

CREATION A BLOCKCHAIN-BASED PRODUCT QUALIFICATION SYSTEM FOR E-MARKETPLACE USING PRACTICAL BYZANTINE FAULT TOLERANCE ALGORITHM

Irvan Nizuar*¹⁾, Linda Handayani²⁾, Sunny Arief Sudiro³⁾

1. Gunadarma University, Jakarta, Indonesia
2. Gunadarma University, Jakarta, Indonesia
3. Gunadarma University, Jakarta, Indonesia

Article Info

Keywords: Blockchain; Product Qualification; Hyperledger Fabric; PBFT; Consortium.

Article history:

Received 1 May 2023

Revised 15 May 2023

Accepted 29 May 2023

Available online 1 December 2023

DOI :

<https://doi.org/10.29100/jipi.v8i4.4001>

* Corresponding author.

Irvan Nizuar

E-mail address:

linda.handa20@gmail.com

ABSTRACT

An indicator of e-marketplace effectiveness is to facilitate transparent transactions between sellers and buyers by offering good products and adequate infrastructure to handle system requirements. A good product must be in accordance with the product qualification which consists of the classification of feasibility or not, and the benefits of the product. However, this often has problems including those caused by frequent problematic system conditions, non-transparent product qualifications, undistributed data, human error until data security. The researcher proposes the creation of a product qualification system for e-marketplace using blockchain technology. The product qualification system consists of data verification and validation carried out by stakeholders involved in the decentralized network. Transactions that have been executed and validated will be stored in the distributed ledger. The focus of this research is the implementation of blockchain that is allowed to use the PBFT algorithm on product qualification records in the e-marketplace. The purpose of this research is to create a blockchain-based product qualification system for e-marketplace that is transparent and immutable and the successful performance of the system in information storage and management security. The results show the creation of a web-based apps product qualification system that implements a blockchain using Hyperledger Fabric and the performance obtained through system testing and data validation testing.

I. INTRODUCTION

Performance and efficiency of the technology infrastructure in the form of hardware, software and telecommunications have continued to improve in the last few decades. Moore's law, based on Gordon Moore's observations in 1965 and later adjustments in 1975, states that the size of transistors shrinks so rapidly that twice as many as every two years go into a single computer chip [1]. Operating systems, programming languages and apps that function as controls for computer systems make significant updates. As well as telecommunications networks that will be released in 2020 [2]. These advances have revolutionized many aspects of our social life, one of which is in the field of commerce.

Commerce in Indonesia has utilized technological infrastructure in carrying out the buy and sell business process. E-commerce is a process of buying or selling products electronically by consumers and from company to company with computers and intermediary for business transactions [3]. Some of these e-commerce companies provide space for other business actors to display their products on the electronics website. This has led to the creation of an e-market called an e-marketplace. Every seller on the e-marketplace can display their products for sale without having to build a system. The seller only needs to use the facilities in the e-marketplace in the form of information and product quality, price, delivery, payment and product reviews.

An indicator of e-marketplace effectiveness is to facilitate transparent transactions between sellers and buyers by offering good products and adequate infrastructure to handle system requirements. A good product must be in accordance with the product qualifications which consists of classification of feasible or not and the benefits of the product [4]. However, this often has problems, including caused by frequent system conditions, non-transparent products, undistributed data, human error until data security.

Centralized systems have common drawbacks. Data stored centrally has a central point of failure which computer crackers can exploit. These systems are handled by a single organization, so that data can be secretly

manipulated by those with administrative access to the database [5]. The log history can be used as a trace for a human error monitoring system. centralized file log storage is not recommended because if the file is deleted, there is no record of action. This is a problem that occurs in the database [6]. Recent developments can solve this problem. Blockchain is a 2020 digital technology trend that has advantages in new strategies and business models, better security, better speed than the current system, better transparency and lower spending [7]. Blockchain is a distributed ledger technology based on the principles of peer-to-peer networking and cryptographic primitives [8]. Blockchain can be used outside of cryptocurrency and is starting to explore how blockchain can be used for existing systems, including in the processing of data records in the e-marketplace system.

