MAXIMIZING THE USE OF BLOCK PAYLOAD IN BLOCKCHAIN-BASED E-VOTING

[1]ANDI, [2]GEDE PUTRA KUSUMA

[1, 2] Computer Science Department, BINUS Graduate Program - Master of Computer Science,
Bina Nusantara University, Jakarta, Indonesia, 11480

E-mail: [1]andi012@binus.ac.id, [2]inegara@binus.edu

ABSTRACT

Nowadays, internet users are increasing in Indonesia so that it is a good opportunity if traditional paper-based voting is replaced with online voting called e-voting. E-voting has many advantages including environmentally friendly and increased efficiency. However, security issues have always been a barrier to its implementation because e-voting uses a centralized system. Therefore, many researchers propose a combination of e-voting with a decentralized system called blockchain. They believe that voting data becomes more secure because of the immutability of blockchain. However, the process of storing and validating on blockchain is quite slow, so it is not yet feasible to combine it with e-voting. Hence, there are researchers who propose validation from a centralized system such as a fingerprint, but this validation is not secure enough because it can be tampered with. There are also researchers who propose reducing or enlarging the block size on the blockchain to speed up processes on the blockchain. However, increasing the block size slows down the propagation of the block to other nodes. On the other hand, reducing the block size escalates the block composition time to clear all the transactions from the memory pool called mempool. Eventually, e-voting remains inapplicable although e-voting is combined with a modified blockchain in particular blocksize. Thus, we propose optimizing the capacity utilization of a block without changing the capacity of the block itself. Experimental results reveal that the more transactions that can be contained in a block, the faster the data search process, especially in the validation process of uniqueness feature in e-voting and vice versa.

Keywords: *Voting, E-voting System, Blockchain System, Block Payload, Uniqueness Validation*

1. INTRODUCTION

Indonesia is a democratic country so that Indonesian citizens have the freedom to play an active role in elections. Indonesia is also one of the most densely populated and developing countries. The number of internet users in this country used to be very low but nowadays it has increased along with the development of technology from 3G to 5G. Besides, since the first quarter of 2020, the COVID-19 pandemic has spurred its increase. At that time, students from elementary students to college students were required to study online. The workers are also required to work from home using the internet. Responses to the pandemic also have accelerated the adoption of digital technologies. This technological progress and the rapid transformation of human habits can influence and facilitate human activities in the political field, especially elections. Therefore, it is certainly a good opportunity if the future election in Indonesia is also conducted online using an internet called e-voting. The e-voting provides enhanced features of the election system compared to the conventional election. E-voting can create environmentally friendly because the use of paper can be reduced to as little as possible [15]. Moreover, it provides convenience to vote without geographical restrictions. The vote count is also faster because the ballots are inputted directly into the system and counted automatically by the system [11]. However, the issue of security is always the most frequent concern for the e-voting system because it uses a centralized system with one organization that has complete control over the database and the system. So eventually, it is possible to tamper with the database of significant opportunities [18].

Blockchain is one of the solutions for the security issue of e-voting systems. The entire database is owned by a peer-to-peer network [18] and maintained by a system of consensus [7]. Moreover, it is distributed and immutable [6]. Blockchain structure consists of several blocks that are linked to each other and in a block can have multiple transactions [19]. The attempt to change the information in a block will be more troublesome as it must change the next blocks. In addition, because of its immutable characteristic, the blockchain system does not tolerate any changes. Furthermore, every transaction can be traced back to its origin, likewise every transaction of e-voting can be traced without disclosing the voter's identity [20].

Despite those advantages, blockchain is slow and requires a lot of energy to perform validation. On the other hand, people are used to being pampered with easy access to their needs quickly due to the adoption of digital technology. As a result, using it for e-voting may not yet be feasible [3]. As we know, the following features should be included in a practical secure e-voting:

- ✔ Confidentiality: no one can know who the voter voted for when voting took place.
- ✔ Eligibility: voting is restricted to registered and authorized voters [09, 14].
- ✔ Uniqueness: every voter can only vote once [14].
- ✔ Integrity: the votes cannot be changed [16].
- ✔ Verifiability: the votes can be counted correctly [17].

In addition, there are three phases of general election such as registration phase, voting phase, and tallying result phase. Based on the features above, the unique feature requires validation, especially in the voting phase. Our proposed model will be focused on validation. Due to the disadvantages of blockchain in terms of time consuming, many researchers have proposed blocksize optimization, including increasing or decreasing the size of a block with the aim of obtaining the most appropriate block size so that the blockchain process becomes faster. However, this goal has not been achieved and instead created new problems. Therefore, we propose a maximum usage model of a block in the blockchain to achieve efficient validation for the voting phase, especially in uniqueness validation.

