Development of Decentralized Electronic Payment System Using Blockchain

Putu Gery Wahyu Nugraha
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
gerywahyunugraha@gmail.com

Rinaldi Munir
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
rinaldi@informatika@org

Anggrahita Bayu Sasmitae
School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Bandung, Indonesia
anggrahita.bayu@informatika.org

***Abstract*—In Indonesia there are countless of electronic payment provider, but almost all of them use a centralized server architecture. This caused quite a few problems such as single point of failure, fragmentation of services and concern of privacy. To solve this problem a decentralized system will be made to utilize blockchain as a mean to do transaction. The chosen blockchain technologies are Stellar and Hyperledger. Stellar is used for transaction while Hyperledger is used for managing identity. The decentralized system made in this paper able to do basic functionality of electronic payment system while also accomplish the requirement for stability and response time.**

Keywords—blockchain, stellar, hypeledger, decentralized, e-money

I. Introduction

In Indonesia, electronic payment account for more than 21 trillion rupiah in 2018, this number is predicted to increase to around 600 trillion rupiah in 2023 [1]. The boom of electronic payment causes a lot of companies to take part in the scheme. Bank Indonesia react to the phenomenon by issuing Peraturan Bank Indonesia Nomor 20/6/PBI/2018 which regulates the implementation of electronic payment system in Indonesia. Most of the authorized electronic payment provider in Indonesia use a centralized server architecture.

Centralized server is an architecture where a single entity or company operates a server. This contrast a decentralized architecture approach where multiple entities or companies own a network of server. Although centralized server architecture has some advantages compared to the decentralized approach, it also has weaknesses, mainly single point of failure, fragmentation of services, and privacy of data.

Single point of failure is a weakness where a system can go down when one of their servers went offline. The effect is quite severe as it can cause a confusion for the user. The other, more challenging weakness is fragmentation of services and privacy of data. With each company fully owning their system, a user is locked to the company's system without any option to move their balance towards other payment provider without using intermediaries. This is not a problem for a healthy company, but it can cause confusion for the user if a company goes bankrupt and shut down their services. A company fully owning their system also poses risk in the privacy section. Usually, the users trust the company to secure their data and not use it for any illegal uses. But recent cases of data breach have raised the question of whether a user should trust the companies in taking care of their data or not.

Many of the problems stated above can be solved by using Blockchain. Blockchain is a technology that is first used by Nakamoto in implementing Bitcoin. Blockchain can solve the problem of untrusted peers in a distributed system [2]. Blockchain at first was only used for cryptocurrency, but throughout the year, by adding smart contract, blockchain has evolved to solved different use-cases such as online voting system and identity management.

Currently, there are various blockchain technologies that implement their own consensus protocol. As such, there is a need to compare various blockchain technologies to use in designing a decentralized electronic payment system.

II. Related Work

There are several research and work that focus on creating a decentralized electronic payment system. These works have different approach of the solution preferred in creating a decentralized electronic payment system.

BABB proposes a solution to revamp the banking system by building one on top of blockchain technologies [3]. Compared to payment related system built on top of blockchain like Bitcoin, BABB provides full banking functionalities on their system. BABB runs on top of Ethereum blockchain and use a modified ER20 (default token for Ethereum) token called BAX. To manage the identity of the user, BABB use *federated blockchain*. In a *federated blockchain* architecture, each entity that owns a stake to BABB system runs a node in the blockchain network. This enable ease of transfer of data from one entity to another entity and ensure the transparency of the data that resides in the system.

Another technology that is worth mentioning is Stellar. Stellar use its own consensus protocol called Stellar Consensus Protocol (SCP) [4]. The protocol, which derives from Practical Byzantium Fault Tolerance (PBFT) ensures a fast consensus between nodes in the network. Compared to other consensus protocol, SCP creates a network that is more decentralized than distributed by having a hierarchy of validators or node that will validate an incoming transaction.

III. Analysis and Design Solutions

A. Technology Selection

One of the constraints in developing an electronic payment system using blockchain in Indonesia is the regulation. There are two points that are considered important

in designing a correct blockchain-based system, first one is there needs to be an identity management system and the second one is all transactions should use IDR as the currency. This constraint makes Stellar and BABB, as mentioned in related works, not suitable to be use as electronic payment system in Indonesia as both of them use their own currency as a method of transaction.

