Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 6 No. 2 (2022) 295 - 304 ISSN Media Electronic: 2580-0760

Food Fraud Prevention using a Blockchain-Based System: Case Study Slaughterhouse in Sidoarjo

Daniel Soesanto[1], Liliana[2], Bambang Prijambodo[3]
[1, 2, 3]Teknik Informatika, Fakultas Teknik, Universitas Surabaya
[1]daniel.soesanto@staff.ubaya.ac.id, [2]lili@staff.ubaya.ac.id, [3]prijambodo@staff.ubaya.ac.id

Abstract

In the supply chain of foodstuffs derived from livestock, data consistency and security are the most important thing to pay attention to. However, the length of the existing supply chain and the lack of facilities to store and maintain the correctness of livestock data are problems in ensuring the security of the data. The use of blockchain in livestock has existed in Indonesia, but its scope is only to agriculture or only a small part of the livestock supply chain. Therefore, this research was conducted to develop the application of a blockchain-based livestock supply chain system in Indonesia. This research begins with conducting a literature study, analyzing the slaughterhouses. Then the development of a blockchain system for the food supply chain was carried out. System validation was carried out through interviews with the head of the slaughterhouses, breeders, and wholesalers. The results state that the system helps in ensuring the security of food supply chain data from livestock. However, the problem that has not been resolved is the validation process of data on the weight loss of foodstuffs in the course of the supply chain process. In the future, this technique is expected to be used for halal certification.

Keywords: livestock, food fraud, blockchain, trust, supply chain

1. Introduction

Currently, the system for recording livestock data, from farmers (upstream) to traders or end-users (downstream) in Indonesia, is still done conventionally because there is no Information System specifically created to handle this [1]. This condition causes the data to be prone to errors, and it ultimately affects the accuracy of the data received by each party who needs the livestock data. Moreover, the currently used system only records livestock health on farms and records fluctuations in livestock prices. Ramadhan et al. (2012) stated that a comprehensive livestock data monitoring system should be as crucial as other e-Government systems because this can affect the stability of livestock supply in Indonesia [2]. Furthermore, in other studies, a complete model of the livestock information system has been developed for use in Indonesia, starting from a list of each party involved and their duties to the system's flow being modeled [3].

However, the spread of livestock farms makes it difficult for government agencies to monitor or record data on the status of livestock spread in the market safely [4]. Currently, a system that integrates livestock to slaughterhouses, called *Rumah Potong Hewan* (RPH), has not yet been implemented in Indonesia so recording the distribution of livestock can potentially be less transparent [5]. Furthermore, according to BPS, the data on the distribution of livestock is still held by each farm and slaughterhouse, thus causing a lack of up-to-date data to determine the country's needs [6]. In addition, the disruption of the distribution channel for livestock in Indonesia makes it challenging to know the origin of products from livestock. Whereas consumers, such as restaurants or supermarkets, want to know the safety and clarity of their products.

The distribution of livestock that is not adequately recorded can also cause food fraud [7], [8]. Food fraud is an act of damaging or changing products along the supply chain [9]. An example of food fraud actions is changing the label of pork to beef. In addition, the individual can also increase the selling price of the product by labeling poor-quality products with high-quality product labels. Thus, consumers will be interested in buying because they have a high-quality product label even though they have a slightly higher price.

Security in the supply chain requires a system that can run transparently [10]. On the other hand, this system must also ensure that other parties cannot change data that certain parties have entered without the consent or knowledge of the initial data owner. This is done to avoid fraud in the supply chain [11]. Blockchain technology has a data storage system that cannot be changed and deleted when entered into a blockchain [12]. Furthermore, with the blockchain, tracking data can be done transparently [13]–[16] so that users can check the validity of a product by looking at the history of the product [17]. In 2019, Burke explained that blockchain could indeed be a precious asset when implemented in livestock supply chains. However, it is undeniable that blockchain still has some weaknesses to be implemented in real terms, such as the consensus process, which still takes a long time, and the need for machines with computing power for the miners to run [18]. Nevertheless, in 2021 Garaus and Treiblmaier revealed that customer trust in sellers increases when they can track food ingredients data through a blockchain-based supply chain system [19].

In 2018 and 2020, several researchers developed a blockchain application design that combines the Hyperledger Fabric platform with existing conventional infrastructure in China, including the Internet of Things (IoT) [20], [21]. This application aimed to track data on food ingredients, from farms to final consumers. This application was built with the lowest possible complexity and cost with the hope that it would be easy to implement. In the same year, other researchers also developed a livestock data tracking system by utilizing Radio Frequency Identification (RFID), cloud computing, and blockchain to increase the trust of its users [22]. RFID was used to receive data from various sensors on the farm then processed it on a cloud platform capable of high computing. The data in the cloud was also secured by applying the blockchain concept in data storage. Another study made the blockchain a reward mechanism for those who provided food ingredients data correctly from upstream to downstream [23]. This study also highlighted the weakness of storing livestock or food data, which retrieved data from physical objects via barcodes or QRCODE, whereas these media are effortless to cheat. Thus, in this study, the data stored was in the form of original photos of physical objects to be stored, which were read and processed using machine learning.

In a study conducted by Mao et al., blockchain in the food supply chain system was used as a tool to track data and used to monitor each party involved in the supply chain [24]. In this system, each party who conducted transactions on the network would provide a review of the seller. This review data would be processed using the Long Short Term Memory (LSTM) algorithm to analyze the tendency of the review to be good or bad. The results of this analysis can be seen by the regulator, which is the Government or policymaker, so that it can be used as a basis for fostering every party in the supply chain.

In 2020 Loke and Ann designed a food tracking system using blockchain. However, the difference with other research was that this system recorded the location for inputting each food data entering the supply chain system [25]. This research revealed that securing data by blockchain would be in vain if the data entered into the system had indeed been rigged from the beginning of the input. Therefore, the recording was also made from the data input location using the QRCODE tag, which would also be stored the ID in this food supply chain system in the system design developed by Loke and Ann.