The rest of this paper is organized as follows: Section 2 presents previous work related to this research in the context of e-voting systems based on blockchain. Section 3 presents an overview of the proposed e-voting system followed by details of the experiment in section 4. Section 5 concludes this paper.

2. RELATED WORKS

Numerous protocols have been proposed to enable voting to be conducted in a manner that takes advantage of the distributed and immutable structure of blockchains. Moreover, many researchers have recognized that blockchain can bring to the field of voting such as election [6]. During the election, general election commissions may encounter a variety of issues. Duplication or double voting is the most prevalent issue. To guarantee that an eligible voter can only vote one-time, secure authentication is crucial [9]. So there are many studies that use various kinds of technology to overcome this issue.

Khan, Arshad, & Khan proposed blockchain-based e-voting with an open-source platform called multichain [2]. This system implements an authentication mechanism using fingerprint technology so that only valid voters can access the system. The use of biometrics also allows the system to protect against multiple voting. After the voter has successfully cast a vote, the user is sent an email containing a unique transaction id in the form of a cryptographic hash. This transaction id can be used by voters to track if their vote was included in the counting process or not. From the results of the evaluation, it is found that one voter cannot vote more than once. However, this authentication mechanism is still using a centralized system, not a direct checking on the data recorded in the blockchain. So, it is still possible to vote more than one time if the data on the centralized system has been changed.

Varalakshmi, Malarvizhi, Shamitha, Srimathi, & Vinisha proposed e-voting based on the Ethereum blockchain [4]. The voting event takes place via a web application, where voters are allowed to cast their vote from anywhere. The server will authenticate each user with an Aadhar number - a unique 12-digit number issued by the Unique Identification Authority of India (UIDAI) taking into account a person's biometric details such as iris and fingerprint scans, as well as demographic information such as date of birth and address. In order for the user to enter into the application, a suitable OTP will be generated for the registered mobile number. Users must enter the Aadhar number and OTP. From this research, it was found that the confidentiality of voting data is maintained by encrypting the data and storing it on the blockchain as blocks and duplicating votes during the election process can be avoided because the coin or token can only be used one time only. However, if the coin is filled again, it will be an opportunity for voters to vote again.

Kazi Sadia proposed a decentralized e-voting system using blockchain technology. This protocol utilizes smart contracts into e-voting to deal with security, accuracy, and privacy issues of voters during voting so that voting transactions cannot be edited and verified independently [1]. This protocol consists of 3 interdependent phases, namely: the pre-voting phase, the voting phase, and the post-voting phase. In the pre-voting phase, the voter list must contain the voter's name, national identification number, fingerprints and other information based on the direction of the election commission. Alternatives to fingerprints like pins are also prepared. In the voting phase, voters provide their public key to be verified. Then the private key (fingerprint that is converted into binary data) is also given to match the fingerprint. After the voter casts a vote, the vote is added to the blockchain while the connected blocks remain not broadcast. In the post-voting phase, all connected blocks will be broadcast one by one consecutively so that each voter can count the results of the vote and does not require a third party. This protocol is believed to reduce the constraints of manual voting and other electronic voting systems based on blockchain that use the least amount of third-party involvement. The techniques used in this protocol are quite simple and easy to understand and are designed to reduce memory and time consumption and thus be faster. Voters can monitor the entire process and their privacy is protected. There is already a check that the voter has cast his vote or not in the blockchain data. However, every single transaction is instantly included as a block in the blockchain. Eventually, there will be many blocks in that blockchain.

Gottfried Christophorus Prasetyadi proposed a voting protocol, a redesigned block structure utilizing the SHA3-256 hash technique, and a ballot design as a block transaction using UUID version 4 [5]. A ballot must be at least 43 bytes long. In this proposed voting system, each voter has to input a public key and there is no need for a fingerprint for verification. The public key associated with a signed ballot is marked or erased in the proposed system when it has been determined that it is valid. Eventually, the voter could only vote once. However, using the stored public key deletion method is similar to using a centralized system. Moreover, the maximum size of each block is 1,000 bytes and each block has only 20 transactions. Unfortunately, there is no evaluation regarding the limit of 20 transactions per block.

Yousif Mohammed Wahab, Alaan Ghazi, Arns Al-Dawoodi, Muthana Alisawi, Sirwan Saber Abdullah, Layth Harmnood, Asmaa Yaseen Nawaf proposed a secured blockchain based framework for e-voting in Iraq [8]. Voter authentication is based on a unique key shared after voter registration. Each transaction of the voter is stored in smart contracts to prevent duplicate voting. Moreover, each block can only contain one transaction. So, there will be many blocks formed because the number of blocks is directly proportional to the number of voters. Each block has a header, and its size is 80 bytes [25]. Eventually, there will be more storage consumption.

