Terakreditasi SINTA Peringkat 2
Surat Keputusan Direktur Jenderal Pendidikan Tinggi, Riset, dan Teknologi, Nomor: 158/E/KPT/2021
masa berlaku mulai Volume 5 Nomor 2 Tahun 2021 sampai Volume 10 Nomor 1 Tahun 2026

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 2 (2022) 275 - 280 ISSN Media Elektronik: 2580-0760

Designing an Ethereum-based Blockchain for Tuition Payment System using Smart Contract Service

Moch Sholeh[1], Esther Yolanda Talahaturuson[2], Maulana Rizqi[3], Agustinus Bimo Gumelar[4]
[1,2,3,4]Fakultas Ilmu Komputer, Universitas Narotama, Surabaya
[1]moch.sholeh@fik.narotama.ac.id, [2]estheryola99@gmail.com, [3]maulana.rizqi@narotama.ac.id, [4]bimogumelar@ieee.org*

Abstract

The security and disclosure of information in online transaction data remains a sensitive subject until this day. Whenever a collection of data from a transaction process is accessible over the internet system, certain parties may misuse any one of these data. Blockchain technology, which includes the use of smart contracts, is thought to be capable of overcoming this problem due to the blockchain's decentralized and distributed nature. Blockchain allows transaction data to be accessed openly and transparently while still being securely protected by hashing encryption owned by smart contracts. This enables users to have detailed access privileges to each transaction's data. The development of smart contracts will be carried out in the production of microservices payment gateways based on decentralized apps (DApps) on the Ethereum blockchain in this research, with the payment gateway generated being used in the tuition payment process. The Truffle framework and the Metamask wallet will be used to assist the Ethereum payment process during the DApps development process. The results of testing the functionality of each smart contract feature reveal that the payment system can be utilized effectively and that there are no issues that cause transaction failures.

Keywords: *Blockchain, Ethereum, Smart Contract, Payment Gateway*

1. Introduction

Indonesia is the one of the countries with the most growing users on online payment system (named payment gateway) [1][2]. At least the process of e-commerce transactions in Indonesia has increased to 38.3% in 2020 [3]. Payment gateway systems is applicable for every sectors, including in the academia sector. Online-based tuition payment usually done using respective university's academic information system.

According to the findings of a Pricewaterhouse Coopers survey performed in 2017, information technology security is the most difficult challenge in the field of finance technology [4]. Any existing data will be entered into the internet system, and there is a risk of data misuse, whether intentional or inadvertent [5]. In 2020, a group of hackers managed to sell 1.2 million pieces of data from e-commerce consumers of Bhinneka.com on a dark web in order to profit from the transaction [3]. With the transaction process being made easier, the issue of user data security remains an important point to consider. A series of transactions, regardless of the form of the transaction process, will still contain sensitive information, and the nature of the centralized payment gateway makes misuse of private data quite likely. This issue can be solved by employing blockchain technology [6], which employs hashes, cryptographic algorithms, and public key certificates [7].

Since its initial launch in 2010, blockchain has remained the main topic of discussion in a variety of fields to this day. The blockchain concept was initially applied to digital currencies, but it may also be employed in a variety of industries alongside advancements, one of which is payment gateways [6].

Rahardja et al. previously applied blockchain technology in an e-commerce business to guarantee the secrecy and security of data in transactions [8]. Their study was successful in distributing transaction data between sellers and buyers by storing data in blocks with a specific code. With the use of blockchain technology in the purchase transaction process, clarity of information regarding transaction details from the date of the transaction to the nominal generated from the transaction can be seen [8].

Diterima Redaksi: 04-03-2022 | Selesai Revisi: 21-04-2022 | Diterbitkan Online: 29-04-2022

Darlen et al. did another study presenting blockchain technology and its use in credit card payments [6]. According to Darlen et al., there are four stages involved in transferring payment transaction information, including card holder, fund verification, card requirements, and payment authorisation. There is a high danger of data misuse at each information exchange point. The usage of blockchain is thought to be capable of reducing the number of validation points by allowing payment card information to be encrypted data on transaction data, allowing for less misuse of transaction data and minimizing processing expenses [6].