Ching-Nung Yang et al s research in 2019 proposed a blockchain-based product scoring system to handle big data and business models. blockchain technology used to accelerate product assessment verification and maintain product data security. Product assessment system based on product quality scores. The product appraisal business model consists of suppliers providing product alliances that include e-commerce and trusted organizations such as international consumers, non-profit organizations for product appraisal. Data product verification and appraisal is written into new blocks on the blockchain via smart contract. Buyers can buy products with high value to be insured for product quality [12].

Yi Weng Chan research in 2019 developed a blockchain-based e-marketplace. Researchers made use of blockchain technology to develop a decentralized e-marketplace platform that was already installed. Researchers use smart contracts to secure deposit transactions and process payments. Every transaction is verified via the blockchain and recorded to a decentralized ledger. Transactions are carried out without a third party because the smart contract is self-executed. Smart contracts can make credible transactions and transactions on the blockchain are traceable and irreversible. Buyers and sellers can't break the contract. The processes recorded on the blockchain include product launches, purchases, deliveries and payments [13].

Wang and Hannafin's research in 2005 defined design-based research (DBR) as a research method that focuses on exploring systematic but flexible techniques targeted at improving practice through iterative analysis, design, development, and implementation requiring collaboration between researchers and practitioners and new useful principles. Design-based research is concerned with research design and methodology that provides direct, measurable alternative approaches and simultaneous improvements in research, theory and practice [14].

This research uses the object of identification analysis based on the workflow and product record from e-marketplace. Analyse the identification of participants or stakeholders who are involved in the decentralized network. Participants add data to the blockchain and send it to the network. Transaction requests are taken by miners or validators. The compiled and validated transactions will be sorted and packaged into a candidate block which is timed. Transactions will be stored in a general ledger and distributed over the network.

Hyperledger is a blockchain platform managed by the Hyperledger Linux Foundation project [9]. The platform was built for a consortium of businesses interested in building and deploying their blockchain applications with shared data. The Hyperledger platform used to build the blockchain is the Hyperledger Fabric with data storage transactions in the form of a chaincode. Chaincode is a program code on a network that functions like a smart contract on other blockchains [10]. The Practical Byzantine fault Tolerance algorithm is a consensus protocol used to secure blockchain networks, The advantages of the PBFT algorithm are at the time of the final transaction, there is no waiting period to ensure the transaction is safe and can reach network consensus without requiring energy-intensive calculations [11].

This research focuses on blockchain implementation using the Practical Byzantine Fault Tolerance algorithm on product qualification records in e-marketplace run on a decentralized system and stored on a distributed ledger. The creation of a web-based system uses chaincode to ensure transparent, secure and immutable product data as well as the safety of authorized participants using Hyperledger Fabric.

II. METHOD

This research adopts a design-based research methodology at the blockchain implementation stage in creating product qualification systems. The stages of implementing blockchain into the system using Hyperledger Fabric platform are as follows: set up the blockchain network on docker, set up the structure Go, registering admin and organization, ledger query and blockchain transactions using the PBFT algorithm.

A. Set up the Blockchain Network on Docker

Researchers used chainhero and docker configurations to build virtual computers that would handle different roles on the blockchain. Hyperledger fabric requires multiple certificates to ensure encryption during end to end processes (TLS, block signing authentication and other processes).

```
version: '2'
networks:
  default:
services:
  orderer.hf.chainhero.io: ...
  ca.org1.hf.chainhero.io: ...
  peer0.org1.hf.chainhero.io: ...
  peer1.org1.hf.chainhero.io: ...
```

In the Docker configuration, researchers define several services, including:

1) *Orderer*

The order node places order transactions together with other nodes that make up the ordering service. The fabric design relies on a deterministic consensus algorithm, that is, each block validated by the peer as generated by the ordering service is guaranteed to be complete and correct. The order list is used to send transactions and requests for channel creation or updates. Currently only one order is required. If more than one is defined then the one used by the SDK.

2) *Certificate Authorities (CA)*

The CA issues certificates for identification purposes on fabric networks. The provision of certificates is done in a separate process outside the network runtime. Fabric CA is a special certificate authority that provides a REST API for dynamic management certificates (enrol, revoke, re-enrol).