Although Stellar itself use another currency to be used in transaction, a developer can create a custom token in Stellar so that the token can be used as a currency for the transactions. By making the token to not fluctuate against IDR or in other words making it values stable in the context of IDR, the system can be designed to support IDR, as in their stable token representation, in doing transactions. This design will solve the problem of using IDR as a currency representation, but it still doesn't solve the second problem of identity management. Although Stellar supports storing arbitrary data in their transactions (memo), Stellar capped the size of each data in a transaction to be only 28 bytes. This amount will not sufficive the requirements made by Indonesia government as it is usual for an Indonesia to have name with 20 characters or more (20 bytes).

To address the problems of identity management, one can use a blockchain technology that enable arbitrary format of data to be saved. One such technology is Hyperledger. Hyperledger itself is framework for creating blockchain based system [5], it allows arbitrary data to be saved in the blockchain and use PBFT as their consensus protocol. By combining Stellar and Hyperledger, a system can be created that utilize the strength of Hyperledger in managing arbitrary data and the strength of Stellar in managing transactions in the system.

B. Interaction Between Entities

In a decentralized system, the processing units of the system are the nodes that coordinate with each other, this also applies to a system that is made on top of blockchain. One of the caveats in maintaining multiple nodes is determining the ownership of each node. In this paper, the ownership of the nodes and the interaction between entities that own the nodes can be seen in the picture below

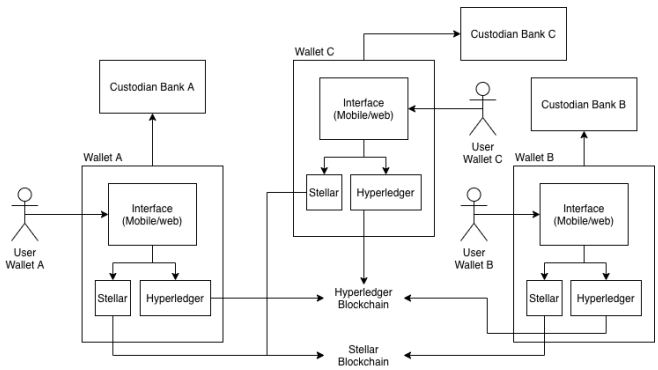

Fig. 1. Interaction between entities in the electronic payment system

As we can see from the image above, the electronic payment system will not be composed of a single company, instead, each company will implement their own system as a payment provider to interact with the customers. This ensures that each company still have freedom to innovate their business and attract customer independently. One thing that separates this implementation from a regular centralized electronic payment system (where each company can still implement their own products) is that each company implement their products on top of an existing blockchain network. This network will be composed of node of Hyperledger and Stellar blockchain that is run by each entity that maintain the system. Because each transaction is done and saved on-chain, theoretically this architecture will enable user from each payment provider to transfer their money seamlessly between payment provider.

C. Ensuring Privacy of Data

One of the caveats in designing a decentralized system is how to ensure the privacy of each user data. Unlike a centralized system where majority of data is owned by a single entity, and the user is forced to trust that entity, in a decentralized environment the ownership of the data is shared between the entities that run the system. This can cause a problem as there are no ways to ensure that the entity that runs the node have the same intention as others.

The solve this problem we can design the system so that each user data that is going in and out of the system is encrypted. To make this encryption possible, we can use a private/public key system where a user encrypts their data using their private key and save the resulting ciphertext in the blockchain. This solve the problem of data ownership as now nothing but the user able to decrypt their data and the ciphertext that is saved in the blockchain is practically useless for each node's owner.

D. System Design

The design of the system will be composed of two components. The first one is the frontend, where the system directly interacts with the user. As stated in interaction between entities, each entities/company can implement their own interface for the user to interact with. As such, there are various type of systems that can be considered frontend component, it might be in the form of a website or a mobile app. Although the type of system differs, one thing that is certain is that each frontend component should be able to generate their own symmetric/asymmetric key to be used during encryption and signing of a request to the blockchain.

The second component that is required by the system is the blockchain. As stated, there are two blockchain technologies that are going to be used in this paper, Stellar and Hyperledger. Unlike frontend components where each entity can implement one as they wish, all of the entities that runs the electronic payment system will use the same blockchain network.

