TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

Design and Evaluation of a Decentralized E-Voting System Using Ethereum Smart Contracts

Ludgerdus Pati Hurit*, Yo'el Pieter Sumihar, Haeni Budiati

Faculty of Computer Science, Informatics Engineering Study Program, Universitas Kristen Immanuel, Yogyakarta, Indonesia
Email: [1,*]Ludgerdusl@gmail.com, [2]pieter.haro@ukrimuniversity.ac.id, [3]heni@ukrimuniversity.ac.id
Correspondence Author Email: Ludgerdusl@gmail.com

Abstract–The widespread implementation of electronic voting systems poses ongoing challenges related to data integrity, transparency, and centralized control, which can increase the risk of vote manipulation and reduce traceability. To address these issues, this study designs and evaluates a decentralized electronic voting system implemented using Ethereum smart contracts. The objective of this research is to test the ability of blockchain technology to support a secure, transparent, and tamper-resistant voting process in a decentralized environment. The research methodology includes requirements analysis, system design, system implementation, and functional testing. Black-box testing was conducted to verify the system's functionality throughout the voting process. The proposed system permanently records voting transactions on the blockchain, preventing unauthorized modifications while allowing transaction verification by network participants. Voter privacy is maintained by separating voter identity data from voting records and implementing blockchain address abstraction, ensuring that individual votes cannot be directly linked to voter identities. System evaluation focuses on transaction costs and confirmation times. Performance testing was conducted using six test transactions on the Sepolia blockchain network. The total transaction cost recorded was 0.006076 ETH, with an average cost of 0.001013 ETH per transaction. The minimum transaction cost of 0.000091 ETH occurred during voting operations, while the maximum cost of 0.005596 ETH was associated with smart contract deployment and higher network base fees. The average transaction confirmation time was approximately 12 seconds. Although the evaluation was based on a limited number of transactions, the results indicate that the proposed system demonstrates reliable transaction execution, acceptable gas usage, and high transparency. Further large-scale testing is recommended for future work.

Keywords: Blockchain; E-voting; Ethereum; Next.js; Smart Contract

1. INTRODUCTION

The rapid advancement of information technology over the past decade has significantly transformed how digital activities are conducted, including decision-making processes that were traditionally performed manually. One application that has gained considerable attention in this context is electronic voting (e-voting). E-voting systems aim to improve efficiency, transparency, and accuracy by digitizing the voting process, thereby reducing administrative costs, minimizing human errors, and accelerating vote counting and result dissemination (A. Singh et al., 2023). As democratic processes increasingly rely on digital platforms, the demand for secure and trustworthy electronic voting mechanisms continues to grow, particularly in environments that require high levels of integrity and public confidence.

Despite these advantages, conventional e-voting systems face substantial technical and organizational challenges. Most existing implementations rely on centralized architectures, where voting data is stored and processed by a single authority or server. Such architectures are vulnerable to various security threats, including data manipulation, cyberattacks, unauthorized access, and single points of failure (Khan et al., 2021). Moreover, centralized databases often lack publicly verifiable audit trails, making it difficult for stakeholders to independently verify election integrity. This limitation reduces public trust in election outcomes and raises concerns regarding transparency and accountability in digital voting systems (Ohize et al., 2024). Dependence on a central authority also increases the risk of abuse of power, insider threats, and undetected alterations of voting results, which are critical issues in democratic decision-making processes.

To address these challenges, blockchain technology has emerged as a promising alternative for building secure and transparent e-voting systems. Blockchain is a distributed ledger technology that enables data to be recorded across multiple nodes in a decentralized manner. Its core characteristics, decentralization, transparency, immutability, and traceability, make blockchain particularly suitable for applications that require high levels of integrity and trust (I. Singh et al., 2024). Once data is recorded on a blockchain, it cannot be altered without consensus from the network, ensuring that voting records remain tamper-resistant throughout the election process and accessible for public verification.