Shee-Ihn Kim and Seung-Hee Kim [7] conducted similar research in proposing an electronic payment platform with blockchain as its key component. In the authentication process, the suggested payment system has three components: seller, buyer, and blockchain. The seller's and buyer's digital signatures will be confirmed on the blockchain using the verification function [7]. According to past research, the blockchain system is transparent and decentralized [9][10]. As a response, in this study, a smart contract on the Ethereum blockchain was created for the production of decentralized tuition payment gateway microservices at universities.

The rest of this paper is organized as follows: Section 1 provides an overview of the research problem and its context. Section 2 discusses the methodology that will be used in this study. Section 3 presents the results of the method's implementation. Finally, Section 4 discusses the conclusions that can be derived from this study's findings.

2. Research Methods

The Truffle framework and the Solidity programming language were used to create smart contracts in this study. Tokens and smart contract wallets will be used in transactions by Metamask, which depicts Ethereum as a digital currency. Finally, the smart contract testing procedure will employ the Infura Testnet in conjunction with the Metamask wallet [11][12]. Figure 1 depicts the research flow, which will be discussed in depth in this section.

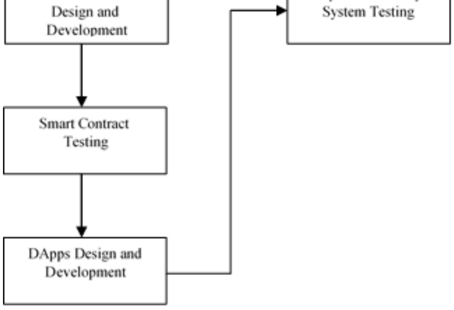

Smart Contract Design and Development
Smart Contract Testing
DApps Design and Development
Payment Gateway System Testing

Figure 1. Our Proposed Methodology

2.1 Smart Contract Design and Development

It is vital to choose the type of blockchain that will be used during implementation before designing a smart contract. Blockchain is defined as a distributed database with transaction records linked by a network. Figure 2 depicts the workings of the blockchain, where all distributed data is validated by all nodes and if there are data differences between networks, other networks will prevent the data from being processed.

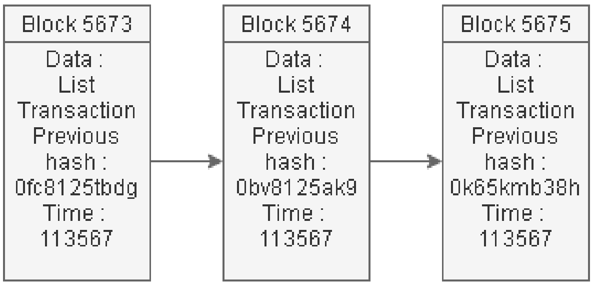

Block 5673
Data : List Transaction Previous hash : 0fc8125tbdg Time : 113567
Block 5674
Data : List Transaction Previous hash : 0bv8125ak9 Time : 113567
Block 5675
Data : List Transaction Previous hash : 0k65kmb38h Time : 113567

Figure 2. Blockchain Workflow

There are already several types of blockchain, including: (1) public blockchain, where anyone can become a user, execute nodes, and make transactions, and all nodes can participate by consensus as a determination of data validation. Else is (2) Consortium Blockchain, which is the inverse of Blockchain Public in that only members of the consortium can run full nodes and make transactions. The last is (3) private blockchain, in which the process is solely managed by an enterprise, resulting in a single trust domain. [13][14][15]. Based on numerous considerations regarding the nature of each blockchain, the Ethereum public blockchain was eventually chosen as the development platform, one of which being that every transaction that occurs will be logged in a ledger and can be examined via the etherscan transaction history.

The Ethereum Blockchain is a decentralized network that records every transaction using a consensus technique known as Proof of Work (PoW) [16]. PoW is used to ensure that only confirmed transactions are recorded in the blockchain [17][18]. This consensus protocol necessitates that the calculated value be equal to or less than a provided value [19]. As a result, the number of valid blocks can be calculated from each value that fits the requirements [8][20]. When one node reaches the goal value, it will communicate the block to the other nodes, and other nodes must validate the truth of the value they have [19]. Figure 3 depicts the usual flow of transactions that occur on Ethereum.