3) *Peers*

List of peers to which various requests, support, queries and event listener registrations will be sent.

B. *Set up the Structure Go*

Researchers use the Go language structure in managing program code. The Go structure arrangement can be seen in Figure3.

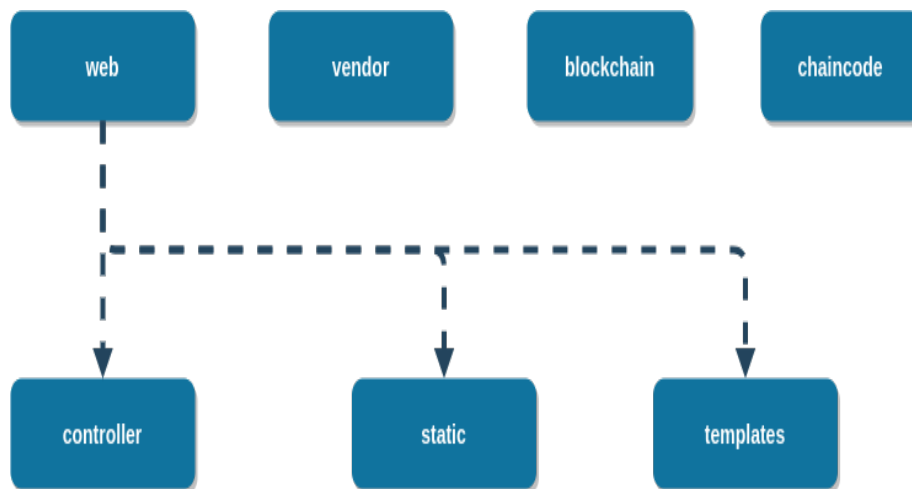


Figure 3: Structure Go

C. *Register Admin and Organization*

Researchers register admins and organizations used for access to the blockchain system to make transactions. Admin and participants that have been created will be configured in the Hyperledger Fabric.

```
certificateAuthorities:
  ca.org1.hf.chainhero.io:
    url: http://localhost:7054
    httpOptions:
      verify: false
    registrar:
      enrollId: admin
      enrollSecret: adminpw
```

```
caName: ca.org1.hf.chainhero.io
```

D. Ledger Query

Queries are needed to perform data transactions on the ledger as needed. The QueryRecord function is to invoke (“invoke”) a query. This function will return the response result of the query record from the chaincode.

```
func (setup *FabricSetup) QueryRecord() ([]byte, error) {  
  
    // Prepare arguments  
    var args []string  
    args = append(args, "invoke")  
    args = append(args, "query")  
    args = append(args, "record")  
  
    response,err:= setup.client.Query(channel.Request{  
        ChaincodeID: setup.ChainCodeID,  
        Fcn: args[0],  
        Args: [][]byte{[]byte(args[1]), []byte(args[2])}  
    })  
    if err != nil {  
        return nil, fmt.Errorf("failed to query: %v", err)    }  
        return response.Payload, nil  
    }  
}
```

The following is the InvokeRecord function which functions to perform data transactions in the chaincode. The prepared argument is called for the ledger key record. The eventId declaration and data description for the invoke request then register the event chaincode and execute the transaction request.

```
func (setup *FabricSetup) InvokeRecord(value string) (string, error) {  
  
    // Prepare arguments  
    var args []string  
    args = append(args, "invoke")  
    args = append(args, "invoke")  
    args = append(args, "record")  
    args = append(args, value)  
  
    eventID := "eventInvoke"  
    // Add data that will be visible in the proposal, like a description of the invoke request transientDataMap :=  
    make(map[string][]byte)  
    transientDataMap["result"] = []byte("Transient data in log invoke")  
    reg,notifier,err:= setup.event.RegisterChaincodeEvent(setup.ChainCodeID, eventID)  
    if err != nil {  
        return "", err  
    }  
    defer setup.event.Unregister(reg)
```

```
// Create a request (proposal) and send it
response,err:= setup.client.Execute(channel.Request{
ChaincodeID: setup.ChainCodeID,
  Fcn: args[0],
  Args: [][]byte{
    []byte(args[1]),
    []byte(args[2]),
    []byte(args[3])
  },
  TransientMap: transientDataMap
})
if err != nil {
  return "", fmt.Errorf("failed to move funds: %v", err)
}

// Wait for the result of the submission
select {
  case ccEvent := <-notifier:
    fmt.Printf("received CC event: %+v\n", ccEvent)
  case <-time.After(time.Second * 20):
    return "", fmt.Errorf("did NOT receive CC event for eventId(%s)", eventId)
}
return string(response.TransactionID), nil }
```