In 2019 research conducted by Maghfirah, has tried to implement blockchain technology in agriculture in Indonesia using the HARA platform. However, this research has only tried to implement it in the agricultural sector, so the impact has not been seen if it is implemented in the livestock sector, especially in the livestock meat supply chain [26]. While in the field of animal husbandry itself through research that has been carried out by Sugihartanto and Hakim, HARA actually already has a project to develop blockchain for chicken farming [27]. Furthermore, research conducted by Afrianto et al in 2020 also confirmed that the implementation of blockchain in agriculture in Indonesia is not yet on a large scale, there are still many challenges that must be analyzed and thought about for solutions, because people in Indonesia are so dynamic. [28].

Asfarian et al in 2020 also conducted research on the factors that are important and must exist in a blockchain-based livestock system. Through this research, it was found that the blockchain system built must be easy to use by users who have low digital literacy, and must get support from every party, especially the government. This research has not yet developed a blockchain application that will be used in the livestock system, but only until the design of the system requirements [29].

Blockchain technology has been applied in the food supply chain, where in the research conducted, blockchain is used to ensure the halal nature of a food. This blockchain-based system includes users such as suppliers, distributors, wholesalers, and retailers [30]. This system has also been proven to be able to secure data properly, so that the quality of halal food is maintained. In this system, stakeholders are covered from suppliers to retailers, so there is still the potential for data inconsistency to occur from farmers to slaughterhouses.

A fairly complete study covering every stake holder in the livestock supply chain, from breeders, slaughterhouses, to buyers, has been carried out by

DOI: https://doi.org/10.29207/resti.v6i2.3937

Ramadhan et al in 2021 [3]. The system, called e-Livestock, is only at the modeling stage, so it has not yet been completed. the real application can be seen and cannot be tested directly on its users.

Previous research that conducted outside Indonesia considered to be quite complex to be implemented in Indonesia if the readiness of infrastructure in livestock in Indonesia is taken into account and compared to overseas farms, which are even familiar with IoT [18], [21]. Furthermore, the research that has been done in Indonesia, is still focused on agriculture, or on animal husbandry but on a narrow scale. Blockchain research on livestock that has been carried out in Indonesia and is quite complete, is also only at the modeling level, so it is still not possible to see the direct impact if it is implemented in the form of a real application. Thus, in this research, a blockchain-based livestock management information system was being developed and tested on users in Indonesia. However, most users tested were from the government side because a livestock supply chain system will only work well if regulations and the Government provide full support [31]. In addition, this is also done to assist government agencies in monitoring the spread of existing livestock and livestock farms and to see the Government's readiness to use blockchain-based systems.

2. Research Methods

The methodology used in this research was the Architecture-based Model, shown in Figure 1. The first stage in this model was defining the system to be created, and the internal researchers conducted it. The next stage was analyzing the direct user to understand the user's needs and compare them with the system design defined by the researcher. After that, architectural design and system design were made. However, before they were implemented, it was necessary to check the suitability of the design with the initial requirements, and this was an iteration that would continue to be carried out until the design as per the initial requirements [32]. This method was used to get much feedback from potential users since the system developed implemented many new things for potential users.

Data collection activities both for the needs of the initial analysis and iteration and for testing and evaluation were carried out using qualitative methods. The qualitative method chosen was an interview. This method was chosen because it could obtain detailed feedback from users. In addition, with the limited number of potential users available, the interview method was also more appropriate than using questionnaires. Another advantage of this method was related to the speed of data collection.

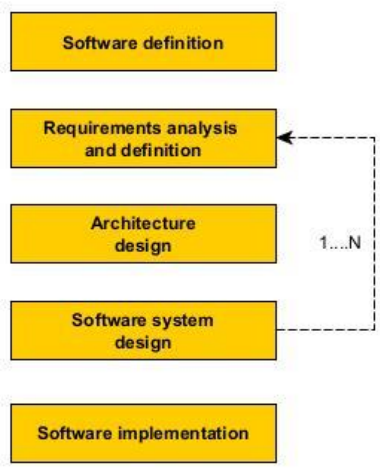

Figure 1. Architecture-based Model

In addition, the detailed data obtained from this method was appropriate to be used as an evaluation material for the system design being tested. Researchers could confirm user feedback directly if something were unclear [33]. This interview method could also reduce the iteration time needed in the Architecture-based model because understanding between researchers and users can be achieved more easily with direct interaction [34].

3 Results and Discussions

After doing the initial definition of the system to be created and analyzing the needs of its potential users, several results were obtained, which are parts of the system created. In the first part, the results were obtained from the analysis of current conditions, where these results were obtained through interviews with the head of the RPH Krian, Sidoarjo. In the second part, the data design was generated to be used as the data storage structure of this system. In the third part, the design of the process that would run on this system was produced, and the design of this process has been adapted to the flow used by the current livestock supply chain. In the fourth section, the results of experiments conducted by researchers on various system features were obtained. Finally, in the fifth section, the evaluation results were obtained by conducting interviews again with potential users. Prior to the interview, potential users were allowed to see the developed system while explaining the research team.

3.1 Analysis Results of the Current Condition

Through the results of the analysis carried out by direct observation, literature studies, and interviews with the head of the RPH, the results obtained are in the form of problems that have occurred or have the potential to occur at this time. The first problem was the incomplete data brought by farmers when slaughtering livestock at the RPH. This made data on the origin of livestock undetectable. In addition, livestock feeding has not been

DOI: https://doi.org/10.29207/resti.v6i2.3937

recorded by the livestock traders. This lack of various records was exacerbated by an integrated system between livestock farms, livestock traders, and RPHs. Therefore, the history of livestock may not be recorded at all or not recorded in detail from upstream to downstream. Another potential problem was data mismatch from one place to another so data integrity can be terrible.

