Logistic Supply Chain Management System Modeling Using *Blockchain*

Muhammad Alif Solihin[1a)], Dahliah Nur[1)] , Eddy Tungadi [1)] , Iin Karmila Yusri [1)]
[1] Informatics and Computer Engineering, Politeknik Negeri Ujung Pandang, Jln. P.Kemerdekaan KM 10. Makassar
[a] alif.solihin72@gmail.com
[b] dahlia@poliupg.ac.id
[c] eddy.tungadi@poliupg.ac.id
[d] iin.yusri@poliupg.ac.id

***Abstract*-Indonesia's economic growth in the logistics sector is becoming increasingly significant, alongside the rising demand for goods. However, the complexity of logistics supply chain management in Indonesia still faces challenges, including security, human error, and lack of transparency. This research proposes the use of blockchain technology for a logistics supply chain management system, which is capable of storing transaction data in a decentralized and distributed manner. Ganache is used as a local blockchain server, Metamask for transactions, and smart contracts to manage user flows on a web3 dapps-based website. The research procedure includes literature study, design, development, and system testing. The test results indicate that all features and functions of the system operate as expected. The system not only successfully handles high demand and transaction spikes but also demonstrates enhanced data security, improved transparency through real-time tracking, and robust performance under high workloads. Overall, this blockchain-based logistics supply chain management system effectively secures and manages transaction data while enabling data tracking and tracing, thereby increasing transparency and handling high concurrent transactions efficiently.**

Keywords: Blockchain Network, Supply Chain, Ganache, Metamask, Logistics

I. Introduction

Indonesia's steady economic growth has been seen in recent years. In the first quarter of 2023, the Central Bureau of Statistics (BPS) reported that Indonesia's economy grew by 5.03% compared to the same period the previous year. One sector that showed significant growth was the transportation and warehousing sector, which recorded an increase of 15.93% [1]. This growth reflects increased activity in the logistics sector, which is becoming growingly important in the context of trade globalization.

However, globalization and complexity in supply-chains create their own challenges. More and more complex supply chain management systems increase potential constraints, especially with regard to coordination between the parties involved [2]. This lack of coordination can lead to increased production costs and continuously more the risk of *human error*, as well as unnecessary repetition of data [3][4]. These constraints are gradually more relevant in the context of supply-chain management in the logistics sector, particularly at ports, which often face issues of process complexity, lack of transparency, and risks to data security and integrity [5][6].

Several traditional solutions such as Enterprise Resource Planning (ERP) and Supply Chain Management Systems (SCMS) have been used to address this issue, but they have limitations in terms of real-time visibility and data integrity. A recent study by Chang (2020) showed that ERP and SCMS solutions still rely on database data, which is prone to manual errors and difficult to audit [7].

Given these shortcomings, blockchain technology emerges as a promising solution to enhance transparency and security in logistics supply chain management. HMZ Cargo, for instance, is conducting model trials as part of its plan to transition its logistics supply chain to a blockchain-based system. It is hoped that this technology will deliver a more efficient and transparent solution [8].

II. Research Methods

The research method requires a research approach to ensure the right structure and direction, so that the results of the research carried out are in line with the predetermined objectives. For the flow of research conducted can be seen in Figure 1 below.

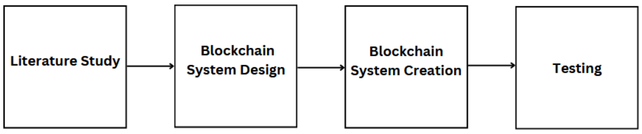

Figure 1. Research flow

2.1 Literature Study

Literature studies are conducted better to understand how to create a system with *blockchain* technology to manage a transparent supply chain process. At this stage, it is carried out to obtain and collect accurate data from various sources of information such as books, journals, theses, and other literature. In this activity, information relevant to the case or research problem is identified, studied, and recorded, in order to find the information needed related to the research conducted.

2.2 *Blockchain* System Design

At the stage is a step to create a system design so that it can clearly get an overview of the research to be done, it will make the design of the blockchain system and the workflow of the role of each *user* involved in this process. The design can be seen in Figure. 2, 3 and 4.