Kashif Mehboob Khan, Junaid Arshad, and Muhammad Mubashir Khan talked about how well the secure e-voting system based on blockchain worked. They asserted that the scalability and overall performance of blockchain-based solutions are significantly influenced by block size. The number of transactions in a block, block generation rate, and block size have significant effects on blockchain-based systems [22]. In addition, a larger block size raises the transaction fee but lowers the likelihood of successfully mining blocks [23].

Blocks having a high size reduce block generation time while also improving transaction costs and overall performance. Since nodes with lesser bandwidth cannot compete with those with higher computational sources, this could lead to less decentralization. This lack of competition reduces the security of blockchains [10]. Additionally, once a block has been formed in a PoW network, it is distributed throughout the node network. Therefore, transferring blocks takes longer as block sizes increase. Additionally, it weakens security and boosts centralization [12]. On the other hand, a smaller block is more efficient, but it will require higher block composition time to clear all the transactions from the mempool [26].

According to Nicola Dimitri [13], optimizing the block size resolves the conflict between maximizing revenue and becoming an alternative payment system. In addition, when building an encryption using a cryptographic hash function, choosing a block size compromises security for speed. Security becomes a significant concern if the block size shrinks. On the other hand, speed of hashing and transmission becomes an issue if the block size is larger. From the perspective of miners, utilizing a large block size raises transaction fees. As previously stated, both large and small block sizes bring about several issues.

When paper-based voting is implemented, all voters can easily be convinced that those who have voted cannot vote again because after they have voted, one of their fingers is dipped in ink which is very difficult to remove within 1 day while the time to vote is usually only carried out in 1 day. However, when voting takes place in an electronic environment, most people can accept and use e-voting, but people have doubts about the safety and accuracy of e-voting. They cannot easily trust the e-voting system unless the e-voting system is validated. If the validation process is applied to the e-voting system, then the level of trust will increase and more voter participation can be easily achieved. From some of the literature above, e-voting has been combined with blockchain and is equipped with the validation. However, their model creates speed problems related to block size which can be summarized in Table 1.

Table 1: Summary of Literature Review in Previous Research.

Research	Method	Evaluation Result
[2]	Voter validation uses a centralized system such as the use of biometrics.	It is still possible to vote more than once if the data in the centralized system has been changed.
[4]	Voter validation by limiting the use of coins (only once)	If the coin is topped up, it will be an opportunity for voters to vote again.
[1]	Each vote is stored as one block.	There will be many blocks on the blockchain.
[5]	Validation of voter uniqueness by deleting the list of public keys whose votes are declared valid and sent to the blockchain.	The data deletion method is the same as a centralized system because there is no data deletion system on the blockchain.
[8]	Using smart contracts to prevent double voting. Every 1 vote equals 1 block.	There will be many blocks formed because each transaction equals one block.
[22]	Both blockchain scalability and performance are affected by block size.	Increasing block size can increase transaction costs but decrease the success of block mining.
[10]	Evaluate the influence of the size of the block size	Large blocks reduce the security of the blockchain, and it takes a long time to transfer blocks.
[26]	Evaluate smaller block sizes	Smaller blocks are more efficient but require higher block composition time to remove all transactions from the mempool.
[12]	Evaluate large block sizes	Blocks with large size can weaken security and increase system centralization.
[13]	Evaluate the influence of the size of a block	Both reducing or enlarging the block size both cause several problems.

Therefore, our proposed model is no longer based on block size parameters for blockchain-based e-voting systems. Our proposed model will be more about the number of transactions that can be loaded until it reaches the maximum payload or capacity of a block. We will conduct experiments to prove that a block whose capacity is maximally used will be faster (fully filled without changing the block size).