The results of the analysis of the problems that occurred or had the potential to occur were then continued with the analysis that produced the minimum system requirements that must exist in the system being developed. The first part that must exist in the system is being developed to record complete and detailed data on livestock, starting from the birth of the cattle to the moment before the livestock is slaughtered at the RPH. The second part was to integrate livestock data with the RPH so that the RPH could track the condition of the livestock to be slaughtered. Finally, blockchain technology was implemented in each of these integrated data stores so that the data that has been entered can still be tracked but cannot be changed arbitrarily.

3.2 Data Design

Data design is the part that includes the formation of structures to store data on the system. In this developed system, data was not stored in a database like an ordinary information system but rather on blocks in the blockchain circuit. The program used to store or read data was Solidity, while the blockchain platform used was Ethereum. Data storage was carried out in the form of smart contracts, and there were three central contracts in this system, namely UserManager, LivestockManager, and SlaughterManager [35]. The UserManager contract can be seen in Table 1. The contract is used to hold the address of the user. When the contract is first sent to the Ethereum server, the sender's address will be set as admin. Through this UserManager contract, the system can find out the position or role of the user so that at the same time, it can protect the access rights of each user.

The UserManager contract managing its data will be connected to the user struct. There are userAddress, name, role, timeCreated, timeUpdated, and active in this structure. Struct user is used to storing user object data which will later be stored in the mapping. The complete structure of the struct user can be seen in Table 2.

The following contract is LivestockManager, which is used to manage livestock on the blockchain. In this contract, there are four structs, namely Livestock, WeightRecord, HealthRecord, and Cowshed. Struct Livestock is used to store livestock data, while WeightRecord is used to record the history of livestock weight and size. Livestock health history is stored in the HealthRecord struct, while Cowshed is used as a marker that the user has his cage for his livestock. Details of the LiveStockManager contract can be seen in Table 3.

Table 1. UserManager

Type	Name	Description
Attribute	admin: address	Saving admin address.
Attribute	users: mapping(address => User)	Mapping or storing data from users that have been entered into the struct.
Function	registerUser(address, string, uint)	Adding a user that will be stored in the users' mapping.
Function	getRole(address) uint	Taking the role from the address entered, then it will be returned to uint.
Function	checkRole()	Checking the role calling this function.
Function	onlyFarmer()	The modifier is used to limit the use of the Farmer role function.
Function	onlyStocker()	The modifier is used to limit the use of the Stocker role function.
Function	onlyButcher()	The modifier to limit the use of the Butcher role function.
Function	checkUser(address) bool	Checking whether there is a user or not.
Function	isAdmin(address) bool	Checking if the address entered is admin or not.
Function	toggleUser(address)	Changing the user's active status from active to inactive or vice versa.

Table 2. Struct user

Type	Name	Description
Attribute	userAddress: address	The userAddress attribute to find out the address of the user
Attribute	name: string	The name attribute to find out the name of the user.
Attribute	role: Role	The role attribute is to find out the position of the user.
Attribute	timeCreated: uint	The timeCreated attribute determines when the data was created based on the number of blocks formed.
Attribute	timeUpdated: uint	The timeUpdated attribute is used to determine when the data was updated based on the number of blocks formed.
Attribute	active: bool	The active attribute is used to determine whether the user already exists or not.

Struct Livestock is one of the most critical data storage places in this system because it contains all detailed data from livestock. The data stored in this struct includes farmIndex, lsId, fatherId, motherId, birthday, eartag, gender, wrCount, hrCount, transferCount, stateCount, status, timeUpdated, and timeCreated. Details of each attribute data can be seen in Table 4.

Next, the WeightRecord struct is used to store the history of the weight and size of each livestock in the system. Through this struct, the user can see the development of the livestock in terms of the weight and size of the livestock, so that tracking the condition of the livestock can be more easily done. Details of each attribute for WeightRecord can be seen in Table 5.

Table 3. LivestockManager

Type	Name	Description
Attribute	gobalLSCount: uint	This attribute is used to calculate all livestock in the system.
Attribute	cowsheds: mapping(address => Cowshed)	This attribute is used to store data from the cage.
Attribute	cowshedsAddress: address[]	This attribute is used to hold the address of the user's cage and to count the number of cages on the system
Attribute	livestocks: mapping(uint => Livestock)	Mapping or storing data from livestocks that have been entered into the struct.
Attribute	livestockRace: mapping(uint => Race)	This attribute is used to determine the race of livestock.
Attribute	livestockOwner: mapping(uint => address)	This attribute is used to determine the ownership of the livestock.
Attribute	livestockCounts: mapping(address => uint)	This attribute is used to determine the number of livestock owned by the owner.
Attribute	wRecords: mapping(uint => mapping(uint => WeightRecord))	This attribute is used to store data on livestock weight and body size.
Attribute	hRecords: mapping(uint => mapping(uint => HealthRecord))	This attribute is used to store health data from livestock.
Attribute	livestockStates: mapping(uint => mapping(uint => State))	This attribute is used to find out what state the livestock is in (farmer, stocker, butcher, beef)
Attribute	livestockTransfers : mapping(uint => mapping(uint => address))	This attribute is used to determine the history of the livestock movement.
Function	registerCowshed(address)	This function is used to add a new cage data and as a status that the account is active.
Function	registerLivestock(uint, uint, uint, string, uint)	This function is used to add new livestock data and can only be called by farmer's role.
Function	registerWRecord(uint, uint, uint, uint)	This function is used to add weight history to livestock.
Function	registerHRecord(uint, string, string, bool)	This function is used to add the health history of livestock.
Function	transfer(uint, address, address)	This function sends livestock from the initial owner to the next owner.
Function	changeState(uint, address)	This function changes the state from livestock to (farmer, stocker, butcher, beef).
Function	changeStatus(uint)	This function is used to change the status of livestock from alive to dead. Only farmers role can do it.
Function	stringToBytes32(string): return bytes32	This function is used to convert a string to bytes32.
Function	bytes32ToString(bytes32): return string	This function is used to convert bytes32 to string.