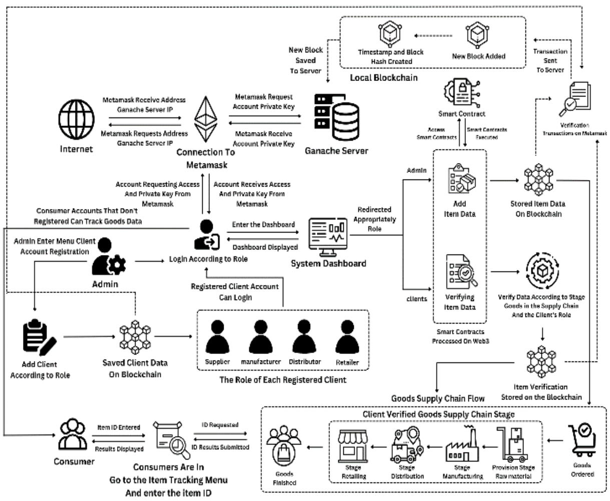

Figure 2. Modeling the Supply Chain System with *Blockchain*

Figure 2 explains the flow of the *blockchain* system starting with ensuring all accounts are connected to the *ganache* server network and integrated with *metamask.* Each account must get a *private key* from the *ganache* server and *log in* according to the predetermined role.

Admins can add new *client* accounts and input goods data into the *blockchain* network, while *clients* consisting of suppliers, manufacturers, distributors, and retailers verify the goods data received according to the stages in the supply chain. The supply chain process starts with the admin inputting the goods data, which is then forwarded through the client to the consumer.

All transactions are recorded and stored on a local *ganache blockchain* network through a *smart contract* protocol, where each transaction generates a hash and timestamp that is recorded on the *blockchain*. For unregistered consumers, they can track item data by item ID, which will display the last point of verification by the relevant *client.*

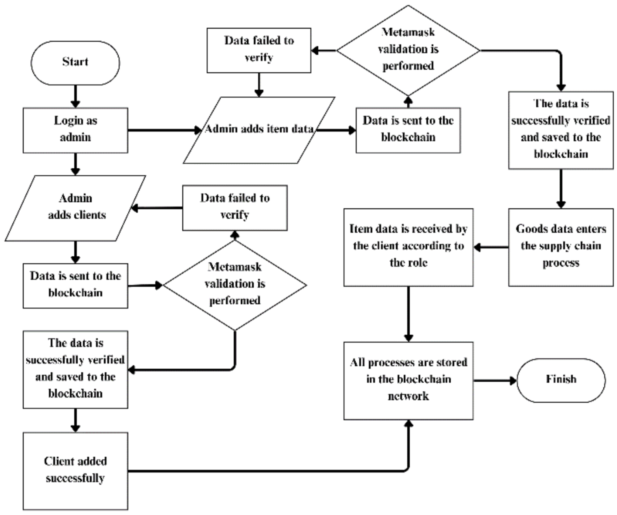

Figure 3. Admin Process *Flowchart*

In Figure 3 explains the *flowchart of* the admin process flow. Starting from the admin *logging in* and choosing to add a *client* or add item data. First add a *client*, the results of the data input are sent to the *blockchain* network and validated using a metamask, if successful, then the *client is* successfully added. Second, adding item data, the results of the data input are sent to the *blockchain* network and validated using a metamask, if successful, the item data will enter the supply chain flow stage. The data is received by the *client* according to its role. All input results to the *blockchain* will be stored in the *blockchain* network.