3. PROPOSED SYSTEM

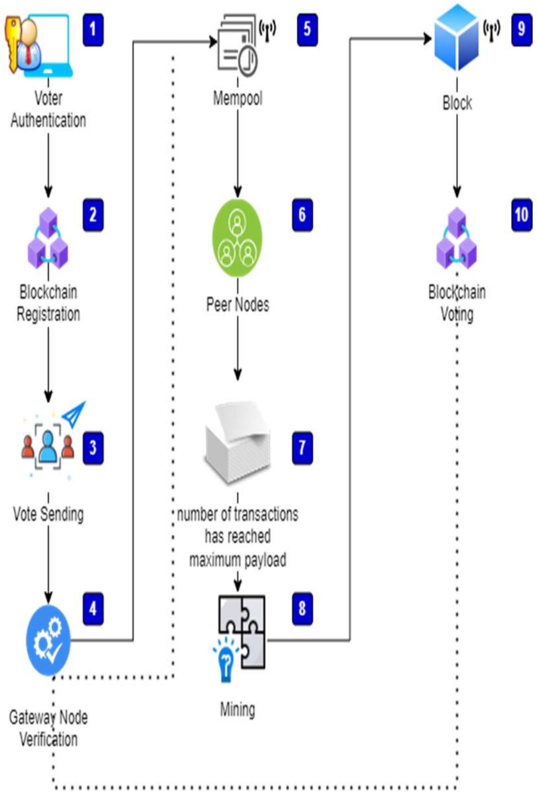


Figure 1: Proposed Model for Voting Phase

Our proposed model has shown in Figure 1. The voting phase begins on voter authentication. In this step, the voter submits the voter's public key address. This address will be checked based on data stored on blockchain registration. If the public key address is valid, the voter can start to choose one of the candidates. Each candidate is also represented by the candidate's public key address. After choosing one of them, the vote is sent to a node, often called a gateway node. That node will verify whether the voter's public address already existed or not because each voter is allowed to vote once. If the verification is valid, then the vote is added to the mempool. The mempool is the gateway to the blockchain. Before the vote can be written on a block, it must first move through the mempool. Since the node is connected to a group of peers, it broadcasts the transaction to other nodes called peer nodes. These peer nodes will receive the vote, validate it, move it into their own mempool, and broadcast to additional peers, essentially replicating the vote across the network. Miners in Proof of Work Consensus or Stakers in

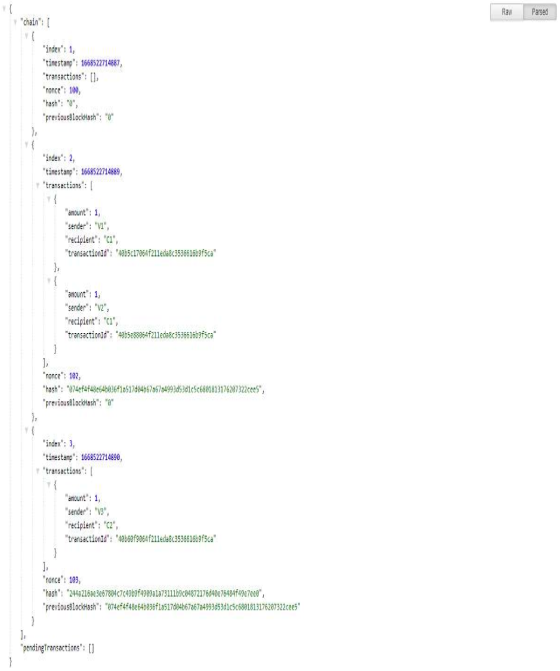
Proof of Stake Consensus, as a specific kind of node, also receive the transaction from peers, validate it, and attempt to add it as a new block if the number of transactions in the mempool has reached the maximum payload on a block. We assume the size of a block is 1 MB following the standard block size in bitcoin [21]. We also assume every vote or transaction has an average size 250 bytes following the average size of bitcoin transaction [21]. So, the maximum payload for a block is approximately 4,000 transactions. Eventually, a successful miner or a chosen staker adds a block with the collection of votes. The new block is broadcast over the network to existing chain namely blockchain voting.

In terms of speed, our model is actually based on the basic concept where a set of data stored in a database will be faster in searching if it is combined into a table at once compared to being split into several tables. In terms of savings, this model can also be likened to a group of goods that will be sent to the same place, combining them into a container (as long as it meets its capacity) will be more economical than breaking it into several containers.