Table 4. Struct Livestock

Type	Name	Description
Attribute	farmIndex: uint	The farmIndex attribute is used to determine the index of the livestock contained in Cowshed's lsId array.
Attribute	lsId: uint	The lsId attribute is used to determine the id of the livestock.
Attribute	fatherId: uint	The fatherId attribute is used to find out the father id of the livestock.
Attribute	motherId: uint	The motherId attribute is used to find out the parent id of the livestock.
Attribute	birthDay: uint	The birthDay attribute is used to determine the date of birth of the livestock. Time is stored in the form of epoch.
Attribute	earTag: bytes	The earTag attribute is used to determine the eartag of livestock.
Attribute	gender: bool	The gender attribute is used to determine the sex of livestock.
Attribute	wrCount: uint	The wrCount attribute is used for iterations of the weightRecords mapping.
Attribute	hrCount: uint	The hrCount attribute is used for iterations of the healthRecords mapping.
Attribute	transferCount: uint	The transferCount attribute is used for iterations of the livestockTransfers mapping.
Attribute	stateCount: uint	The stateCount attribute is used for iterations of the livestockStates mapping.
Attribute	status: bool	The status attribute is used to determine whether the livestock is still alive or not.
Attribute	timeUpdated: uint	The timeUpdated attribute determines when the data was created based on the number of blocks formed.
Attribute	timeCreated: uint	The timeCreated attribute determines when the data was created based on the number of blocks formed.

The health history of livestock is also stored in this system. Therefore, tracking of the health condition of livestock can also be easily seen, and this data is a guarantee of the quality of the livestock to be marketed. The stored livestock health data structure can be seen in Table 6.

Table 5. Struct WeightRecord

Type	Name	Description
Attribute	actor: address	The actor attribute is used to find out the person doing the checking.
Attribute	lsId: uint	The lsId attribute is used to determine the id of the livestock.
Attribute	weight: uint	The weight attribute is used to determine the weight of livestock.
Attribute	length: uint	The length attribute is used to determine the length of the livestock.
Attribute	heartGrith: uint	The heartGrith attribute is used to determine the chest circumference of livestock.
Attribute	timeRecord: uint	The timeRecord attribute is used to store the date the record was entered.
Attribute	timeCreated: uint	The timeCreated attribute determines when the data was created based on the number of blocks formed.

Table 6. Struct HealthRecord

Type	Name	Description
Attribute	actor: address	The actor attribute is used to find out the person doing the checking.
Attribute	lsId: uint	The lsId attribute is used to determine the id of the livestock.
Attribute	sick: bool	The sick attribute is used to determine whether the livestock is sick or not.
Attribute	description: bytes	The description attribute is used to describe the disease suffered by livestock.
Attribute	action: bytes	The action attribute is used to describe the action performed.
Attribute	timeRecord: uint	The timeRecord attribute is used to store the date the record was entered.
Attribute	timeCreated: uint	The timeCreated attribute to determine when the data was created based on the number of blocks formed.

The last struct in LivestockManager is Cowshed. This section is used to store additional data from the user and check user activity. The user's active status will determine the livestock data that the user can store because only active users are allowed to add livestock data to this system. Details of the Cowshed struct can be seen in Table 7.

Table 7. Struct Cowshed

Type	Name	Description
Attribute	ownerId: address	The ownerId attribute is used to find out the address or address of the owner.
Attribute	lsId: uint	The lsId attribute is used to store the id of the livestock. So if the livestock does not belong to the owner, it will be removed from this array.
Attribute	timeCreated: uint	The timeCreated attribute is used to determine when the data was created based on the number of blocks formed.
Attribute	timeUpdated: uint	The timeUpdated attribute is used to determine when the data was created based on the number of blocks formed.
Attribute	status: bool	Status attribute to know whether cowshed already exists or not.

The third or final contract is the SlaughterManager, which is used to store various data related to the slaughtering process. There are two parties who will be closely related to this contract, namely the stocker and the butcher. The stocker itself is the party that will receive the livestock from the breeder and then makes a request for slaughtering the livestock at the RPH. Meanwhile, the butcher is the party who slaughters the livestock until it sells the results of the slaughter of the livestock. Details of the SlaughterManager contract can be seen in Table 8.

Table 8. SlaughterManager

Type	Name	Description
Attribute	beefCount: uint	The beefCount attribute is used for iterations of the beef mapping.
Attribute	beefs: mapping(uint => Beef)	The beefs attribute is a mapping used to store data processing livestock into beef.
Attribute	livestockBeefStatus: mapping(uint => bool)	This attribute stores whether livestock with this id is processed into beef.
Attribute	livestockHasBeefs: mapping(uint => mapping(uint => uint)	This attribute stores the beef ID for livestock that have been processed into beef.
Attribute	livestockBeefCounts: mapping(uint => uint)	This attribute is used to calculate how many livestock are processing beef.
Attribute	beefApprorval: mapping(uint => address)	The beefApproval attribute is a mapping used to designate the RPH to process livestock into beef.
Attribute	beefApproveds: mapping(uint => mapping(uint => uint)	This attribute is used to store the processing of livestock that have been received completely.
Attribute	approveCount: uint	Atribut ini digunakan untuk iterasi beefApproveds.
Attribute	beefDenieds: mapping(uint => mapping(uint => uint)	Atribut ini digunakan untuk menyimpan permrosesan hewan ternak yang ditolak oleh RPH.
Attribute	deniedCount: uint	Atribut ini digunakan untuk iterasi beefDenieds.
Function	registerBeef(uint, address)	This function is used to apply for processing livestock into beef to RPH.
Function	checkAntemortem (uint, uint, bool, string)	This function is used to store the results of the antemortem checking process.