In this situation, the Ethereum blockchain functions as a virtual machine, running a smart contract peer to peer with the Ethereum digital currency. Smart contract development is carried out using the Solidity programming language, which runs on the Ethereum Virtual Machine (EVM) to run bytecode contracts. Consequently, the smart contract code is separated from the host computer's file system, network, or other activities. The smart contract code compiler process in the EVM is depicted in Figure 4. Meanwhile, Figure 5 depicts the flow of creating a contract on remix.ethereum.org with the name smart contract deposit.

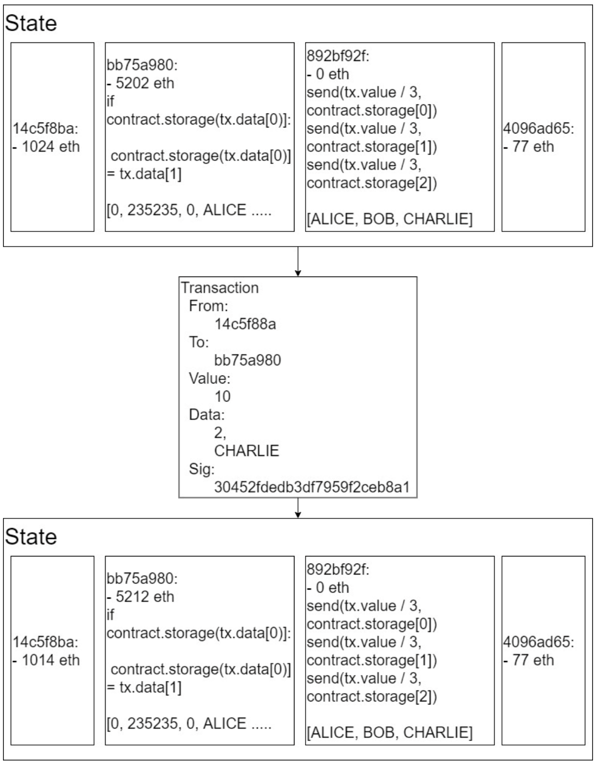


Figure 3. Ethereum Workflow

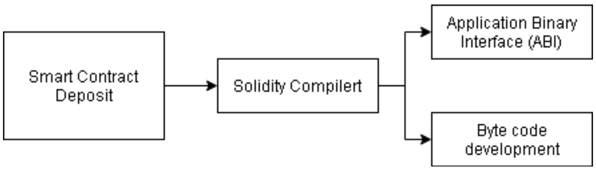


Figure 4. Smart Contract Compiler Workflow

The smDeposit class contract in the deposit smart contract includes four functions: deposit, myDeposit, myCounter, and withdraw. The contract definition, on the other hand, is not code that is run on the Ethereum network, but will be separated into two contract definition files during the compilation process using the Solidity Compiler. The first file will include the bytecode, which is the bytecode that will actually be deployed on the Ethereum network. The second file is the Application Binary Interface (ABI), which is illustrated in Figure 6 through a flow diagram. This ABI file will be the user layer for interacting with smart contracts via DApps web applications.

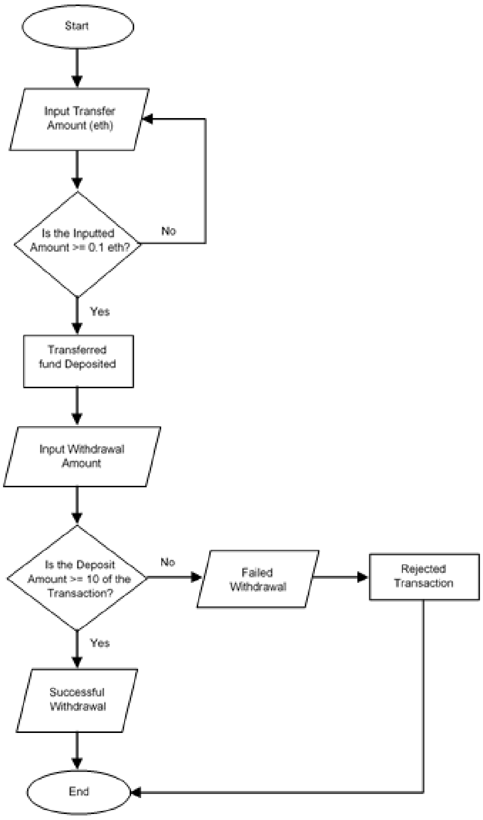

Start
Input Transfer Amount (eth)
Is the Inputted Amount >= 0.1 eth?
No
Yes
Transferred fund Deposited
Input Withdrawal Amount
Is the Deposit Amount >= 10 of the Transaction?
No
Failed Withdrawal
Rejected Transaction
Yes
Successful Withdrawal
End