E. Blockchain Transactions Using the PBFT Algorithm

The Practical Byzantine Fault Tolerance algorithm is used to model the service as a state machine that is replicated at the node for ledger distribution. Send requests made by participants to carry out transactions to the main service. Main multicast requests to backup. The replica executes the request and sends the transaction to the participant. This process occurs in the QueryRecord, chaincode and ledger functions. The responses obtained in the QueryRecord function come from the function queries in the chaincode. Process is carried out to retrieve the status ledger (GetState) for each key (args [1]) of requested arguments.

```
func (t *HeroesServiceChaincode) query(stub shim.ChaincodeStubInterface, args []string) pb.Response {
  if len(args) < 2 {
    return shim.Error("The number of arguments is insufficient.")
  }

  // Get the state of the value matching the key hello in the ledger
  state, err := stub.GetState(args[1])
  if err != nil {
    return shim.Error("Failed to get state of hello")
  }

  // Return this value in response
  return shim.Success(state)
}
```

The response obtained in the InvokeRecord function from the website application is obtained from the chaincode call function. The invoke function is performed to pass the state data ledger (PutState) to the key (args [1]) and contains the value (args [2]) of the requested argument.

```
func (t *HeroesServiceChaincode) invoke(stub shim.ChaincodeStubInterface, args []string) pb.Response {
    if len(args) < 2 {
        return shim.Error("The number of arguments is insufficient.")
    }
    if len(args) == 3 {

// Write the new value in the ledger
        err := stub.PutState(args[1], []byte(args[2]))
        if err != nil {
            return shim.Error("Failed to update state of hello")
        }

// Notify listeners that an event "eventInvoke" have been executed
        err = stub.SetEvent("eventInvoke", []byte{ }) if err != nil {
            return shim.Error(err.Error()) }

// Return this value in response
        return shim.Success(nil)
    }

// If the arguments given don't match any function, we return an error
    return shim.Error("Unknown invoke action, check the second argument.")
}
```

III. RESULT AND EVALUATION

This research was conducted using health product data. In general, the process carried out is collecting data, designing the system, creating a product qualification system by applying blockchain technology and evaluating the results. The application of the system used is Hyperledger Fabric for blockchain technology, CouchDb as the blockchain ledger, Postgre sql as the database for the admin system and Go as a programming language.

This chapter discusses the results obtained during the study. The results and this discussion are about the results of making a product qualification system based on a web application and the results of system performance from the tests performed.

A. Product Qualification System based on Web Application

Making a web application-based product qualification system with the implementation of blockchain technology on the Hyperledger Fabric which was carried out in Chapter 3. The product qualification system consists of several features used by participants.