Function	checkPostmortem(uint, uint, bool, string)	This function is used to store the results of the postmortem checking process.
Function	packingBeef(uint, uint)	This function is used to change the status of the beef that has been packed or finished.
Function	slaughtering(uint)	This function is used to change the status of livestock to dead or slaughtered.

In the SlaughterManager contract, there is only one struct, namely struct beef. This structure is used to record livestock that is ready to be processed into beef. The stocker can only do the making of this beef struct data because it is the stocker who requests the slaughter of livestock to the RPH. Details of struct beef can be seen in Table 9.

Table 9. Struct beef

Type	Name	Description
Attribute	beefId: uint	This attribute is used to find the id of beef.
Attribute	lsId: uint	This attribute is used to indicate the id of the livestock.
Attribute	dateAnte: uint	The dateAnte attribute is used to store the date of the antemortem check. Time saved into epoch.
Attribute	datePost: uint	The datePost attribute is used to store the postmortem check date. Time saved into epoch livestock.
Attribute	datePack: uint	The datePack attribute is used to store the wrapping date of the beef. Time saved into epoch.
Attribute	ante: bool	The ante attribute is used to indicate whether the antemortem check was accepted or not.
Attribute	post: bool	The post attribute is used to indicate whether the postmortem check was accepted or not.
Attribute	slaughterHouse: address	The slaughterHouse attribute is used to store the address of the slaughter that will process livestock into beef.
Attribute	desc: bytes	The desc attribute describes the condition if the livestock is rejected in a certain process.
Attribute	timeCreated: uint	The timeCreated attribute stores the request date and is saved into the epoch.
Attribute	timeUpdated: uint	The timeUpdated attribute stores the request change date and is stored in the epoch.

3.3 Design Process

The primary process designed in this system is the transfer of livestock. This process will record the movement of livestock from one user to another. This process will utilize the blockchain with the Ethereum platform so that any party can no longer change the data that has been entered. Thus, it can be used to track livestock data and the status of each of these livestock. Details of the design of this process are shown in Figure 2.

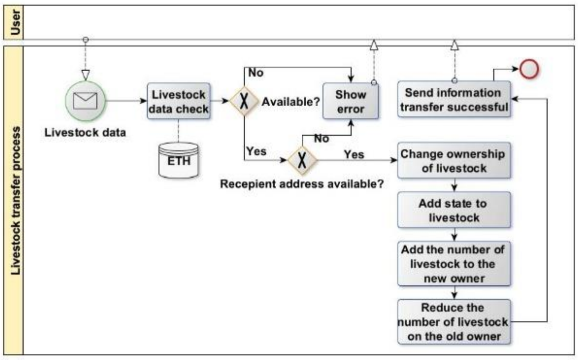

Figure 2. Process design for livestock transfer

Figure 2 shows that the process starts from the livestock data that the user wants to transfer. This data is taken from the ETH blockchain, and it is checked whether the data exists. If it does, the step will proceed to check the recipient's address. If not, a notification message will be given. Furthermore, if the recipient's address is not valid, a notification message will also be given. If it is valid, it will proceed to change the ownership of the livestock. The process of changing ownership is followed by the process of changing the status of livestock, reducing the number of livestock to the sender, and increasing the number of livestock to the recipient. After all these processes run, the process will end. Once the data has been processed, changes can no longer be made because it has been entered into the blockchain.

3.4 Test Results

The test was carried out by running the existing system, using data from the user, but within the scope of a simulation. When starting to use the application, each user must register with the existing system. Since this system was developed on a blockchain-based basis for data storage, every user who wants to register must have an address to connect to the blockchain first. The process of forming this user address uses a third-party application called Metamask. After the user gets the address, an admin will verify and then immediately enter the user's address into the blockchain system that was created. The display of the stored registrant data can be seen in Figure 3. The figure shows that the identity of each user was in the form of a hash string.

DOI: https://doi.org/10.29207/resti.v6i2.3937

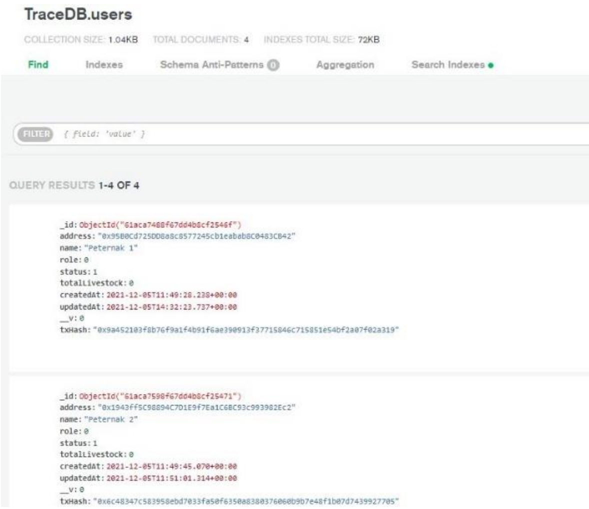

Figure 3. User's data

The address obtained from the Metamask application will appear on the registration page, which can be accessed by the admin of this system. In this system, the admin will act like a miner who can verify the data that will enter the blockchain network. After the admin has processed the addition of a user to the blockchain network, the Metamask application will display a confirmation page, as shown in Figure 4. The confirmation page will also display details of the data added to the blockchain network. However, it is necessary to remember that the nominal ETH is not used for buying and selling transactions but only as a marker that someone can enter the blockchain network.