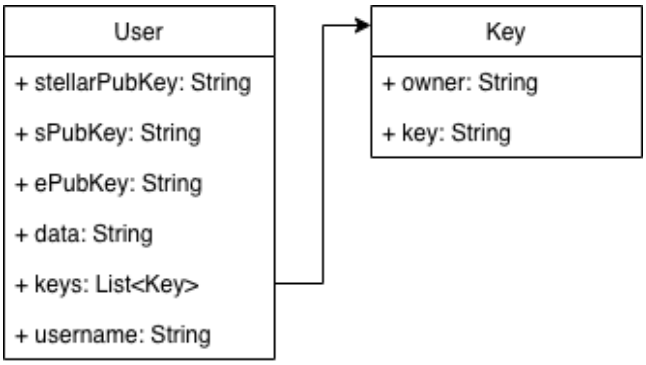

Fig. 2. User data schema for Hyperledger blockchain

As depicted in Figure 2, each user will save the public part of their generated asymmetric key inside the blockchain, their username, and ciphertext of their data. Two of the keys (stellarPubKey and sPubKey) will be used to sign the transaction for Stellar and Hyperledger respectively. While the last key (ePubKey) will be used in an multi party public key encryption that will be explained in the following section.

For now, the state of the system design is that every user will have their own asymmetric key for encryption and will only sent the ciphertext to the blockchain. This raises one problem mainly how can a third-party entity like the government verify the user data as all of their data is going to be encrypted. This problem requires a solution that can allow other permitted party to read other user data while still maintaining the privacy of user data in the blockchain. One of the solutions is to just decrypt the data in user's mobile app and send the decrypted data to the recipient server. This flow will work if and only if the recipient is discoverable (running on a public server), in many instances this is not the case as sometimes a user want to permit other user that doesn't have any public facing server.

The other solution to solve this problem is by using a multi-party public key encryption scheme. This scheme is mainly use in Pretty Good Privacy (PGP) where for each data there will be a single symmetric key that is used to encrypt the data. This single symmetric key will then be encrypted by all the other recipient's public key and saved in an online repository, or in this case the blockchain. In this way, each recipient will have their own version of encrypted public key of a user in the blockchain. This way if a recipient A want to read another user B data, they only need to fetch the encrypted symmetric key that is owned by A and use A private key to decrypt the encrypted symmetric key to get the key that will decipher user B data. In Hyperledger, this scheme is enabled by saving the list of encrypted keys and their owner in the variable keys in the User Scheme.

As there is quite a lot of addition in a decentralized electronic payment system, some of the most significant flow of information will be presented here using a DFD diagrams.

1) Registration

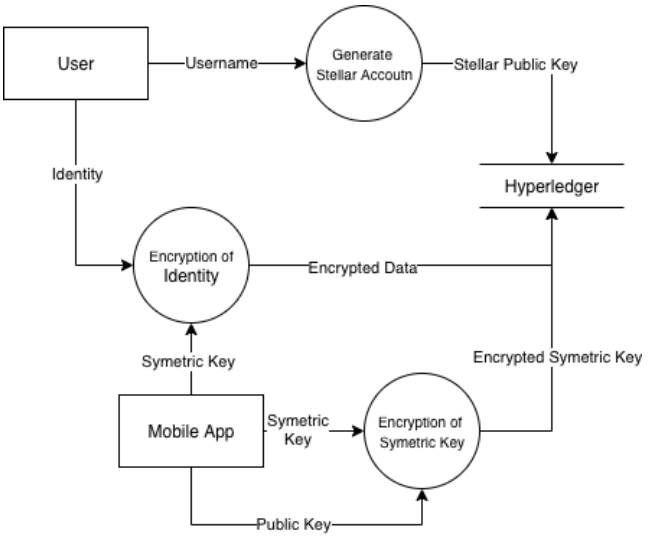

Fig. 3. Registration data flow

As shown in Figure 3, in registration, one of the most prominent change in the DFD is the generation of asymmetric key in the process of creating an account. In this step, there will three asymmetric keys created, one will be for signing transaction for Stellar and the other will be for signing and encrypting transaction in Hyperledger blockchain.

Each data that user input into the system will be encrypted first using a generated symmetric key, the resulting ciphertext will then be saved into Hyperledger blockchain. One interesting thing to note is that although there is no permitted user yet for the multi-party public key encryption scheme, there is one encrypted symmetric key that will be saved on Hyperledger. This encrypted symmetric key is owned by the registered user itself; this is logical as the first user that is permitted to read our own data will be us themselves.