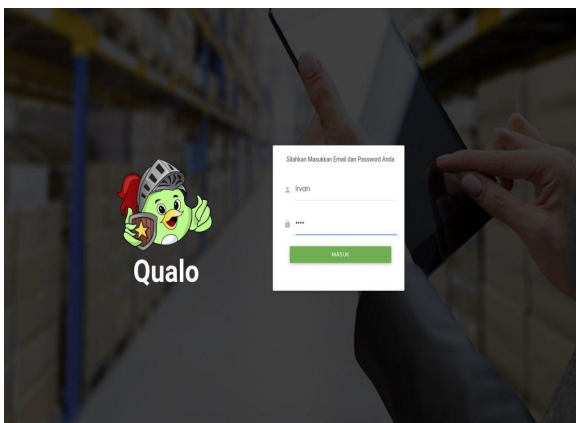


Figure 4: Login

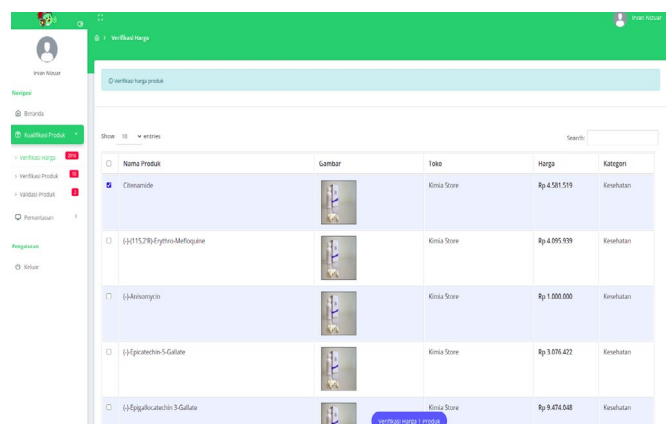


Figure 5: Price Verification

7.	Product Val- idation	Admin clicks on the product val- idation menu	Displays a product validation page that lists products, show data entry and search	Success displays a product validation page that lists products, show data entry and search
8.	Blocks	Admin clicks on the Blocks menu	Displays the blocks page containing the block height, prev block hash, current block hash and details of the block	Success displays the blocks page containing the block height, prev block hash, current block hash and details of the block
9.	Show Entry	Admin can enter the number of product lists that you want to dis- play	Displays a list of products according to the number of show entries	Success Displays a list of products according to the number of show entries
10.	Search	Admin performs product data search input	Displays the product data sought	Success displays the product data sought
11.	Logout	Admin clicks the logout menu	Displays the login page	Success displays the login page
12.	Validation Data	If the data is successfully or- dered and the transaction block is validated	The data will be updated in the ledger and sent to the database system	Success
13.	Validation Data	If the data is not successfully or- dered and the transaction block is not validated	The data will not be updates in the ledger and not sent to the database system	Success

Latency on each page :