After the addition of users to the blockchain network is confirmed, a successful notification of the addition of the data will appear, as shown in Figure 5. Furthermore, the data added to the blockchain network cannot be changed again. Changes to data on the blockchain network can only be done by adding new data, which is a revision of the previous data. This ability causes blockchain-based systems to be able to track data more efficiently.

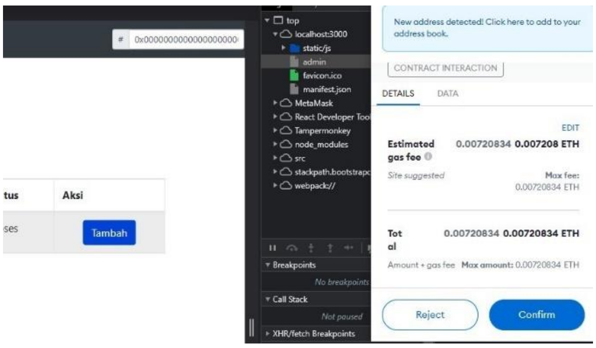

Figure 4. Confirmation of Metamask

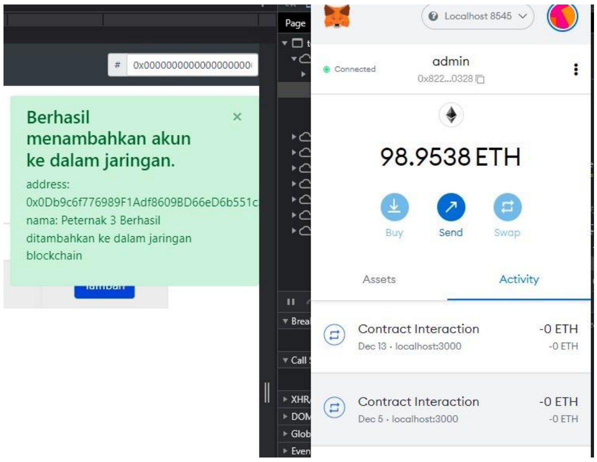

Figure 5. Notification of data added

If the user wants to transfer livestock, the user will face a livestock transfer form, as shown in Figure 6. This form contains data that must be filled in, such as the livestock that will be transferred by the user, along with livestock data such as weight, chest circumference, and length. In addition, the user must also enter the hash address of the recipient so that only users who have been actively registered in the blockchain network can be the recipient.

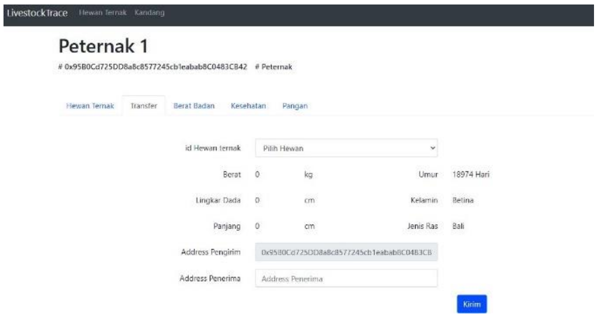

Figure 6. Livestock transfer form

Users can also enter the history of the livestock they own, ranging from the size of the livestock body, the health of the livestock, and the type of food from the livestock. The appearance of the form for each historical record is relatively the same.

Figure 7 shows an example of the historical record form for the body size of livestock. In this form, there is only attribute data related to the livestock, and no hash address has been requested. This is because, for history storage, it will not be entered directly into the blockchain network. Instead, new historical data will be entered into the blockchain network when the cattle are transferred from one party to another. This mechanism is because the data security needed is when the data is transferred from one party to another in a livestock supply chain. Therefore, when the data is only stored by the user who is the owner of the livestock itself and does not involve other people, the data does not need to be entered into the blockchain network.

DOI: https://doi.org/10.29207/resti.v6i2.3937

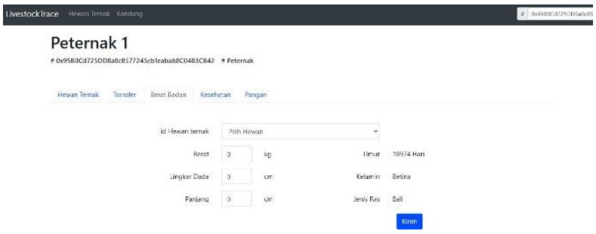

Figure 7. Livestock size history form

Other processes will be processed into the blockchain network, such as adding livestock. This is because the addition of livestock will be considered livestock data transfer from the Genesis block to users already on the blockchain network. The Genesis block itself is the first block that will appear automatically when a blockchain network is created [12], [36]. In addition to all these processes, many other processes include sending livestock to the slaughterhouse, stockers, or between users. However, they all have the exact mechanism when adding new users to the blockchain network and when transferring livestock. Thus, each delivery must fill in the recipient's address in the form of a hash of the recipient on the blockchain network. After that, it ends with confirmation via metamask and waits for the data to enter the blockchain network successfully. Each user can also see the livestock he owns and track livestock moving from or to that user.

3.5 Evaluation

The evaluation was carried out by conducting interviews with the Head of the RPH, farmers, and wholesalers who had been shown and explained how the developed system worked. The results of this evaluation were that all parties agreed that the system developed would be very helpful in recording and tracking livestock data. This was the impact of data integration between each party involved through this system. Thus, the flow of data for each livestock can be monitored, tracked, and guaranteed the truth. In addition, recording features such as livestock size history, health, and livestock feed allows the system to help provide accurate data if users face problems that arise with their livestock. The blockchain technology that has been implemented and explained to its potential users has also received a good response because it is believed to be able to increase the security of the data entered into this system.

4 Conclusion

Through the results of the design, development, testing, and evaluation that has been carried out, it was found that this system was able to handle the problems that occurred related to the integrity of livestock data in the livestock supply chain in Indonesia. The problem is the data retrieval process to ensure the correctness of the data on food ingredients from livestock sold, including type, size and price.