4. EKSPERIMENTS

4.1 Experimental Design

To conduct the experiment, we represent blockchain voting data in JSON format which consists of two main arrays, namely chain and pendingTransactions. The pendingTransactions array represents the mempool of transactions that are valid but not yet confirmed to be block while the chain array represents the voting blockchain which consists of many objects. Each of these objects consists of an index, a timestamp, a transaction in the array, a nonce, a hash, and the previous hash. Each object represents a block in the blockchain, and each transaction array represents the transactions stored in that block. In the first index of block, the transaction is always empty, the hash is always zero, and the previous hash is always zero which indicates that there are no transactions in this first index block and there are no previous blocks because this block is the very first block formed as a genesis block. In the second index of block, the transaction array consists of two transactions with different senders, namely V1 and V2 and the same recipient, namely C1, which illustrates that the block consists of different or unique voter public key addresses but chooses the same candidate. The blockchain structure is an illustration of the blockchain structure with the block type of two transactions per block (TPB). From this example, the number of voters is three, so the first two transactions are in the second index of block and the rest is in the third index of block. To make it easier to understand, the JSON format data structure of the blockchain is illustrated in Figure 2.



```
{
  "chain": [
    {
      "index": 1,
      "timestamp": 1668522714887,
      "transactions": [],
      "nonce": 100,
      "hash": "0",
      "previousBlockHash": "0"
    },
    {
      "index": 2,
      "timestamp": 1668522714889,
      "transactions": [
        {
          "amount": 1,
          "sender": "V1",
          "recipient": "C1",
          "transactionId": "40b5c17064f211eda8c3536616b9f5ca"
        },
        {
          "amount": 1,
          "sender": "V2",
          "recipient": "C1",
          "transactionId": "40b5e88064f211eda8c3536616b9f5ca"
        }
      ],
      "nonce": 102,
      "hash": "074ef4f48e64b036f1a517d04b67a67a4993d53d1c5c6801813176207322cee5",
      "previousBlockHash": "0"
    },
    {
      "index": 3,
      "timestamp": 1668522714890,
      "transactions": [
        {
          "amount": 1,
          "sender": "V3",
          "recipient": "C2",
          "transactionId": "40b60f9064f211eda8c3536616b9f5ca"
        }
      ],
      "nonce": 103,
      "hash": "244a216ae3e67804c7c49b9f4909a1a73111b9c04872176d40e76484f49e7ee0",
      "previousBlockHash": "074ef4f48e64b036f1a517d04b67a67a4993d53d1c5c6801813176207322cee5"
    }
  ],
  "pendingTransactions": []
}
```

Figure 2: Blockchain Structure in JSON Format

In short, the illustration of the blockchain structure is as follows:

- Number of transactions = 3, block type 2 transactions per block (TPB)
- Number of sender/voter/voter addresses = 3
- Number of recipients/selected/candidate addresses = 2
- List of senders that have been encrypted and hashed = {V1, V2, V3}
- List of recipients that have been encrypted and hashed = {C1, C2}
- First index => genesis block (no transactions, hash = 0, previous hash = 0)
- Second index => block with number of transactions = 2, sender = {V1, V2}, recipient = {C1, C1}. The sender must be different/unique but the recipient can be the same.
- Third index => block with number of transactions = 1, sender = {V3}, recipient = {C3}