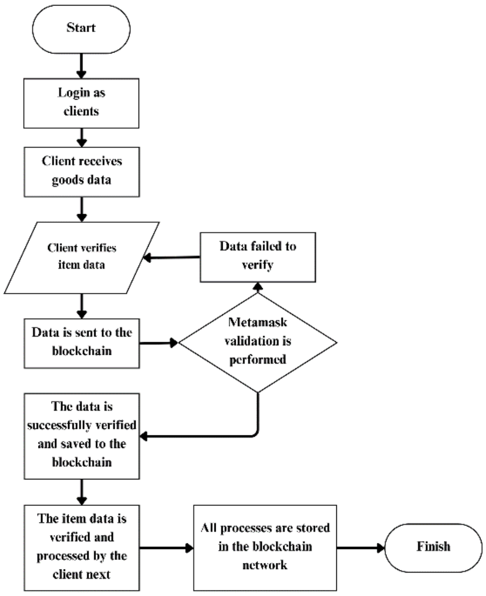

Figure 4. *Client* Process *Flowchart*

In Figure 4, explains the *flowchart of* the *client* process flow. Starting from the *client logging in* the *client* receives item data according to the role of the *client* who *logged in*. After the item data is received, the *client* verifies the item data. The results of the verification will be sent to the *blockchain* network and validated using *metamask*. If the process is successful, the item data will be verified and the next stage is processed by the succeeding *client*. All processes carried out will be stored in the *blockchain* network.

2.3 *Blockchain* System Creation

At this point, the modeling system is created based on the previously developed system design phases. Software that supports blockchain system development is used to implement this design. A smart contract system is created using the Solidity programming language, with Remix IDE utilized for testing. The blockchain network operates using Ganache, which provides a controlled local environment for testing without transaction fees, while transactions are facilitated through Metamask e-wallets, which manage gas fees efficiently. Additionally, the React JS framework, capable of interacting with smart contracts, is used to develop the user interface. To build a blockchain system effectively, all necessary supporting tools, including Ganache and Metamask, are essential for a streamlined and efficient development process.

2.4 Testing

The goal of this blockchain system testing is to determine whether the system can function properly or if faults still need to be fixed. In this study, load testing, scalability testing, and blackbox testing were the test methodologies employed. Below is a screenshot of the test scenario:

1. *Blackbox* Testing Scenario

Blackbox testing involves evaluating the program's functionality without needing to understand its internal workings, code structure, or implementation specifics. The function of each page and the condition of whether the value is true or false are tested in this test case.

Blackbox testing is said to be successful if the expected results match the test results, while the test is said to fail if the expected results do not match the test results.

2. *Load Testing* Scenario

Testing the software's ability to handle workloads under varied conditions is known as load testing. The test scenario is conducted using Apache JMeter, one of the tools for software performance testing.

Apache JMeter is used to simulate the workload on the software by adding users gradually. The test results are carried out by measuring the latency value, response time and average *throughput* of the various test loads carried out. *Load* testing is said to be successful if the average value of latency and response time of each test scenario is almost the same as the results of other test scenarios.

3. *Scalability Testing Sc*enario

Scalability testing is done by testing the performance of the software in handling the number of transaction spikes that occur at the same time. The test scenario carried out is to connect the *client* device with the *server* and then process the transaction and calculate the duration inputted from the *client* until it is displayed back to the system.

The *client* devices used have different specifications, so as to determine the performance of the hardware in processing data. The test results are carried out by measuring the transaction duration and average response time of the device being tested. *Scalability testing* is said to be successful if the average value of latency and response time from each test scenario is almost the same as the results of other test scenarios.

III. Results and Discussion

The results obtained from this research are based on the design, implementation stages, and testing in accordance with the subject matter and scope of the research. This research focuses on modeling a logistics supply chain management system that uses *blockchain* technology. This system uses *React JS* for the user interface, *Solidity* programming language to write *smart contract* functions, and *Metamask* as a digital crypto *e-wallet* connected to *Ethereum*. *Ganache* is used as a local *server* to run the *blockchain* process in simulation.

3.1 UI Display Interface Result

3.1.1 *Login* Page Display

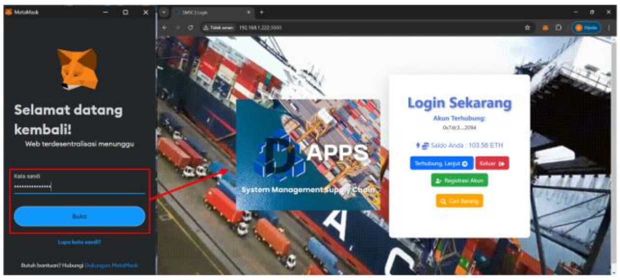

Figure 5. *Metamask Login* and *Login* Page

Figure 5 displays the page to enter the main page, the *dashboard* page will appear according to the role of the user who *logs in* on the page.