With good data integration, the trust of the livestock industry players can be increased, and the comfort of end consumers in consuming food ingredients from livestock is also guaranteed. In addition, this research is expected to be the basis for determining halal certification for food ingredients derived from livestock. The problem that has not been addressed in this system is the process of depletion of food ingredients from livestock that occurs during the delivery.

The digital literacy of each user can be one of the inhibiting factors for this system to be widely implemented. Thus, further development of this research requires a broader trial to see the percentage of potential users throughout Indonesia who can understand and accept the implementation of this blockchain-based system. In addition, feedback from each trial can be used to make the system more suitable for the conditions throughout Indonesia.

Reference

[1] D. Prasetyani, D. T. Ardianto, and A. A. Firdaus, “Integration of livestock supply chain strategy as part of the creative economy and creative industry in Indonesia: Literature review,” in *IOP Conference Series: Earth and Environmental Science*, 2021, vol. 902, no. 1, doi: 10.1088/1755-1315/902/1/012051.

[2] A. Ramadhan, Muladno, D. I. Sensuse, and A. M. Arymurthy, “E-Livestock: Its definition for indonesia,” *Eur. J. Sci. Res.*, vol. 84, no. 2, pp. 304–327, 2012. https://www.researchgate.net/profile/Arief-Ramadhan-2/publication/342411079_EJSR_84_2_15_Arief_Muladno_Dana_Aniatipdf/data/5ef308f2a6fdcc158d260918/EJSR-84-2-15-Arief-Muladno-Dana-Aniati.pdf

[3] A. Ramadhan, A. M. Arymurthy, D. I. Sensuse, and Muladno, “Modeling e-Livestock Indonesia,” in *Heliyon*, 2021, vol. 7, no. 8, p. e07754, doi: 10.1016/j.heliyon.2021.e07754.

[4] K. Direktorat Jenderal Peternakan dan Kesehatan Hewan, “Populasi Sapi Potong menurut Provinsi (Ekor), 2018-2020,” 2020. https://www.bps.go.id/indicator/24/469/1/populasi-sapi-potong-menurut-provinsi.html (accessed Feb. 13, 2022).

[5] S. Ihwan, A. Usman, and P. Bw, “Model Sistem Pencatatan Digital Pada Peternakan Sapi Nasional Melalui Implementasi Teknologi Agromilenial Sebagai Solusi Jitu Tercapainya Swasembada Sapi di Indonesia S Ihwan, A Usman, B W Putra,” in *Seminar Nasional Teknologi Peternakan dan Veteriner*, 2019, pp. 292–303, doi: 10.14334/pros.semnas.tpv-2019-p.292-303.

[6] Subdirektorat Statistik Peternakan, “Peternakan Dalam Angka,” 2019. http://pusvetma.ditjenpkh.pertanian.go.id/upload/statistik/1589426278.Buku_Statistik_2019.pdf

[7] A. Wisniewski and A. Buschulte, “How to tackle food fraud in official food control authorities in Germany,” *J. fur Verbraucherschutz und Leb.*, vol. 14, no. 4, pp. 319–328, 2019, doi: 10.1007/s00003-019-01228-2.

[8] A. CARTÍN-ROJAS, “Food fraud and adulteration: a challenge for the foresight of Veterinary Services,” *Rev. Sci. Tech. l’OIE*, vol. 36, no. 3, pp. 1015–1024, 2017, doi: 10.20506/rst.36.3.2733.

[9] D. Grace, “Food Fraud,” *Encycl. Food Secur. Sustain.*, vol. 1, pp. 238–248, 2019, doi: 10.1016/B978-0-08-100596-5.21577-1.

[10] F. Yiannas, “A New Era of Food Transparency Powered by

Blockchain," *Innov. Technol. Governance, Glob.*, vol. 12, no. 1–2, pp. 46–56, 2018, doi: 10.1162/inov_a_00266.

[11] C. Elliott, "Elliott Review into the integrity and assurance of food supply networks-Final report: A national food crime prevention framework," no. September, 2014. http://www.rafa2015.eu/pdf/Speakers_pdf/Chris%20Elliott.pdf

[12] N. Satoshi, "Bitcoin: A Peer-to-Peer Electronic Cash System," *J. Gen. Philos. Sci.*, vol. 39, no. 1, pp. 53–67, 2008, doi: 10.1007/s10838-008-9062-0.

[13] M. M. H. Onik, C. S. Kim, N. Y. Lee, and J. Yang, "Privacy-aware blockchain for personal data sharing and tracking," *Open Comput. Sci.*, vol. 9, no. 1, pp. 80–91, 2019, doi: 10.1515/comp-2019-0005.

[14] R. Neisse, G. Steri, and I. Nai-Fovino, "A blockchain-based approach for data accountability & provenance tracking," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1305, 2017, doi: 10.1145/3098954.3098958.

[15] I. Abu-elezz, A. Hassan, A. Nazeemudeen, M. Househ, and A. Abd-alrazaq, "The benefits and threats of blockchain technology in healthcare: A scoping review," *Int. J. Med. Inform.*, vol. 142, no. February, p. 104246, 2020, doi: 10.1016/j.ijmedinf.2020.104246.

[16] P. Helo and A. H. M. Shamsuzzoha, "Real-time supply chain—A blockchain architecture for project deliveries," *Robot. Comput. Integr. Manuf.*, vol. 63, no. December 2019, p. 101909, 2020, doi: 10.1016/j.rcim.2019.101909.