Figure 5. Smart Contract Deposit Workflow

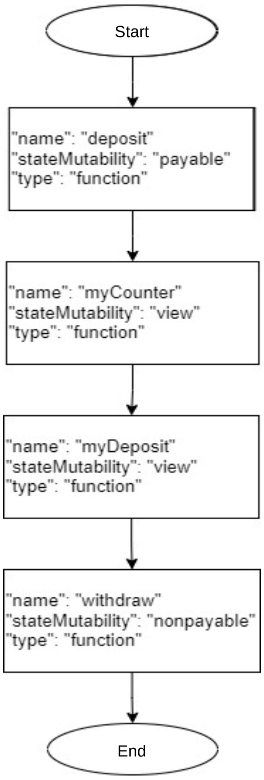


Figure 6. ABI *Smart Contract* Deposit Workflow

DOI: https://doi.org/10.29207/resti.v6i2.3917

Table 1. Our Smart Contract's Functions and Variables

Items	Types	Usage
PaymentSystem	Function	To create smart contract
StudentDetail	Function	To create the details of students
withdraw	Function	To withdraw funds
generateTuition	Function	To generate students' bills
pay	Function	To make payments
getInvoic	Function	To check invoices
editStudentProfile	Function	To edit the details of students

At this point, various functions that will be applied to smart contracts are being developed, including: issuing new bills by administrators, payment process (by student), checking bills, verifying payment information, and withdrawing. Some of these elements are adequate to illustrate the payment gateway's mandatory features. The variables utilized for each feature used in the development of smart contracts are detailed in Table 1.

2.2 Testing the Smart Contract

Remix Ethereum is a web or desktop platform from Ethereum that is designed to assist testing of developed smart contracts, and the use of Remix in this study is meant as a simulation of blockchain principles on local Ethereum by executing transactions utilizing developed features. This testing process ensures that each script, which is made of each feature, is operating in accordance with the smart contract logic.

2.3. DApps Design and Development

Truffle is currently being utilized as a framework for constructing the DApps payment gateway interface. Each variable used in the development of DApps is detailed in depth in Table 2.

Table 2. Functions and Variables of DApps Payment Gateway

Items	Types	Usage
App	Function	To create a DApps project
resetForms	Constant	To reset students' data entry
editProfile	Constant	To edit students' details
generateTuition	Constant	To generate students' bills
pay	Constant	To make payments
getInvoic	Constant	To check invoices
getDetail	Constant	To get students' details

In this study, tuition transactions via DApps will necessitate the presence of an ERC token, which is a standardized token from Ethereum, with each ERC token on Ethereum serving a distinct purpose. Table 3 lists the many types of ERC tokens and their usage.

Based on the function of each token, the ERC-20 token was adopted in this study because students will utilize a crypto wallet token type in simulating financial payments, allowing its use for transaction purposes through the Testnet network. Tokens are created using Metamask, which is already linked to the smart contract payment gateway network.

Table 3. ERC Token Types

ERC Token	Types	Usage
ERC-20	Fungible token	Token wallet
ERC-721	Non-Fungible token	Representation of a physical asset
ERC-1155	Standar token ethereum hybrid	Decentralization in games

2.4 Testing the Payment Gateway System

We utilize Metamask for payment gateway testing once the development stage of smart contracts and DApps has been completed. When the connection between Metamask and Remix is established, a confirmation display containing the address of the transaction to be made and the amount of the fee to be paid will appear in the screen. Metamask also provides extensive information from the smart contract in encrypted hexadecimal format via the transaction address [21].

This testing stage differs from the testing stage performed by Remix. Remix testing focuses on each function of the features produced using Remix's Testnet network. While testing at this stage is designed to assess the viability of the payment gateway system, which is currently operational with a prototype system design at the start of development. Each user's feature interacts with the smart contract [11]. Consequently, at this level, testing is carried out by incorporating all of the features that are owned by users with access permissions such as administrators and students.