3.1.2 *Register* Page Display

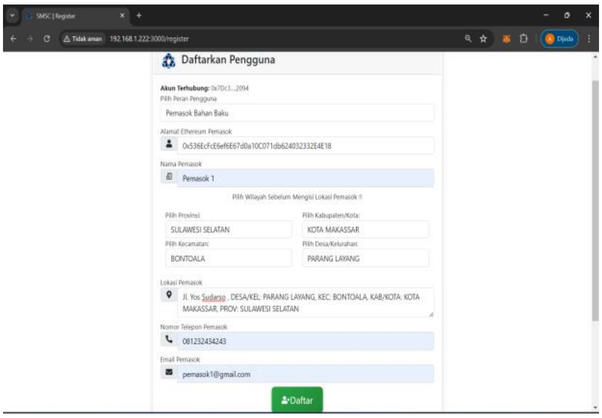

Figure 6. User *Register* page

Figure 6 displays the page for registering a user account. User registration can only be done by the admin. If it is not an admin who registers, the inputted data cannot enter the *blockchain* network.

3.1.3 Display of Track Data Page

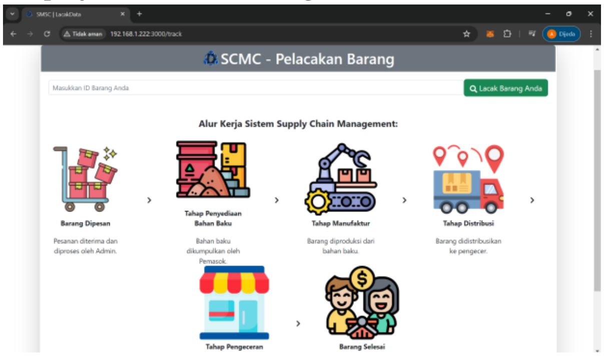

Figure 7. Track Data page

Figure 7 displays a page for tracking item data, each user can search for the desired item data by entering the Item ID.

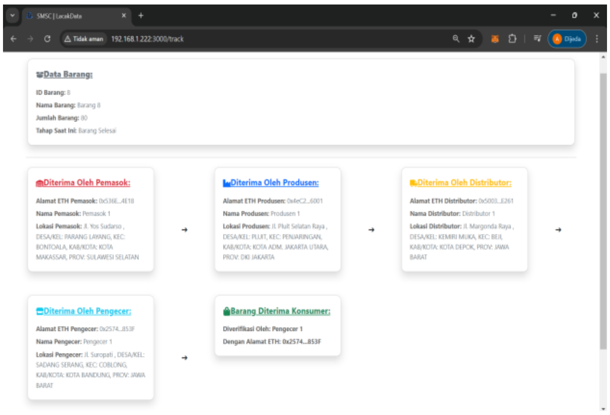

Figure 8. Track Item Search Results

Figure 8 displays the search results display page of the desired item data, it can be seen the process of searching for item data, from the initial stage to the last stage.

3.1.4 Admin Page View

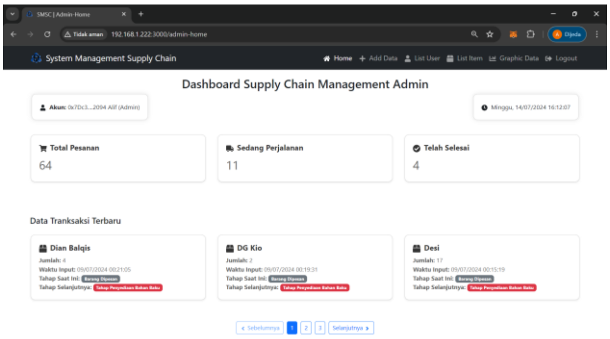

Figure 9. Admin *Home* Page Display

Figure 9 displays the *home* display page of the admin, for the admin there are five menus such as *home*, *add* data, *list users*, *list items and graphic* data. Each menu has a special function that only exists for an admin.