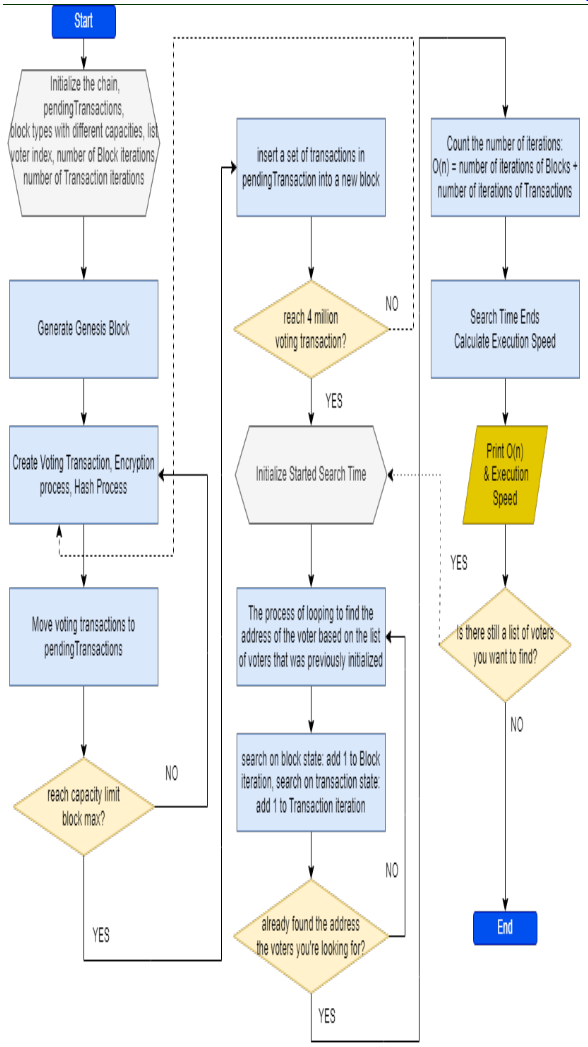

Start
Initialize the chain, pendingTransactions, block types with different capacities, list voter index, number of Block iterations, number of Transaction iterations
Generate Genesis Block
Create Voting Transaction, Encryption process, Hash Process
Move voting transactions to pendingTransactions
reach capacity limit block max?
NO
YES
insert a set of transactions in pendingTransaction into a new block
reach 4 million voting transaction?
NO
YES
Initialize Started Search Time
The process of looping to find the address of the voter based on the list of voters that was previously initialized
search on block state: add 1 to Block iteration, search on transaction state: add 1 to Transaction iteration
already found the address the voters you're looking for?
NO
YES
Count the number of iterations: O(n) = number of iterations of Blocks + number of iterations of Transactions
Search Time Ends Calculate Execution Speed
Print O(n) & Execution Speed
YES
Is there still a list of voters you want to find?
NO
End

Figure 3: Experimental Design Flowchart

Evaluation of the experimental results is based on a comparison of the complexity of the time needed to find the voter's public key address (V1, V4, V40, V400, V4000, V40000, V400000, and V4000000) based on block type variables (1TPB, 4TPB, 40TPB, 400TPB, and 4000TPB). The smaller the time complexity, the better the model and vice versa. Similarly, the smaller the execution speed in seconds, the better the model is.

In order to generate JSON format data from the blockchain as well as to measure performance, we initialize the number of voters equal to 4,000,000. The variable number of transactions in one block varies from 1, 4, 40, 400, and 4,000. The voter's public key address variable also varies from V1, V4, V40, V400, V4000, V40000, V400000, and V4000000. The pseudocode for searching the voter's public key address is as follows:

```
Function isExist(chain, searchAddress)
    timeComplexityBlock = 0
    timeComplexityTransaction = 0
    isExist = False

    for each block in chain do
        timeComplexityBlock = timeComplexityBlock + 1
        for each transaction in block do
            timeComplexityTransaction = timeComplexityTransaction + 1
            if transaction.sender = searchAddress then
                isExist = True
                timeComplexityTotal = timeComplexityBlock + timeComplexityTransaction
                return isExist
            end if
        end for of transaction
    end for of block
    return isExist
End Function
```

To measure the performance of whether the line of code created is fast or efficient enough, we need a methodology to calculate it. One tool that can be used is Big-O Notation. It is known as time complexity. It relates to how long lines of code are executed. Why is there a need for a method to calculate code efficiency? Because we cannot just say that this set of code can run for one, two, or three seconds. Even though there are so many other determining factors such as the amount of data, connection, latency, amount of memory, processor speed and many others. Therefore, we need a measuring tool to calculate the relative efficiency of the code like this big O notation. Big-O notation is a way or method for analyzing a programming algorithm against execution time.

Time complexity is measured using big O notation, namely O(n) by calculating in detail based on iterations on the block index and on the transaction index. Then the speed of execution is measured by reducing the time after execution with the time before execution. The execution speed measurement algorithm is as follows:

```
let startTime = performance.now();
let objAddress = await findVoter(this.chain, address);
let endTime = performance.now();
timeFind = (endTime - startTime)/1000;
tFind = Number(timeFind).toFixed(6);
```

4.2 Eksperimental Results

The experimental results in Table 2 are the results of the voter block index stored in blockchain-based voting data. For ease of understanding, the actual voter addresses are in the form of hashed public keys denoted by V1, V4, V40, V400, V4000, and V40000. Blocks with 1 transaction per block are denoted by 1TPB. Blocks with 4 transactions per block are denoted by 4TPB. Blocks with a total of 40 transactions per block are denoted by 40TPB and so on.

It is assumed that every Vn votes in the n-queue. Then V1 votes in the first order so that the block index on V1 always starts at the second index for each type of block because the first block is always filled with blocks without transactions which is also known as the genesis block. Then the block at V40 starts with the block at index 41 for block type 1TPB, calculated from 40 divided by 1 then added by 1. As for the 4TPB block type, the address of the 40th voter is at index 11, calculated from 40 divided by 4 then added by 1. Then for the 40TPB block type, V40 is at the second index, calculated from 40 divided by 40 then added by 1 and so on.

If you pay attention, it can be concluded that the more transactions that are loaded into a block, the smaller the voter's address block index will be, which means that the number of blocks is also getting smaller in the voting blockhain.