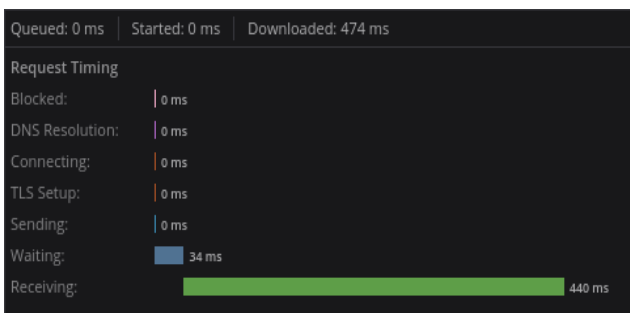


Figure 11: Latency on Price Verification Page

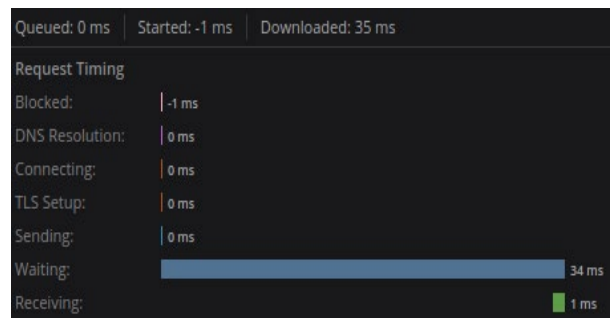


Figure 12: Latency on Product Verification Page

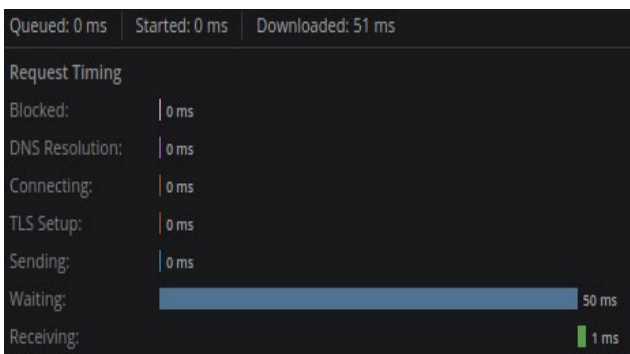


Figure 13: Latency on Product Validation Page

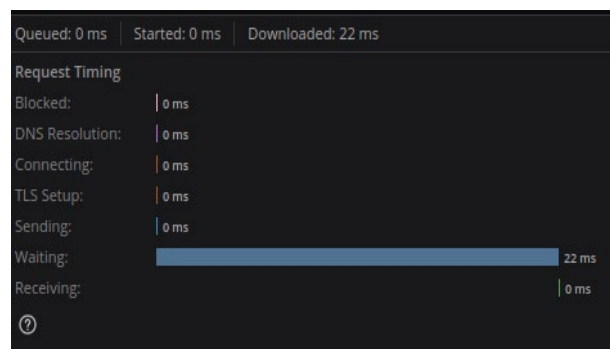


Figure 14: Latency on Block Page

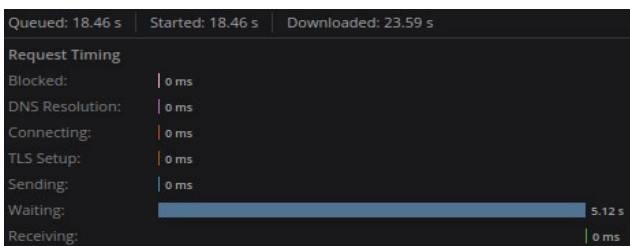


Figure 15: Latency on Verifying Page

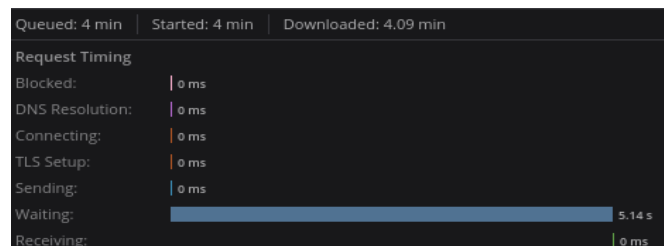


Figure 16: Latency on Validating Page

There are 10 users who use this system. The user is the representative with the participant role in the system. The indicators assessed are the flow of use cases from the system, security in using the system, verification and validation data displayed, monitoring blocks in the system, access time and ease of use. The results obtained are 90% of users find it easy to follow the system in using the application. The activity log contained in the system block can be used to monitor the activity of each role in the system. The resulting latency time is 22 ms that the system can be accessed more quickly.

IV. CONCLUSION

The implementation of the permissioned blockchain on the product qualification system that was made has been successfully carried out. Implementation progress is displayed in the form of a web application based on workflows, needs and consortium types in the product qualification records in the e-marketplace. The web application runs on a decentralized network using chaincode and the data is stored in a distributed ledger. The data stored is in the form of product records that have been qualified through the process of price verification, product verification and product validation which is recorded in the event log and blocks. Each transaction made will generate a transaction id (TxId), chaincode id, event name, payload, block number, sourceURL (peer localhost). Data and transactions are immutable, accountable and transparent so they can be used for tracking audits with a more secure monitoring system.

The tests that have been carried out state that from a functional point of view, the system has successfully carried out every feature and function displayed in the form of a web application in accordance with the workflow and design. The data validation test states that the scenario carried out is in accordance with the output expected by the stakeholders.

The tests that have been carried out state that from a functional point of view, the system has successfully carried out every feature and function displayed in the form of a web application in accordance with the workflow and design. The data validation test states that the scenario carried out is in accordance with the output expected by the stakeholders.