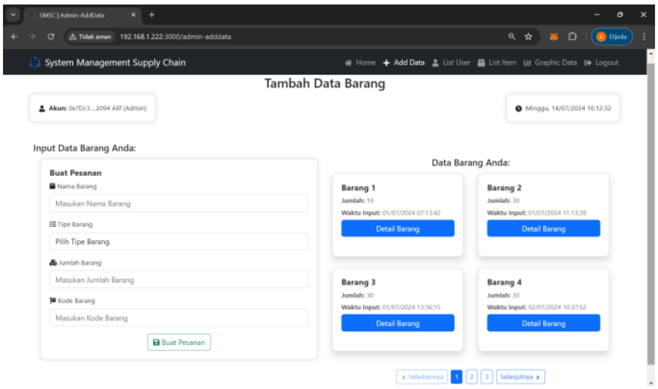

Figure 10. *Add* Data Page Display on Admin

Figure 10 displays the display of the *add* data page on the admin or menu to add or input item data into the *ganache blockchain* network. The inputted item data will enter the first stage, namely the ordered goods stage, which means that the item has been recorded and accepted by the admin and will be processed by the next account.

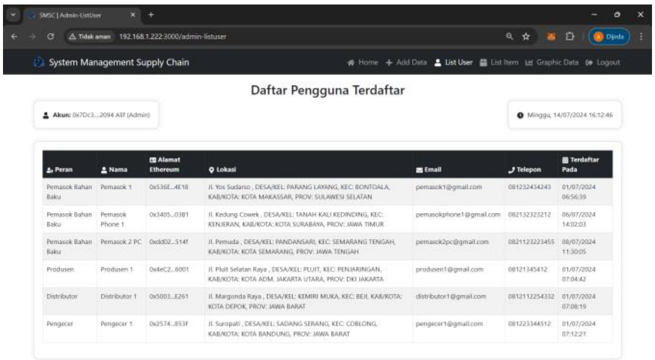

Figure 11. Admin Display of Registered User Data Table

Figure 11 displays the display of the *list user* page on the admin or menu to display user data that has been registered by the admin. Users who have been registered by the admin will appear in the table and will display account information in the form of role, name, ETH address, location, email, telephone and time the account was registered.

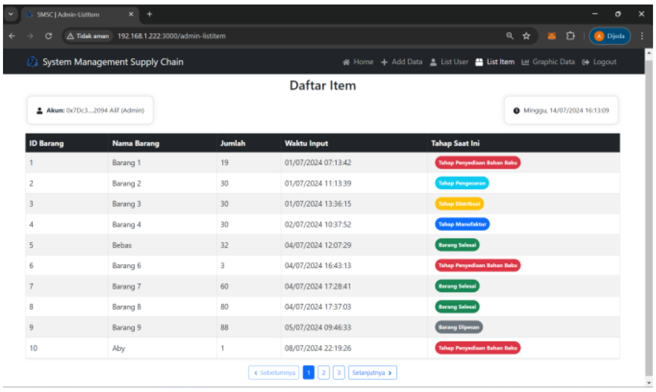

Figure 12. Display of Item Data List Table on Admin

Figure 12 displays the display of the *item list* page on the admin or menu to display item data that was added by the admin. Item data display, such as item ID, item name, quantity, input time, and also the current item stage. In the process of the stages of goods, colors are given to make it easier to know the current stage of the item. Gray color means goods ordered, red color means raw material provision stage, dark blue color means manufacturing stage, yellow color means distribution stage, light blue color means retail stage, and finally green color, which means Goods completed. These are the stages that all must go through until the goods are received by the consumer.

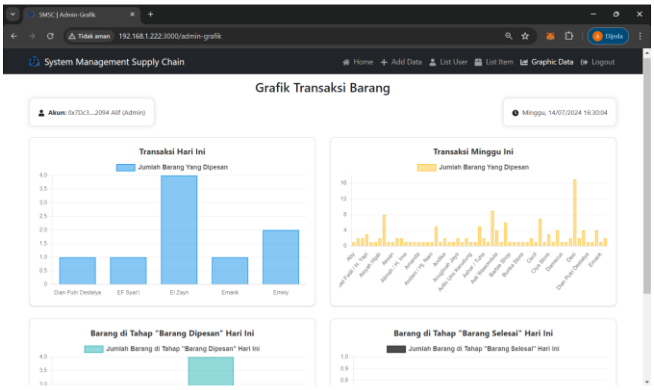

Figure 13. Display of Goods Data Graph on Admin

Figure 13 displays the display of the *graphic* data page on the admin or menu to display item data presented in graphical form. Graphic data displayed such as transactions today, this week, goods in the ordered goods stage today and goods in the finished goods stage today.