Table 2: Block Index of Voter Address Based on TPB

	Block Index (block 1 is always a genesis block)				
Voter Index	1TPB	4TPB	40TPB	400TPB	4000TPB
V1	2	2	2	2	2
V4	5	2	2	2	2
V40	41	11	2	2	2
V400	401	101	11	2	2
V4000	4,001	1,001	101	11	2
V40000	40,001	10,001	1,001	101	11
V400000	400,001	100,001	10,001	1,001	101
V4000000	4,000,001	1,000,001	100,001	10,001	1,001

Table 3: Time Complexity Based on TPB

	Time Complexity = O(Block Index) + O(Voter Address Index)				
Voter Address	1TPB	4TPB	40TPB	400TPB	4000TPB
V1	3	3	3	3	3
V4	9	6	6	6	6
V40	81	51	42	42	42
V400	801	501	411	402	402
V4000	8,001	5,001	4,101	4,011	4,002
V40000	80,001	50,001	41,001	40,101	40,011
V400000	800,001	500,001	410,001	401,001	400,101
V4000000	8,000,001	5,000,001	4,100,001	4,010,001	4,001,001

The experimental results in Table 3 are the total time complexity for searching voter addresses based on block types: 1TPB, 4TPB, 40TPB, 400TPB, and 4000TPB. The total time complexity for V40 is 81 for block type 1TPB, calculated from block index V40 with block type 1TPB stored in Table 2, which is 41 plus data transaction index V40 on block index 41, which is 40. The total time complexity for V40 is 51 for type block 4TPB, calculated from block index V40 with block type 4TPB stored in Table 2, which is 11 plus data transaction index V40 on block index 11, which is 40 and so on.

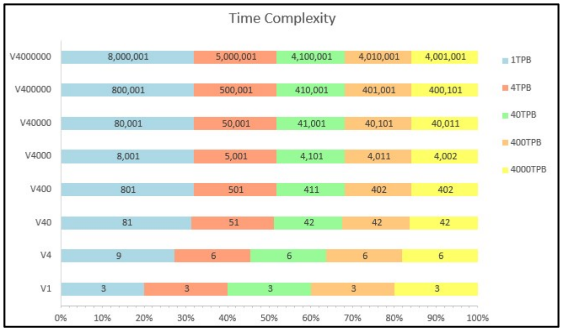


Figure 4: Time Complexity Based on TPB

From Table 3 and Figure 4, it can be concluded that the more transactions are loaded into a block, the smaller the time complexity for searching unsorted data. In other words, the number of iterative processes needed to validate the principle of uniqueness in voting will be more efficient on a blockchain with a smaller number of blocks even though each block has a larger number of transactions compared to a blockchain with a larger number of blocks with each block only consisting of multiple transactions or even just one transaction.

The experimental results in Table 4 are the total time needed to search for voters' addresses based on block types: 1TPB, 4TPB, 40TPB, 400TPB, and 4000TPB. When looking for voters on the largest index, namely V4000000, in blocks 1TPB, 40TPB, and 4000TPB it takes approximately 0.229529 seconds, 0.145646 seconds, and 0.114687 seconds. Thus, we can also conclude that the more transactions are loaded into a block, the faster the execution of data searches tends to be so that the process of validating the principle of uniqueness in voting becomes more efficient in terms of time.

Table 4: Search Execution Speed Based on TPB

	Run Time (second)				
Voter Index	1TPB	4TPB	40TPB	400TPB	4000TPB
V1	0.247318	0.136783	0.165222	0.125977	0.119374
V4	0.260505	0.134037	0.130181	0.115353	0.090490
V40	0.236335	0.134844	0.127632	0.117023	0.089877
V400	0.236722	0.140797	0.133247	0.116572	0.090915
V4000	0.240285	0.136569	0.125083	0.125537	0.092208
V40000	0.239558	0.140675	0.138008	0.146473	0.094583
V400000	0.231826	0.149767	0.135966	0.129760	0.103421
V4000000	0.229529	0.155779	0.145646	0.138072	0.114687

Thus, to create an efficient validation process for e-voting combined with blockchain, it is not necessary to change the blocksize parameter as in previous studies, but what is important is to maximize the capacity of the block itself.

5. CONCLUSIONS AND FUTURE WORKS

So, is e-voting currently applicable when combined with blockchain technology? From the experimental results in this study, it was found that the fewer transactions contained in a block, the more blocks that will be formed. The more blocks that are formed, the longer the search process for unsorted data becomes. Conversely, the more transactions that are contained in a block, the fewer blocks that are formed so that the search process for a data becomes faster.