3. Results and Discussions

3.1 DApps-based Payment Gateway System

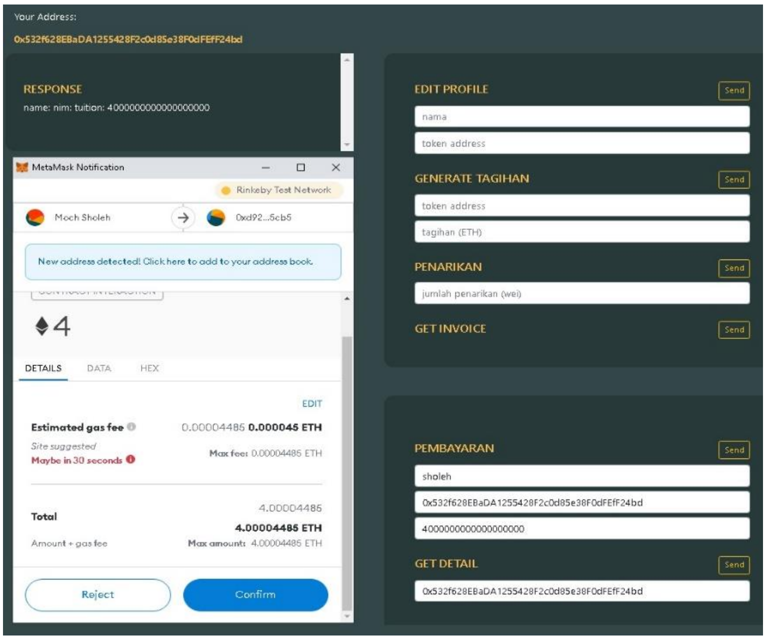
Tuition payment simulations were performed after completing the smart contract and application interface design stages, as shown in Table 4, where each token represents a transaction on DApps with the publicly accessible Testnet network.

The feature testing procedure begins with the user administrator creating invoices for student users, filling in the token for transaction reasons, and collecting the student's token address together with the nominal amount charged to the student. There were no issues discovered throughout this billing simulation.

After the billing procedure is completed by the administrator user, the student user will try to pay the bill. As illustrated in Figure 7, students make payments based on the number of bills displayed; if a disparity is discovered, the transaction is refused.

The value of Gwei and Gas will be indicated as additional costs that must be paid on the transaction confirmation. The testing procedure for each role that occurs in DApps is outlined in Table 5. The testing process validated that every feature on the DApps payment gateway works properly and that no difficulties were discovered.

DOI: https://doi.org/10.29207/resti.v6i2.3917



Your Address:
0x532f628EBaDA1255428F2c0d85e38F0dFEfF24bd

RESPONSE
name: nim: tuition: 4000000000000000000

MetaMask Notification
Rinkeby Test Network
Moch Sholeh → 0xd92...5cb5
New address detected! Click here to add to your address book.

4

Estimated gas fee 0.00004485 0.000045 ETH
Site suggested
Maybe in 30 seconds
Max fee: 0.00004485 ETH

Total 4.00004485
4.00004485 ETH
Amount + gas fee
Max amount: 4.00004485 ETH

PEMBAYARAN
sholeh
0x532f628EBaDA1255428F2c0d85e38F0dFEfF24bd
4000000000000000000

GET DETAIL
0x532f628EBaDA1255428F2c0d85e38F0dFEfF24bd

Figure 7. User Interface of Payment Billing

Table 4. ERC-20 Token Data

ID	Name	Token
1	Account 1	0x6b8a046130470099fEBB53d2112C8C33CC2F04Dd
2	Account 2	0x532f628EBaDA1255428F2c0d85e38F0dFEfF24bd
3	Account 3	0xa66DBcF56Bd7aa0Cbe3d1f9ab9c83Fbc1b068e86
4	Account 4	0x645e48E909f7666a17cA9762269B2834978d1742
5	Account 5	0x426e552f2032FE60B63F9C5CbBB587f829c4cf22
6	Account 6	0xf313b264009905E7Cf5F4B9bAd60b1B283F41837
7	Account 7	0x4bd14d77Fb668F44adDD7BB4A3507a76018B7Adb
8	Account 8	0x84876F815131D9c702F8BEdB395561e36D88f78E
9	Account 9	0x92f5d9AAB92940038872764f0Ce0A43BA8C4f6Bb
10	Account 10	0xc204811cdEfcF78ef9Ccf3f063cb26c2fB11d55A
11	Account 11	0xf91ce9fb31D4215Ab7B085da69a5bd801Da8786A
12	Account 12	0x54616164C7520a37597149E8b83711EE3BeA2baC