3.1.5 Supplier Page View

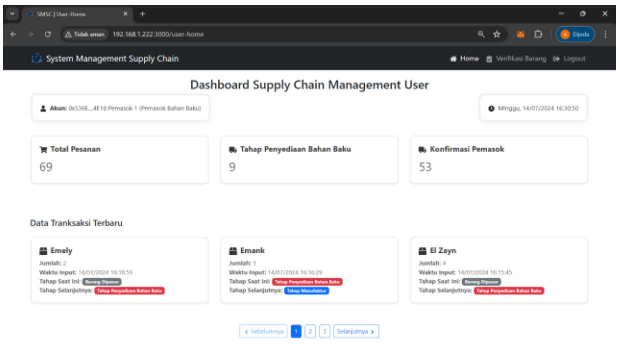

Figure 14. User *Home* Page Display

Figure 14 displays the *home* display page of the user, for users there are two menus such as *home*, and verification of goods. This menu is useful for helping users monitor the amount of item data and verify items that have been inputted by the admin.

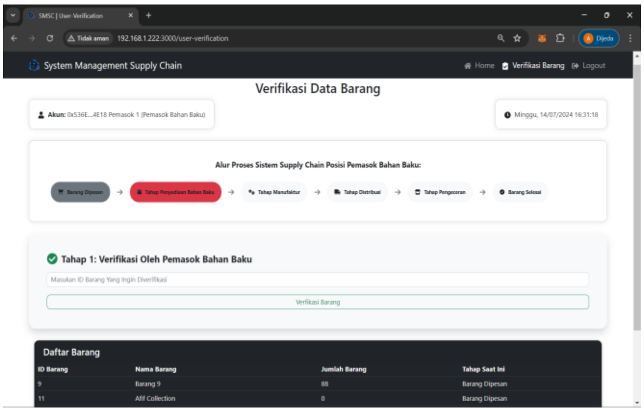

Figure 15. Display of Goods Verification Page on User

Figure 15 displays the user goods verification page or menu to verify the goods received by each user according to their respective roles. At this stage, incoming goods in accordance with the accepted user role will be entered and a verification process will be carried out so that the goods can be forwarded to the next stage.

3.2 Load Testing

Load testing is used to test how the system or software created can handle workloads or users simultaneously. Load testing will give a workload in the form of a request to the software at the same time. For testing can be seen in Table 1.

Table 1. *Load Testing* Results

Number of Requests	Average Latency (Milliseconds)	Average Response Time (Milliseconds)	Average Throughtput
10	6,4 Milliseconds	96,2 Milliseconds	2,2/Seconds
20	5,6 Milliseconds	131,95 Milliseconds	4,1/Seconds
30	6,83 Milliseconds	117,93 Milliseconds	6,1/Seconds
40	5,95 Milliseconds	126,82 Milliseconds	8,0/Seconds
50	10,52 Milliseconds	104,32 Milliseconds	10,0/Seconds
60	9,16 Milliseconds	114,63 Milliseconds	12,1/Seconds

Table 1 is the result of *load testing* where testing is done by making *requests* or requests to the *server* simultaneously. This test is carried out using Apache JMeter *software* to simulate the workload of the software. The variation in the number of requests made is 10, 20, 30, 40, 50 and 60 by measuring the latency value, response time and *throughput*. However, if the request value is above 60, it will experience a response failure.

Based on the test results, the software is able to respond to requests simultaneously well with a maximum value of up to 60 requests at the same time with average results from latency, response time and throughput that are almost the same in each test scenario.

3.3 Scalability Testing

Scalability testing is used to test how the system or software created can handle spikes in the number of transactions that occur at the same time. Scalability testing will test the process of transaction spikes that are carried out at the same time. For the test can be seen in Table 2.