REFERENCES

- [1] Hennesy, J. and Patterson, D. (2017). *Computer Architecture: A Quantitative Approach*. Burlington: Morgan Kaufmann, 4th edition.
- [2] Kurnia, D. (2019). Jaringan 5G Akan Diterapkan 2020 [Online]. Available at: https://www.kominfo.go.id/content/detail/6318/jaringan-5g-akan-diterapkan-2020/0/sorotan_media [Accessed 25 November 2019].
- [3] Apriyadi, D. and Yandi, A.S. (2017). E-Commerce Berbasis Marketplace Dalam Upaya Mempersingkat Distribusi Penjualan Hasil Pertanian. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*. ISSN Media Elektronik : 2580-0760. Vol. 1(2), 131 -136.
- [4] Kotler, P. and Keller, K. (2009). *Manajemen Pemasaran*. Dialihbahasakan oleh Bob Sabran. Penerbit Erlangga Jakarta, Edisi Ketiga Belas.
- [5] Economist (2015). The great chain of being sure about things [Online]. Available at: <https://www.economist.com/briefing/2015/10/31/the-great-chain-of-being-sure-about-things> [Accessed 20 October 2019].
- [6] McDowall, R.D. (2007). Spectroscopy. Validation of Spectrometry Software - Audit Trails for Spectrometer Software, (16). [Online]. Available at: http://files.alfresco.mjh.group/alfresco_images/pharma//2014/08/21/136faec3-ec7a-45ae-b1da-671da9452de8/article-421873.pdf [Accessed 20 October 2019].
- [7] Yosepha, P. (2020). Blockchain Diprediksi Jadi Tren Teknologi Digital 2020 [Online]. Available at: <https://katadata.co.id/ariayudhistira/info-grafik/5e9a498fe0a85/blockchain-diprediksi-jadi-tren-teknologi-digital-2020> [Accessed 20 March 2020].
- [8] Satoshi, N. (2008). Bitcoin: A peer-to-peer electronic cash system [Online]. Available at: <https://bitcoin.org/bitcoin.pdf> [Accessed 20 October 2019].
- [9] Linux Foundation. (2017). Linux foundation hyperledger project [Online]. Available at: <https://www.hyperledger.org/> [Accessed 20 October 2019].
- [10] Hintzman, Z. (2017). Comparing blockchain implementations. Fall Technical Forum SCTE-ISBE and NCTA CABLELABS.
- [11] Crypto Purview. (2019). Media: Understanding Byzantine Fault Tolerance [Online]. Available at: <https://cryptopurview.com/understanding-byzantine-fault-tolerance-bft/> [Accessed 20 October 2019].
- [12] Ching, N. and Chen, Y. (2019). A Reliable E-commerce Business Model Using Blockchain Based Product Grading System. The 4th IEEE International Conference on Big Data Analytics.
- [13] Chang, Y. W. (2019). Blockchain Technology for e-Marketplace. PhD on Pervasive Computing and Communications. Publisher : IEEE International Conference 2019. DOI : 10.1109/PERCOMW.2019.8730733.
- [14] Wang, F. and Hannafin, M. J. (2005). Design-Based Research and Technology-Enhanced Learning Environments. *ETR&D*, Vol. 53, No. 4, 2005, pp. 5-23 ISSN 1042-1629. DOI : 10.1007/BF02504682.
- [15] Wang, W., Azarian, M. and Pecht, M. (2008). Qualification for Product Development. *International Conference on Electronic Packaging Technology & High Density Packaging (ICEPT-HDP 2008)*. 978-1-4244-2740-6. 2008 IEEE.
- [16] IPC, IPC-9701. (2002). Performance Test Methods and Qualification Requirements for Surface Mount Solder Attachments [Online]. Available at: https://global.ihs.com/doc_detail.cfm?document_name=IPC%2D9701&item_s_key=00391814 [Accessed 20 October 2019].
- [17] Imran, B. (2017). *Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks*. Birmingham: Packt Publishing Ltd. ISBN: 1787125440
- [18] Ioannis, K. Marja, P and Nedaa, B. (2018). Design of the Blockchain Smart Contract: A Use Case for Real Estate. *Journal of Information Security*, 9, 177-190. ISSN Online: 2153-1242. DOI: <https://doi.org/10.4236/jis.2018.93013>.
- [19] Zhang, X. (2019). Blockchain Based Secure Package Delivery via Ridesharing. 11th International Conference on Wireless Communications and Signal Processing (WCSP). Publisher: IEEE. DOI: 10.1109/WCSP.2019.8927952.