Table 5. Testing Procedure of DApps

Testing Steps	Role	Status
Entering Metamask	Administrator	Successful
Establishing connection with Testnet network	Administrator	Successful
Billings issuance by the administrator user	Administrator	Successful
Payments by students user based on the amount on the billing issued	Mahasiswa	Successful

3.2 Testing the System's Functionality

Testing the DApps payment gateway system's functionality for each feature established in this study. Table 6 explains the testing procedure in full as well as the results collected. Every feature of DApps was discovered to be functional.

Table 6. Testing the DApps System's Functionality

Test Case	Input & Output	Result
The administrator user fills out the names and token address of the respective student	Input: name (text), token (varchar) Output: the name in the profile page changed when the inputted token is valid	Successful
The administrator user creates the bill for the respective student	Input: token (varchar), eth bill (int) Output: the bill for the respective student's profile is adjusted accordingly when the inputted token is valid	Successful
Withdrawal	Input: amount (int) Output: the administrator user makes a withdrawal to the payment made by the respective student	Successful
The respective student make payment based on the amount shown in the respective bill	Input: name (text), token address (varchar), wei bill (int) Output: after the respective student's payment is successful , the respective bill is labelled as paid when the inputted token is valid	Successful
Checking whether or not there is still an outstanding bill for the respective students	Input: token (varchar) Output: get the list of the students with their respective outstanding bills based on their token	Successful

DOI: https://doi.org/10.29207/resti.v6i2.3917

4. Conclusion

The creation of smart contracts on the Ethereum blockchain plays a critical role in ensuring the security of payment gateway data. In addition to the open and transparent access to transaction history, all detailed transaction data is safeguarded by hashing and unchangeable.

The Testnet network test results confirmed that the transaction is successful and meets the each testing scenario. On the other hand, the quantity of Gwei and Gas that must be paid as a transaction fee is a separate concern in the DApps testing stage.

As a suggestion for future research, payment gateway smart contracts can be combined with a variety of other features that aid in the transaction process. Payment gateways can also be merged with other systems that share the same ecosystem in the transaction process.

References

[1] Y. Hu *et al.*, "A Delay-Tolerant Payment Scheme Based on the Ethereum Blockchain," *IEEE Access*, vol. 7, Mar. 2019, doi: 10.1109/ACCESS.2019.2903271.

[2] J. Zhang, A. Xu, M. Li, X. Huang, N. Xue, and Q. Sheng, "A Blockchain Based Micro Payment System for Smart Devices," *Int. J. Des. Anal. TOOLS INTERGRATED CIRCUITS Syst.*, 2016, [Online]. Available: http://www.ti.com/lit/ml/slyb214/slyb214.pdf.

[3] D. D. Putri and M. H. Fahrozi, "Upaya Pencegahan Kebocoran Data Konsumen Melalui Pengesahan RUU Perlindungan Data Pribadi (Studi Kasus E-Commerce Bhinneka.com)," in *National Conference On Law Studies*, 2020, pp. 978–979.

[4] M. D. K. Perdani, W. Widyawan, and P. I. Santosa, "Blockchain untuk Keamanan Transaksi Elektronik Perusahaan Financial Technology (Studi Kasus pada PT XYZ)," *Open J. Syst. Semnasteknomedia Online*, vol. 6, no. 1, pp. 7–12, 2018, [Online]. Available: https://www.ojs.amikom.ac.id/index.php/semnasteknomedia/article/view/2132/1936.

[5] A. A. N. D. H. Kesuma, I. N. P. Budiartha, and P. A. S. Wesna, "Perlindungan Hukum Terhadap Keamanan Data," *J. Prefer. Huk.*, vol. 2, no. 2, pp. 411–416, 2021.