[17] R. Bhat and I. Jõudu, *Emerging issues and challenges in agri-food supply chain*. 2019. https://dspace.emu.ee/xmlui/bitstream/handle/10492/6087/Elsevier%20chapter%20DRAFT%20Final%20copy%20Bhat.pdf?sequence=1&isAllowed=y

[18] T. Burke, "Blockchain in Food Traceability," *Food Traceability, Food Microbiol. Food Saf.*, pp. 133–143, 2019, doi: 10.1007/978-3-030-10902-8.

[19] M. Garaus and H. Treiblmaier, "The influence of blockchain-based food traceability on retailer choice: The mediating role of trust," *Food Control*, vol. 129, no. March, p. 108082, 2021, doi: 10.1016/j.foodcont.2021.108082.

[20] A. Iftekhar, X. Cui, M. Hassan, and W. Afzal, "Application of Blockchain and Internet of Things to Ensure Tamper-Proof Data Availability for Food Safety," *J. Food Qual.*, vol. 2020, 2020, doi: 10.1155/2020/5385207.

[21] M. Guo, X. J. Liu, and W. Zhang, "Using blockchain technology in human food chain provenance," *WIT Trans. Built Environ.*, vol. 179, pp. 391–396, 2018, doi: 10.2495/UG180361.

[22] L. Yang, X. Y. Liu, and J. S. Kim, "Cloud-based Livestock Monitoring System Using RFID and Blockchain Technology," in *Proceedings - 2020 7th IEEE International Conference on Cyber Security and Cloud Computing and 2020 6th IEEE International Conference on Edge Computing and Scalable Cloud, CSCloud-EdgeCom 2020*, 2020, pp. 240–245, doi: 10.1109/CSCloud-EdgeCom49738.2020.00049.

[23] S. K. Lo *et al.*, *Digital-Physical Parity for Food Fraud Detection*, vol. 11521 LNCS. Springer International Publishing, 2019. https://www.researchgate.net/profile/Sin-Kuang-Lo/publication/333859990_Digital-Physical_Parity_for_Food_Fraud_Detection/links/5d282bbe92851cf4407b0530/Digital-Physical-Parity-for-Food-Fraud-Detection.pdf

[24] D. Mao, F. Wang, Z. Hao, and H. Li, "Credit evaluation system based on blockchain for multiple stakeholders in the food supply chain," *Int. J. Environ. Res. Public Health*, vol. 15, no. 8, 2018, doi: 10.3390/ijerph15081627.

[25] K. S. Loke and O. C. Ann, "Food Traceability and Prevention of Location Fraud using Blockchain," *IEEE Reg. 10 Humanit. Technol. Conf. R10-HTC*, vol. 2020-Decem, pp. 1–5, 2020, doi: 10.1109/R10-HTC49770.2020.9356999.

[26] A. Maghfirah, "Blockchain in Food and Agriculture Supply Chain," *Int. J. Food Beverage Manuf. Bus. Model.*, vol. 4, no. 2, pp. 53–66, 2019, doi: 10.4018/ijfbmbm.2019070104.

[27] M. Faiz Sugihartanto and N. Silmina Hakim, "Challenges of Enhancing Halal Food Supply Chain Traceability Using Blockchain Technology: A Case Study From a Start-Up In Indonesia," in *Enhancing and Strengthening Halal Industry Through Science*, 2021, no. July, pp. 44–47. https://www.researchgate.net/profile/Mumtaz-Rabbani/publication/355844156_Virtual_International_Halal_Science_Conference-_PROCEEDINGS_2021/links/6180f118eef53e51e11a7edb/Virtual-International-Halal-Science-Conference-PROCEEDINGS-2021.pdf#page=56

[28] I. Afrianto, T. Djatna, Y. Arkeman, I. Sukaesih Sitanggang, and I. Hermadi, "Disrupting Agro-industry Supply Chain in Indonesia with Blockchain Technology: Current and Future Challenges," *2020 8th Int. Conf. Cyber IT Serv. Manag. CITSM 2020*, 2020, doi: 10.1109/CITSM50537.2020.9268872.

[29] A. Asfarian, K. I. Hilmi, and I. Hermadi, "Preliminary user studies on consumer perception towards blockchain-based livestock traceability platform in indonesia: An implication to design," *2020 Int. Conf. Comput. Sci. Its Appl. Agric. ICOSICA 2020*, 2020, doi: 10.1109/ICOSICA49951.2020.9243217.

[30] I. Surjandari, H. Yusuf, E. Laoh, and R. Maulida, "Designing a Permissioned Blockchain Network for the Halal Industry using Hyperledger Fabric with multiple channels and the raft consensus mechanism," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-020-00405-7.

[31] M. Tripoli and J. Schmidhuber, "Optimising traceability in trade for live animals and animal products with digital technologies," *Rev. Sci. Tech.*, vol. 39, no. 1, pp. 235–244, 2020, doi: 10.20506/rst.39.1.3076.

[32] J. Yu, "Research Process on Software Development Model," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 394, no. 3, 2018, doi: 10.1088/1757-899X/394/3/032045.

[33] M. Al Kilani and V. Kobziev, "An Overview of Research Methodology in Information System (IS)," *OALib*, vol. 03, no. 11, pp. 1–9, 2016, doi: 10.4236/oalib.1103126.

[34] A. Al-Azri, Z. Al-Salti, and W. Al-Karaghouli, "The successful implementation of e-government transformation: A case study in Oman," in *Proceedings of the European, Mediterranean and Middle Eastern Conference on Information Systems: Global Information Systems Challenges in Management, EMCIS 2010*, 2010, no. October 2014. https://bura.brunel.ac.uk/bitstream/2438/4308/1/C63.pdf

[35] F. Hendaryawan, "Sistem Informasi Manajemen Peternakan Berbasis Blockchain," Universitas Surabaya, 2022.

[36] S. Myung and J. H. Lee, "*Ethereum smart contract-based automated power trading algorithm in a microgrid environment," *J. Supercomput.*, 2018, doi: 10.1007/s11227-018-2697-7.