So, from the results of this research, the following conclusions can be drawn:

- In order to implement an efficient validation process so that e-voting can be combined with the blockchain system, each block on the blockchain voting needs to maximize its load capacity so that voting transactions are not divided into many blocks. This is the same as the basic concept where a set of data stored in a table will be found faster during the search process compared to a set of data that is splitted stored into several tables.
- So that a block on the blockchain does not need to be reduced or increased in its size, the best solution is also to use the maximum capacity of each block in the blockchain. In addition, to speeding up the search for unsorted data, maximizing the load capacity of each block eventually saves the storage capacity needed by the blockchain because each block that is formed requires 80 bytes in the header. Similarly if a number of goods are loaded into the same container as long as it fits its capacity, it will be more saving than if the goods are loaded separately into several containers.

In the e-voting system, there are three important stages, namely, the voter registration stage, the voting stage, and the results calculation stage. Due to time constraints, we only focused on one of the previous research stages, namely the voting stage. However, the voter registration stage is no less important to study, even the results calculation stage is very important to examine the level of efficiency when e-voting is combined with blockchain.

We also suggest that future research on this blockchain-based e-voting system is not limited to research on these two well-known consensuses namely Proof of Work (POW) and Proof of Stake (POS) but also extends to other consensuses such as Practical Byzantine Fault Tolerance (PBFT).

REFERENCES

[1] K. Sadia, Md. Masuduzzaman, R. K. Paul, and A. Islam, Blockchain Based Secured E-voting by Using the Assistance of Smart Contract, 2019, pp 1-15.

[2] K. M. Khan, J. Arshad, and M. M. Khan, Secure Digital Voting System based on Blockchain Technology, 2018.

[3] N. Kshetri, and J. Voas, Blockchain-Enabled E-Voting, 2018.

[4] Ms.V. Varalakshmi, S. Malarvizhi, A. Shamitha, S. Srimathi, and V. Vinisha, Blockvote: Aadhar Based Electronic Voting

System Using Blockchain, 2020.

[5] G. C. Prasetyadi, A. B. Mutiara, and R. Refianti, Blockchain-based Electronic Voting System with Special Ballot and Block Structures that Complies with Indonesian Principle of Voting, 2020.

[6] B. D. Anderson, Improving the Trustworthiness of Electronic Voting Systems Using Blockchain, 2020.

[7] Y. Wu, An E-voting System based on Blockchain and Ring Signature, 2017.

[8] A. Ghazi, L. Hammood, A. Al-dawoodi, A Framework for Blockchain Based E-Voting System for Iraq, 2022.

[9] R. Tas, and O. O. Tanriover, A Manipulation Prevention Model for Blockchain-Based E-Voting Systems, 2021.

[10] I G. A. K. Gemeliarana, and R. F. Sari, Evaluation of Proof of Work (POW) Blockchains Security Network on Selfish Mining, 2018.

[11] E. E. Chotima, and A. Pramanti, E-Voting Systems to Prevent Conflicts caused by False Results in Elections in Indonesia, 2020.

[12] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends, 2017.

[13] N. Dimitri, Transaction Fees, Block Size Limit, and Auctions in Bitcoin, 2019.

[14] F. S. Hardwick, A. Gioulis, R. N. Akram, and K. Markantonakis, E-Voting with Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy, 2018.

[15] G. Kumar, S. Gupta, D. Agarwal, and A. Tiwari, Virtual Voting System, 2021.

[16] L. C. Bollinger and M. A. McRobbie, “Ensuring the integrity of elections” in Securing the Vote: Protecting American Democracy, pp. 103–105, 2018.

[17] G. S. Grewal, M. D. Ryan, L. Chen, M. R. Clarkson, Du-Vote: Remote Electronic Voting with Untrusted Computers, 2015.

[18] R. Hanifatunnisa, and B. Rahardjo, Blockchain Based E-Voting Recording System Design, 2017.

[19] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, J. Kishigami, Blockchain Contract: A Complete Consensus using Blockchain, 2015.

[20] Y. Abuidris, R. Kumar, T. Yang, and J. Onginjo, Secure Large-scale E-voting System Based on Blockchain Contract Using a Hybrid Consensus Model Combined with Sharding, 2020.

[21] J. Gobel, and A.E. Krzesinski, Increased block size and Bitcoin blockchain dynamics, 2017.

[22] K. M. Khan, J. Arshad, and M. M. Khan, Investigating Performance Constraints for Blockchain Based Secure e-Voting System, 2019.

[23] S Jiang, J Wu, Bitcoin mining with transaction fees: a game on the block size., 2019.

[24] I. Chivers, and J. Sleightholme, An Introduction to Algorithms and the Big O Notation, 2015.

[25] The Blockchain, https://www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch07.html.

[26] N. Singha, and Manu Vardhan, Computing Optimal Block Size for Blockchain based Applications with Contradictory Objectives, 2020.