Table 2. Scalability Testing Results

User	Number of Transations	Transaction Duration	Average Response Time (Seconds)
PC A PC B	1	13,67 Seconds 18,25 Seconds	15,96 Seconds
PC A PC B	5	70,8 Seconds 85,77 Seconds	15,65 Seconds
PC A PC B	10	132,35 Seconds 147,70 Seconds	14,00 Seconds
PC A PC B	30	384,15 Seconds 407,22 Seconds	13,19 Seconds
PC A PC B	50	785,30 Seconds 801,52 Seconds	15,86 Seconds

Table 2 is the result of scalability testing, where testing is done by connecting the client PC to the ganache server network. PC A has higher hardware specifications than PC B. Testing is done on client PCs A and B by performing transactions at the same time. Variations in the number of transactions performed are 1, 5, 10, 30, and 50 by measuring transaction duration and average response time.

Figure 13 displays the display of the *graphic* data page on the admin or menu to display item data presented in graphical form. Graphic data displayed, such as transactions today, this week, goods in the ordered goods stage today, and goods in the finished goods stage today.

Based on test results on PCs A and B, the software is able to handle the increase in the number of transactions occurring simultaneously. The average response time duration results are almost the same in each test scenario carried out.

IV. Conclusion

The results of this test resulted in modeling a blockchain based supply chain management system, so there is no direct implementation yet. Based on the results of the research that has been carried out, it can be concluded that:

1. This research successfully built a logistics supply chain management system modeling using *blockchain* technology.
2. The *blackbox* test results show that all system features have run as expected.
3. *Load testing* and *scalability testing* show that the *blockchain* system is able to handle increasing workloads efficiently.
4. The system can handle the surge of transactions from *client* to server with almost the same response time in every test scenario conducted.
5. The *blockchain* network system built is proven to be efficient and stable in handling transactions in *real time*.

Acknowledgments

The author expresses his deepest gratitude to God Almighty, both parents, friends and friends, both supervisors, as well as all lecturers and staff of the Department of Informatics and Computer Engineering Computer and Network Engineering Study Program of Ujung Pandang State Polytechnic, PT HMZ Cargo, and all those who have helped in compiling this research to completion.

Reference

[1] Rachman, "Growing 5.03% in Q1, BPS: Indonesia's Economy Stable," cnbcindonesia.com. Accessed: Jun. 02, 2023. [Online].Available:https://www.cnbcindonesia.com/news/20230505075834-4-434663/tumbuh-503-di-kuartal-i-bps-ekonomi-ri-stabil

[2] . Gurtu and J. Johny, "Supply chain risk management: Literature review,"*Risks*, vol. 9, no. 1, p. 16, 2021.

[3] . Creazza, C. Colicchia, S. Spiezia, and F. Dallari, "Who cares? Supply chain managers' perceptions regarding cyber supply chain risk management in the digital transformation era,"*Supply Chain Manag. An Int. J.*, vol. 27, no. 1, pp. 30-53, 2022.

[4] .Cuandra, R. A. Qadri, D. S. Syaharni, E. G. Fauzi, J. Valeria, and S. Sirait, "Analysis of the Implementation of Supply Chain Management and Erp at Pt Semen Padang,"*Juremi J. Ris. Econ.*, vol. 2, no. 1, pp. 63-72, 2022.

[5] .J. Aroba and G. Prinavin, "Improving supply chain management processes through digital transformation of ERP systems in the oil and gas industry,"*Interdiscip. J. Econ. Bus. Law; Vol. 12, Issue 3*, 2023.

[6] . Indraprakoso, "Exploring the Potential Use of Blockchain in Optimizing Port Management in Indonesia:*A* Literature *Review*," *Sanskara Manaj. And Business*, vol. 1, no. 03, pp. 140-160, 2023.

[7] G. Witjaksono, V. Virginia, A. Z. Amanda, P. Pandri, and L. Judijanto, "Blockchain *Technology* in Supply Chain Management: Improving Transparency and Security," *J. Cahaya Mandalika ISSN 2721-4796*, vol. 3, no. 2, pp. 2146-2152, 2023.

[8] S. E. Chang and Y. Chen, “When blockchain meets supply chain: A systematic literature review on current development and potential applications,” *Ieee Access*, vol. 8, pp. 62478–62494, 2020.