2) Transfer

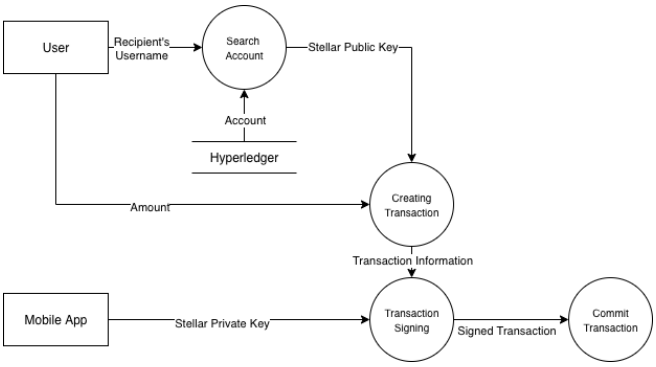

Fig. 4. Figure 1 Transfer data flow

Stellar itself requires the amount to be made, the token that is going to be used, in this case IDR token, and the recipient public key to create a transaction as shown in Figure 4.

To accomplish the needed requirement for a transfer operation, a sender will input recipient username into their mobile app. This username will be used to query in Hyperledger the corresponding Stellar public key of the recipient. If it is found, the sender will create a transaction with the amount set to the amount inputted and the destination

to be the recipient's public key. This transaction will then be signed using the sender's own Stellar private key before then sent and committed to the Stellar blockchain.

3) Payment to Outside Entities

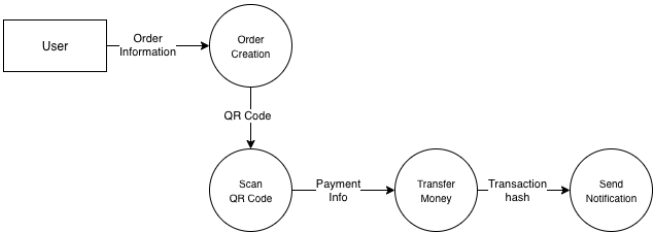

Fig. 5. Flow of data when doing payemnt to outside entities

In a regular centralized system, a payment to outside entities, like ecommerce can be made quite trivial. This is the case because there is a single trusted entity between the two parties (sender and recipient) that can orchestrate the verification of the payment and notify the recipient if a payment has been made. In a decentralized system, there is no such deterministic way for a node in a blockchain network to notify outside observer if a payment has been made as there is no way to ensure only a node will notify the outside observer. To solve this, usually each stakeholder to the network can tap directly into the blockchain and verify the payment itself, but this operation also needs a trigger so that it happens right after the user has sent the money.

To create that trigger, the system can delegate sender's mobile app to notify the recipient if a payment has been made. To do this, sender will need to know a URL of some sort that can be accessed when the sender has made the payment. In this paper, the URL, the order id, and the amount of money transferred will be saved inside a QR Code that sender can scan when starting a transaction. When the money is sent, the sender can send the appropriate transaction has to the recipient by using the URL provided in the QR Code. The recipient will need to verify if the transaction contains the correct amount and recipient by tapping directly into the blockchain.

One of the weakness of this approach is there is no way for the recipient to know if the transaction hash that the user sent has not been used for other order. For example, user A bought an item costing 180.000 IDR and then sent the money to recipient B which resulted in transaction hash C. Because recipient B only check if the transaction mentioned by the hash has the correct amount and the correct recipient, user A can theoretically send the same transaction hash to buy the same item costing 180.000 IDR from the same recipient B. To counter this, the system can leverage 28 bytes of storage that Stellar provides to save the order id of a transaction. By doing this, the system ensures that this transaction can only be used to approve the mentioned order id.

IV. Implementation and Testing

A. Implementation

To implement an electronic payment system in this paper, we will need to implement four main components. Each can be categorized as frontend or backend.

1) Frontend

For the frontend side there will be a mobile app and also an e-commerce site to test the integration between the system and outside entities. As shown in the interaction between entities before, each companies/entity can freely implement their own interface for the user, as such this mobile app implementation is only one of the many implementations that can be used by a company.

Kotlin will be used to program the mobile app. Kotlin itself is a language that runs on JVM and can run at native speed on mobile android devices. Performance is an important aspect on this system because there will be quite of lot of cryptography that is happening inside the mobile app.

For e-commerce site, an e-commerce framework like Woocommerce will be used. Woocommerce is a very popular framework made on top of Wordpress to bootstrap an e-commerce site. Interaction between the site and the system will be done by using a Woocommerce plugin. By doing this, a developer can simply create a payment plugin that targets multiple site that is built on top of Woocommerce.