However, the application of blockchain in voting systems introduces a critical tension between transparency and privacy. While blockchain's inherent transparency enables independent verification of election integrity, it simultaneously poses risks to voter anonymity if not carefully managed. In traditional blockchain implementations, all transactions are publicly visible on the distributed ledger, which could potentially compromise the confidentiality of individual votes. This presents a fundamental challenge in designing blockchain-based voting systems, as democratic elections require both verifiable integrity and guaranteed voter secrecy. Several approaches have been proposed to reconcile this tension, including cryptographic techniques such as homomorphic encryption, zero-knowledge proofs, and ring signatures that enable vote verification without revealing voter identity or vote content (Shahzad & Crowcroft, 2019). Alternative methods involve using commitment schemes where voters submit encrypted vote hashes to the blockchain while maintaining vote confidentiality until a predetermined tallying phase. The balance between ledger transparency and voter privacy remains an active area of research, and any practical implementation must carefully

This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

consider which information is stored on-chain, how voter identities are protected, and how verifiability is maintained without sacrificing anonymity.

In addition to privacy considerations, blockchain platforms such as Ethereum support the execution of smart contracts. Smart contracts are self-executing programs that automatically enforce predefined rules when specific conditions are met. In the context of e-voting, smart contracts can automate critical processes such as voter registration, vote casting, vote validation, and vote tallying without requiring human intervention (B. Wang et al., 2024). This automation significantly reduces the risk of fraud, manipulation, and procedural inconsistencies while ensuring that voting rules are applied uniformly. Consequently, blockchain-based e-voting systems have attracted growing attention from researchers and practitioners as a potential solution to the limitations of conventional e-voting approaches.

Numerous studies have investigated the integration of blockchain technology into e-voting systems. Jain et al. (2023) proposed an efficient blockchain-based voting scheme that emphasizes security and versatility; however, their approach requires further optimization of transaction costs to support large-scale elections. (*Ethereum Gas and Fees*, 2025)developed an Ethereum-based e-voting system using smart contracts to reduce vote manipulation, yet their study did not evaluate transaction performance on public blockchain test networks. (Kim & Kim, 2024) presented a comprehensive survey on blockchain-based voting systems, highlighting both opportunities and challenges, particularly regarding scalability, privacy, and system performance. Their work identified that while blockchain offers significant security advantages, questions regarding throughput capacity and cost efficiency in large-scale deployments remain inadequately addressed in existing literature.

Other researchers have focused on enhancing specific aspects of blockchain-based voting systems. (Alvi et al., 2022) examined the role of consensus mechanisms in improving transaction throughput, although their proposed system was not tested under real-world voting conditions. Jayakumari et al. (2024) explored blockchain as a secure storage medium for election results but did not fully optimize integration with modern web-based user interfaces. (X. Wang et al., 2024) presented a blockchain-based voting case study using an alternative blockchain platform, demonstrating feasibility while leaving questions related to interoperability, usability, and transaction performance unanswered. These studies collectively confirm the potential of blockchain to enhance voting security and transparency but also reveal unresolved challenges related to transaction efficiency, scalability, and practical system implementation.

Beyond architectural considerations, transaction costs and confirmation latency remain critical factors affecting the feasibility of blockchain-based e-voting systems. On public Ethereum networks, transaction execution requires gas fees, which may fluctuate depending on network congestion and protocol design (Aziz & Shukur, 2021). Mechanisms such as Ethereum Improvement Proposal (EIP)-1559 were introduced to stabilize transaction fees; however, their impact on voting-related workloads requires further empirical evaluation (Wood, 2025). Understanding gas consumption and confirmation time is essential for assessing whether blockchain-based voting systems can operate efficiently, economically, and reliably in real-world scenarios. Additionally, scalability considerations must be addressed when evaluating blockchain-based voting systems for different contexts. While public blockchain networks offer maximum decentralization and security, they may face throughput limitations when processing large volumes of simultaneous transactions, as typically occurs during national elections with millions of voters. This research acknowledges these scalability constraints and focuses specifically on evaluating system performance in moderate-scale voting scenarios, such as organizational elections, community decision-making processes, or institutional governance applications where transaction volumes are more manageable. The assessment of blockchain-based voting for large-scale national elections, while an important research direction, is considered beyond the scope of this study and represents an area requiring dedicated investigation with different technical approaches and infrastructure considerations.