[6] D. Godfrey-welch, R. Lagrois, J. Law, and R. S. Anderwald, "Blockchain in Payment Card Systems," *J. SMU Data Sci. Rev.*, vol. 1, no. 1, p. 3, 2018, [Online]. Available: https://scholar.smu.edu/datasciencereviewhttp://digitalrepository.smu.edu.Availableat:https://scholar.smu.edu/datasciencereview/vol1/iss1/3.

[7] S. I. Kim and S. H. Kim, "E-commerce payment model using blockchain," *J. Ambient Intell. Humaniz. Comput.*, no. 2008, 2020, doi: 10.1007/s12652-020-02519-5.

[8] U. Rahardja, Q. Aini, M. Yusup, and A. Edliyanti, "Penerapan Teknologi Blockchain Sebagai Media Pengamanan Proses Transaksi E-Commerce," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 5, no. 1, p. 28, 2020, doi: 10.24114/cess.v5i1.14893.

[9] D. Kaid, M. M. Eljazzar, and I. Member, "Applying Blockchain to Automate Installments Payment between Supply Chain Parties," *2018 14th Int. Comput. Eng. Conf.*, pp. 231–235, 2008, doi: 10.1109/ICENCO.2018.8636131.

[10] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117134–117151, 2019, doi: 10.1109/ACCESS.2019.2936094.

[11] F. Aprialim, Adnan, and A. W. Paundu, "Penerapan Blockchain dengan Integrasi Smart Contract pada Sistem Crowdfunding," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 148–154, 2021, doi: 10.29207/resti.v5i1.2613.

[12] Annisya and E. Haryatmi, "Implementasi Teknologi Blockchain Proof of Work Pada Penelusuran Supply Chain Produk Komputer," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 3, pp. 446–455, 2021, doi: 10.29207/resti.v5i3.3068.

[13] G. Dharma Putra and S. Sumaryono, "Rancang Bangun Identity and Access Management IoT Berbasis KSI dan Permissioned Blockchain," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 4, 2018, doi: http://dx.doi.org/10.22146/jnteti.v7i4.455.

[14] Lathifah Arief, T. A. Sundara, and Heru Saputra, "Studi Perbandingan Jaringan Blockchain sebagai Platform Sistem Rating," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 827–836, 2021, doi: 10.29207/resti.v5i4.2588.

[15] M. Pilkington, "Blockchain technology: Principles and applications. In F. X. Olleros, & M. Zhegu (Eds.)," *Res. Handb. Digit. Transform.*, pp. 225–253, 2016, doi: https://doi.org/10.4337/9781784717766.00019.

[16] L. Aniello, R. Baldoni, E. Gaetani, F. Lombardi, A. Margheri, and V. Sassone, "A Prototype Evaluation of a Tamper-Resistant High Performance Blockchain-Based Transaction Log for a Distributed Database," *Proc. - 2017 13th Eur. Dependable Comput. Conf. EDCC 2017*, pp. 151–154, 2017, doi: 10.1109/EDCC.2017.31.

[17] S. Damai, K. Hu, H. Novianus Palit, and A. Handojo, "Implementasi Blockchain: Studi Kasus e-Voting," *J. INFRA*, vol. 7, 2019, [Online]. Available: https://publication.petra.ac.id/index.php/teknik-informatika/article/view/8069.

[18] A. Susanto, "Implementation of Smart Contracts Ethereum Blockchain in Web-Based Electronic Voting (e-voting)," *J. Inf. Technol.*, vol. 18, no. 1, p. 56, 2020, doi: 10.26623/transformatika.v18i1.1779.

[19] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," *Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017*, pp. 557–564, 2017, doi: 10.1109/BigDataCongress.2017.85.

[20] A. Argani and W. Taraka, "Pemanfaatan Teknologi Blockchain Untuk Mengoptimalkan Keamanan Sertifikat Pada Perguruan Tinggi," *ADI Bisnis Digit. Interdisiplin J.*, vol. 1, pp. 10–21, 2020, doi: https://doi.org/10.34306/abdi.v1i1.121.

[21] A. K. Yadav and R. K. Bajpa, "KYC Optimization using Blockchain Smart Contract Technology," *Int. J. Innov. Res. Appl. Sci. Eng.*, vol. 4, no. 3, pp. 669–674, 2020, doi: 10.29027/ijirase.v4.i3.2020.669-674.