2) Blockchain

There will be two implementations of blockchain used in this system. One is for identity management built on top of Hyperledger and the other is for managing transaction using Stellar. Each blockchain will have their own architecture to be implemented.

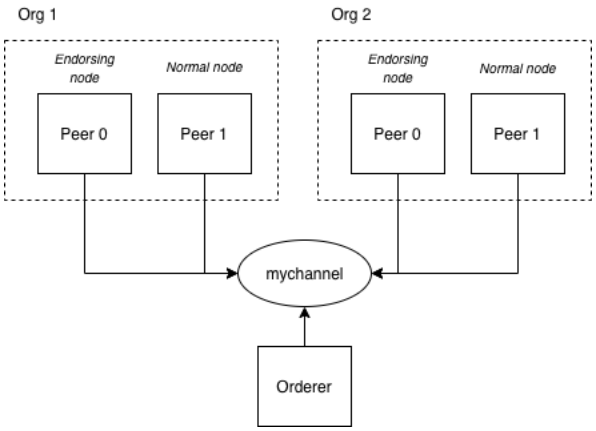

Fig. 6. Hyperledger architecture

For Hyperledger, the blockchain network will consist of four nodes, two will be used to act as an endorser, and the other will act as a normal node. Endorser node is a node that is going to validate/endorse a transaction. An organization in the network has an analogy of the entities/companies that runs the network. By having at least one endorsing peer in an organization, the network ensure that for each transaction that is going to be committed to the system, at the very least all the entities that runs the system will know that the transaction happened.

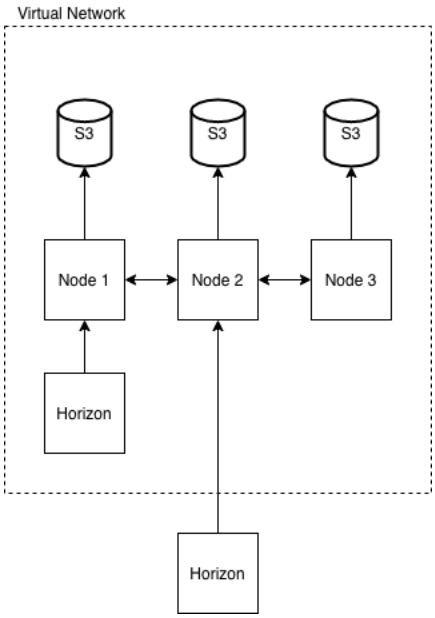

Fig. 7. Stellar Architecture

For Stellar, the architecture will be quite different as the network only has one type of nodes. These nodes will connect to their own AWS S3 to store history ledger and save data on the main programs. Also, Stellar network will expose their system using two special nodes called Horizon, one connects through Node 1 and the other connects through Node 2.

B. Testing

There will be two testing approach used in this system, one approach will focus on testing the stability of the system while the other will focus to test the response time of the system.

1) Stability Testing

To test the stability of the system, one of the nodes in the blockchain will be stopped prematurely and then the system will be forced to receive a new request. This can be done quite easily as all the node in the blockchain network runs on containerization services such as docker. In a successful test, the system should be able to handle the request even though it lost a single node.

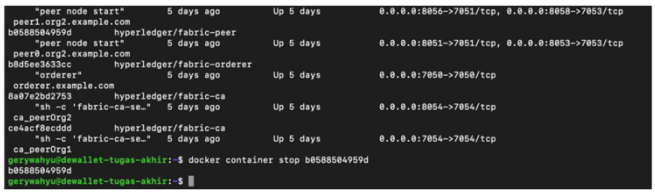

Fig. 8. Process of stopping a node

Figure 8 shows the process of stopping a node in the blockchain network. After this process an HTTP request will then be sent to the network to check if network still operates correctly.


```
1. {
2.   "success": true,
3.   "message": "Successfully invoked the chaincode Org1 to the
   channel 'mychannel' for transaction ID:
   47d53343e9123e59f0a1739b88a1ddbf2734f5e9f62410300d09ceb8126d
   ecf9",
4.   "data": {
5.     "publicKey":
   "GBA573YGQEFZ3IY6UUAHI7ETBHERQ4FQYSJTWIJ4JM4WQGLJEP6FHS7X"
6.   }
7. }
```


Fig. 9. Result of HTTP request

The result of the HTTP request shows that even though the system lost a single node, it can still handle request successfully.