Based on the analysis of existing literature, several research gaps can be identified. First, most previous studies focus on conceptual models, simulations, or isolated system components, with limited attention to full end-to-end system deployment using modern full-stack web frameworks such as Next.js. Second, comprehensive evaluations of transaction costs and confirmation times on public Ethereum test networks, such as Sepolia, remain insufficient, despite their importance in determining system feasibility. Third, limited research has examined transaction stability and performance using contemporary Web3 infrastructures, including Web3.js and third-party Ethereum node providers, within practical application environments. Fourth, the balance between transparency and privacy in blockchain-based voting implementations has not been adequately evaluated in deployed systems operating on public test networks.

To address these gaps, this research develops and evaluates a decentralized blockchain-based voting system deployed on the Ethereum Sepolia test network. The proposed system leverages smart contracts to implement voting logic, Next.js to provide a modern and responsive user interface, and Web3.js to facilitate communication between the application and the blockchain. The specific objectives of this study are: first, to design and implement a full-stack e-voting system integrating smart contracts written in Solidity, a modern web framework using Next.js, and Web3 infrastructure through Web3.js; second, to empirically measure and analyze transaction costs in terms of gas consumption and confirmation times under real blockchain network conditions; third, to evaluate the system's security characteristics, including data integrity, immutability, and the mechanisms employed to balance transparency with voter privacy; and fourth, to assess the practical feasibility of deploying blockchain-based voting systems specifically for organizational or community-level elections where moderate transaction volumes are expected.

The main contribution of this research lies in providing comprehensive empirical evidence regarding the performance, cost-efficiency, and security trade-offs of a fully implemented blockchain-based e-voting system

This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

operating in a public test network environment. Unlike previous studies that primarily rely on simulations or theoretical analyses, this research presents quantitative measurements of actual gas consumption, transaction confirmation latency, and system reliability under real network conditions. Furthermore, this study examines how privacy protection mechanisms can be integrated within a transparent blockchain architecture, offering practical insights into the balance between verifiability and voter confidentiality. The findings of this research are intended to assist researchers, system developers, and decision-makers in understanding the practical considerations, technical limitations, and implementation requirements when considering blockchain adoption for digital voting applications in organizational or community contexts. Ultimately, this research aims to contribute to the growing body of knowledge on blockchain-based e-voting systems by bridging the gap between theoretical potential and practical implementation, thereby facilitating more informed decisions regarding the deployment of decentralized voting technologies in real-world scenarios.

2. RESEARCH METHODOLOGY

2.1 Basic Research Framework

This study adopts an engineering-oriented experimental research approach aimed at designing, implementing, and evaluating a decentralized blockchain-based e-voting system. The research does not involve human respondents directly; instead, simulated primary data are used in the form of Ethereum wallet accounts that represent voter identities and predefined candidate data. The research is conducted in a blockchain test environment using the Ethereum Sepolia Testnet as the research setting, which allows controlled experimentation without real financial risk.

The main variables observed in this research include system functionality, transaction efficiency, and transaction cost performance. System functionality variables focus on voter registration, vote casting, and vote counting processes executed through smart contracts. Transaction performance variables include gas usage, gas price, total transaction cost, and transaction confirmation time. The analysis technique applied in this study is quantitative descriptive analysis, where transaction data obtained from multiple voting experiments are measured, calculated, and compared to evaluate system efficiency and stability.

The research framework consists of requirement analysis, system architecture design, system implementation, and testing and performance evaluation. This framework is designed to ensure that the proposed blockchain-based e-voting system meets functional requirements, operates securely in a public blockchain environment, and provides measurable performance outcomes that can be empirically evaluated.

The methodology employed in this research draws from established approaches in blockchain system evaluation that have been documented in recent literature. Gas cost analysis and transaction performance measurement represent common practices in assessing the efficiency of blockchain-based applications, particularly in Ethereum environments where computational costs directly affect system feasibility. Smart contract testing methodologies have evolved to include both functional validation and security assessment as integral components of the development lifecycle, addressing concerns about contract reliability and vulnerability prevention. The performance evaluation metrics applied in this study, specifically gas consumption behavior and transaction confirmation latency, align with evaluation frameworks that have been utilized in prior research on Ethereum-based voting implementations (Hu & Huang, 2025). These methodological considerations inform the empirical approach taken throughout this research.

