyimpan data ke IPFS
72.
73.     await ipfs.add(this.state.buffer, (err, ipfsHash) => {
74.         console.log(err,ipfsHash);
75.         //setState setting ipfsHash ke ipfsHash[0].hash
76.         this.setState({ ipfsHash:ipfsHash[0].hash });
77.
78.         storehash.methods.sendHash(this.state.ipfsHash).send({
79.             from: accounts[0]

```



```

80.     }, (error, transactionHash) => {
81.         console.log(transactionHash);
82.         this.setState({transactionHash});
83.     }); //storehash
84.     }) //await ipfs.add
85. }; //onSubmit
86.
87. render() {
88.
89.     return (
90.         <div className="App">
91.             <header className="App-header">
92.                 <h1>IPFS & Blockchain Big Data DApps</h1>
93.             </header>
94.
95.             <hr />
96.
97.             <Grid>
98.                 <h3>Choose file to send to IPFS </h3>
99.                 <Form onSubmit={this.onSubmit}>
100.                    <input
101.                        type = "file"
102.                        onChange = {this.captureFile}
103.                    />
104.                    <Button
105.                        bsStyle="primary"
106.                        type="submit">
107.                            Send it
108.                    </Button>
109.                </Form>
110.
111.                <hr/>
112.                <Button onClick = {this.onClick}> Get Transaction Receipt </Button>
113.
114.                <Table bordered responsive>

```



```

115.     <thead>
116.         <tr>
117.             <th>Tx Receipt Category</th>
118.             <th>Values</th>
119.         </tr>
120.     </thead>
121.
122.     <tbody>
123.         <tr>
124.             <td>IPFS Hash # stored on Eth Contract</td>
125.             <td>{this.state.ipfsHash}</td>
126.         </tr>
127.         <tr>
128.             <td>Ethereum Contract Address</td>
129.             <td>{this.state.ethAddress}</td>
130.         </tr>
131.
132.         <tr>
133.             <td>Tx Hash # </td>
134.             <td>{this.state.transactionHash}</td>
135.         </tr>
136.
137.         <tr>
138.             <td>Block Number # </td>
139.             <td>{this.state.blockNumber}</td>
140.         </tr>
141.
142.         <tr>
143.             <td>Gas Used</td>
144.             <td>{this.state.gasUsed}</td>
145.         </tr>
146.     </tbody>
    </Table>
</Grid>
div>

```



```

150.   );
151.   } //render
152. }
153.
154. export default App;
155.

```

### Lampiran 3 Source code DApps (storehash.js)

```

1. const address = '0xd273c4a8dc9dbf467c69e38205586601efa414d0';
2. //ABI dari smart contract remix
3. const abi = [
4.   {
5.     "constant": false,
6.     "inputs": [
7.       {
8.         "name": "x",
9.         "type": "string"
10.      }
11.    ],
12.    "name": "sendHash",
13.    "outputs": [],
14.    "payable": false,
15.    "stateMutability": "nonpayable",
16.    "type": "function"
17.  },
18.  {
19.    "constant": true,
20.    "inputs": [],
21.    "name": "getHash",
22.    "outputs": [
23.      {
24.        "name": "x",
25.        "type": "string"
26.      }
27.    ]
28.  }
29. ]

```



```

27.         ],
28.         "payable": false,
29.         "stateMutability": "view",
30.         "type": "function"
31.     }
32. ]
33.
34.
35.
36. export default new web3.eth.Contract(abi, address);
37.

```

#### Lampiran 4 Source code koneksi ke IPFS (ipfs.js)

```

1. const ipfsApi = require('ipfs-api');
2. const ipfs = new ipfsApi('localhost', '5001', {protocol: 'http'});
3.
4. export default ipfs;
5.

```

#### Lampiran 5 Source code koneksi Web3 Metamask (web3.js)

```

1. import Web3 from 'web3';
2.
3. let web3
4. if(typeof window !== 'undefined' && typeof window.ethereum !== 'undefined'){
5.     //getting Permission to access
6.     window.ethereum.enable();
7.     web3 = new Web3(window.ethereum);
8.
9. }else if (typeof window !== 'undefined' && typeof window.web3 !== 'undefined') {
10.     web3 = new Web3(window.web3.currentProvider);
    legacy MetaMask accounts are always exposed
    {

```



```
14. alert("No MetaMask detected, please install MetaMask first");  
15. }  
16.  
17. export default web3;  
18.
```