2) Response Time Testing

The second testing that is going to be done on the system is a response time test. To ensure that the system can server the user even under load, the system should be able to response through request in the minimum time of 30 seconds. To accomplish the environment needed to produce loads on the system, a tool called locust will be used. This tool can produce a swarm of request directed to the system.

For Hypeledger blockchain, the stress test will involve a thousand concurrent users sending a request for the operation GetUsername. This operation is chosen as it is the most common operation used in the system. The result of the test is as follow.

Average(ms)	Min(ms)	Max(ms)	RPS	Failure
127	43	1288	98.4	0%

Fig. 10. Result of stress test in Hyperledger blockchain

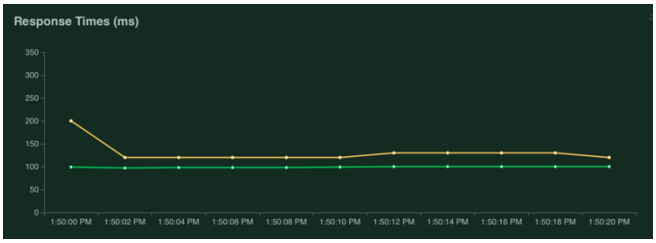

Fig. 11. Response time over time of Hyperledger blockchain

The stress-test result for Hyperledger (Figure 10) blockchain shows that the blockchain can still produce ~90 request per second when there are a thousand concurrent user using it. The graph of response time over time in Figure 11 also shows a very stable system as there is no peak found in the graph.

For Stellar, the stress test will also involve a thousand concurrent users, but this time half of the users will be directed into Horizon 1 while the other half will be directed to Horizon 2. The result of the test is as follow

Operasi	Average(ms)	Min(ms)	Max(ms)	RPS	Failure
Lihat saldo	50	7	1194	49.4	0
Transfer	291	101	6112	13.4	0

Fig. 12. Result of stress test in Stellar blockchain

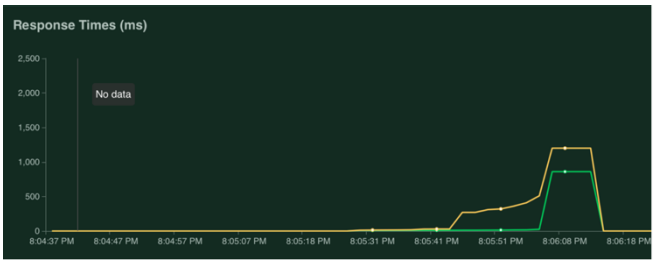

Fig. 13. Response time over time of Stellar blockchain

As shown in Figure 12 and Figure 13, there is a peak in the graph of response over time and also in the maximum response time of Stellar blockchain. This peak indicates that Stellar struggles when handling a thousand concurrent users at the same time.

V. Conclussion

It is shown that it is possible to create an electronic payment system made to comfort with the regulation of Indonesia government. This system can be made in a decentralized manner by using blockchain implemented by Stellar and Hyperledger. System also successfully handles various functional requirements introduced by regular electronic payment system.

The performance of the system, although lower than the maximum response time of 30 seconds, still have some room to improve as shown in the test with a thousand user Stellar blockchain unable to handle the situation without introducing a peak in their response time.

For future work, it is much preferable to find an alternative technology, or design in such way that removes the needs to use two blockchain technologies as this will greatly increase the maintenance-ability of the system. Also, there needs to be a solution to reduce the peak response time during testing of Stellar blockchain.

Acknowledgment

I would like to express my greatest appreciation to Dr. Rinaldi Munir, MT. and Anggrahita Bayu Sasmita, MT. for valuable and constructive guidance and teaching during the whole process of this research. I also would like to thank all my colleagues who also helps me research and inspire me to design the solution for the problem.

References

[1] Bank Indonesia, "Statistik Sistem Pembayaran - Transaksi," 2019. [Online]. Available: https://www.bi.go.id/id/statistik/sistem-pembayaran/uang-elektronik/contents/transaksi.aspx.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[3] BABB, "BABB Whitepaper," December 2017. [Online]. Available: https://resources.getbabb.com/whitepapers/en/babb-whitepaper.pdf.

[4] Stellar, "Stellar Consensus Protocol," 2016. [Online]. Available: https://www.stellar.org/papers/stellar-consensus-protocol.pdf.

[5] C. Cachin, "Architecture of the Hyperledger Blockchain Fabric," Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.