2.2 Research Stages

This research is conducted through six systematic stages as illustrated in Figure 1. The research process begins with requirement analysis and progresses sequentially through system design, implementation, testing, and evaluation. An iterative refinement mechanism is built into the process, allowing the system to be revisited and improved based on testing outcomes before final completion. Each stage contributes specific outputs that inform subsequent phases of the research.

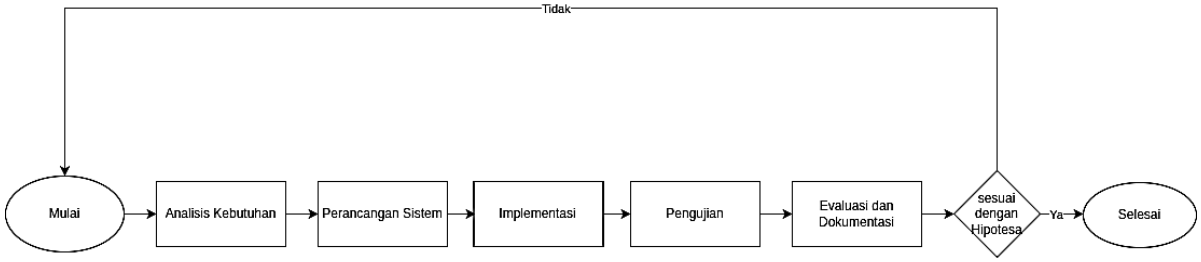

Figure 1. Research Flowchart

Figure 1 shows the research flow used in this study, starting from requirements analysis to identify the needs and constraints of the blockchain-based e-voting system, followed by system design and implementation of the proposed solution. After implementation, testing is carried out to ensure that the core functionalities of the system work as expected. The process continues with evaluation and documentation to assess whether the system results are consistent

Copyright © 2026 the author, Page 1252
This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

with the research hypothesis. If the evaluation results do not meet the expected criteria, the research process may return to earlier stages for improvement; otherwise, the study is concluded once the objectives are achieved.

2.2.1 Stage 1: Requirement Analysis

The initial stage focuses on identifying and documenting both functional and non-functional requirements that the proposed system must satisfy. Functional requirements are determined based on standard voting procedures, which include three primary operations: voter registration to establish eligible participants, vote casting to record individual choices, and vote counting to aggregate results. These functions are designed to be executed through smart contracts deployed on the Ethereum blockchain, ensuring that voting logic operates autonomously without centralized control.

Non-functional requirements address broader system characteristics that affect reliability and usability. Security requirements mandate that voting data remain tamper-proof and resistant to unauthorized modifications. Transparency requirements ensure that voting processes can be verified publicly while maintaining voter privacy through appropriate anonymization mechanisms. Efficiency requirements focus on minimizing transaction costs and confirmation times to make the system economically viable. Cost-effectiveness requirements aim to balance functionality with reasonable gas consumption levels that would allow the system to operate within practical budget constraints on a public blockchain network.

This stage also involves reviewing existing blockchain-based voting implementations documented in recent literature to understand established design patterns and common challenges. Particular attention is given to gas optimization strategies that have been reported in previous Ethereum-based applications, as well as performance metrics that are typically used to evaluate transaction efficiency in decentralized systems.

2.2.2 Stage 2: System Architecture Design

Following requirement analysis, the system architecture is designed to define how different components interact to deliver the required functionality. The architecture consists of four main layers that work together to enable decentralized voting operations. At the frontend, users interact with a web-based interface developed using Next.js, which provides a responsive and intuitive environment for conducting voting activities. The middleware layer utilizes Web3.js library to establish communication between the frontend application and the blockchain network. This library handles transaction signing, contract interaction, and event monitoring.

The smart contract layer contains the core voting logic implemented in Solidity, deployed on the Ethereum blockchain. Smart contracts enforce voting rules programmatically, including voter eligibility checks, single-vote constraints, and result tallying algorithms. The blockchain layer represents the Ethereum Sepolia Testnet, where all voting transactions are recorded permanently in a distributed ledger accessible to network participants. Connection to the Sepolia network is facilitated through a third-party node provider, which offers reliable access to blockchain nodes without requiring local infrastructure.

The complete system architecture is illustrated in Figure 2, showing the sequential flow of data from user interaction through Web3.js middleware to smart contract execution and final storage on the Ethereum blockchain. This design ensures that voting operations benefit from blockchain's inherent characteristics—decentralization, immutability, and transparency—while remaining accessible through familiar web interfaces.

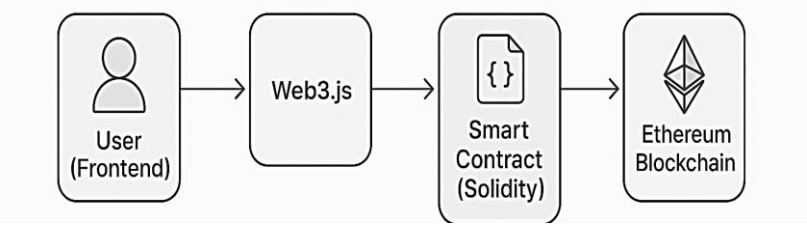

Figure 2. Block diagram design of the blockchain-based e-voting system being developed.

Figure 2 illustrates the system architecture of the proposed blockchain-based voting application. The interaction begins from the user through the frontend interface, where voting actions are initiated and submitted. These requests are then handled by Web3.js, which acts as a bridge between the frontend application and the blockchain layer. Web3.js communicates with the smart contract written in Solidity, which contains the core voting logic such as vote submission and validation. Finally, all validated transactions are recorded and executed on the Ethereum blockchain, ensuring decentralization, immutability, and transparent storage of voting data.

2.2.3 Stage 3: System Implementation

System implementation involves translating the designed architecture into working software components, beginning with smart contract development and progressing to frontend integration.

a. Smart Contract Development and Security Considerations
   The voting smart contract is developed using Solidity version 0.8.20, which includes built-in protections against integer overflow and underflow vulnerabilities. The contract implements core voting functions including voter registration, vote submission, and result retrieval. A critical feature of the implementation is the one-wallet-one-vote

This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

mechanism, enforced through a mapping structure that records whether each Ethereum address has already cast a vote. Once an address submits a vote, it is marked as having voted, and subsequent voting attempts from the same address are rejected by the contract logic.

Security considerations are integrated throughout the development process rather than treated as a separate afterthought. The contract follows established security patterns documented in Solidity best practices, particularly the checks-effects-interactions pattern that helps prevent re-entrancy vulnerabilities. Access control mechanisms are implemented using function modifiers to restrict certain administrative operations to authorized addresses only. Input validation is performed on all user-submitted data before state changes occur, ensuring that invalid or malicious inputs cannot compromise contract integrity.

To validate the security of the implemented contract, a manual code review process is conducted focusing on common vulnerability patterns. The review examines potential attack vectors including unauthorized vote submission, double voting attempts, integer manipulation, and improper access to administrative functions. Each critical function in the contract includes explicit require statements that act as security assertions, validating preconditions before executing state-changing operations. For instance, the vote submission function checks whether the sender address has previously voted, whether the voting period is active, and whether the selected candidate exists before recording the vote.

While automated security analysis tools such as Slither or Mythril provide comprehensive vulnerability detection, the security validation in this study relies primarily on systematic manual review combined with programmatic assertions within the contract code. This approach allows for detailed examination of contract logic in the context of specific voting requirements while ensuring that security checks are explicitly visible in the source code.

b. Frontend Development and Integration

The frontend interface is built using Next.js framework, providing a modern web application environment with server-side rendering capabilities and optimized performance. The interface includes components for wallet connection through MetaMask or compatible Web3 wallets, candidate selection displays, vote submission forms, and result visualization dashboards. Web3.js library is integrated into the frontend to handle blockchain interactions, including contract method calls, transaction signing, and event listening for vote confirmation.

Integration with the Ethereum Sepolia Testnet is configured through environment variables specifying the network RPC endpoint and the deployed smart contract address. This configuration allows the application to communicate with the correct blockchain network and interact with the deployed voting contract. Transaction handling includes proper error management for cases where users reject transactions, insufficient gas fees occur, or network connectivity issues arise.

2.2.4 Stage 4: Testing

System testing is conducted to verify that all implemented functions operate correctly according to specified requirements. The testing approach uses blackbox methodology, where system behavior is evaluated from an external perspective without examining internal code structure. This method simulates how actual users would interact with the system, ensuring that the user experience meets practical usability standards.

Testing covers several critical workflows that represent typical voting scenarios. Wallet connection testing verifies that users can successfully authenticate using MetaMask or compatible wallets and that the system correctly identifies connected addresses. Candidate selection testing ensures that users can view available candidates and select their preferred choice through the interface. Vote submission testing confirms that votes are properly transmitted to the blockchain, recorded in the smart contract, and reflected in the voting results. Vote validation testing checks that the system correctly prevents double voting attempts and rejects votes from unauthorized addresses.

Security-related behaviors are also validated during functional testing to ensure that protective mechanisms operate as intended. Tests are performed to confirm that users cannot submit multiple votes from the same wallet address, that votes cannot be cast outside the designated voting period, and that administrative functions remain inaccessible to regular users. All testing activities are performed on the Sepolia Testnet environment, allowing realistic blockchain interaction without financial risk associated with mainnet deployment.

Throughout the testing phase, transaction data is systematically collected for subsequent performance analysis. Each voting transaction generates a blockchain record containing information about gas consumption, transaction fees, confirmation time, and execution status. This data forms the basis for evaluating system performance characteristics in the next stage.

2.2.5 Stage 5: Evaluation and Documentation

Performance evaluation analyzes the efficiency and cost-effectiveness of the implemented voting system based on empirical transaction data collected during testing. The evaluation focuses on two primary metrics: transaction cost measured in terms of gas consumption and associated fees, and transaction confirmation time measured from submission to blockchain inclusion.

To ensure that performance measurements account for potential variations in blockchain network conditions, a series of 50 independent voting transactions are executed under comparable network states. This sample size is chosen because blockchain performance metrics can fluctuate due to factors such as network congestion, variable gas prices,

Copyright © 2026 the author, Page 1254
This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

and block production rates. A minimum of 50 transactions provides sufficient data points to calculate stable average values and identify patterns in system behavior while remaining practical for a research-scale evaluation.

For each transaction in the sample, several parameters are recorded directly from the blockchain: the amount of gas consumed by the transaction execution, the gas price consisting of base fee and priority fee components, the total transaction cost in ETH, and the timestamp difference between submission and confirmation. These measurements are then aggregated to calculate average values, standard deviations, and consistency indicators that characterize typical system performance.

Results from performance evaluation are compared against the initial requirements defined in Stage 1 to assess whether the system meets its intended objectives. Cost-effectiveness is evaluated by examining whether gas consumption remains within reasonable bounds for practical deployment. Efficiency is assessed by analyzing whether confirmation times are acceptable for typical voting scenarios. Any significant deviations from expected behavior or unexpected cost patterns are documented for further investigation.

2.2.6 Iterative Refinement

The research process incorporates an iterative refinement mechanism represented by the decision point in the flowchart. After testing and evaluation are completed, results are assessed against predefined success criteria to determine whether the system performs adequately. Success criteria include functional correctness (all voting operations work as specified), security compliance (protective mechanisms prevent unauthorized actions), and performance acceptability (costs and confirmation times remain within reasonable ranges).

If evaluation reveals issues such as functional errors, security vulnerabilities, or unacceptable performance degradation, the research process returns to the appropriate earlier stage for correction. For instance, if smart contract logic errors are discovered, the process returns to the implementation stage for code revision. If architectural limitations are identified, the process may return to the design stage for structural improvements. This iterative approach ensures that problems are addressed systematically rather than being overlooked or deferred.

Once all success criteria are satisfied and the system demonstrates stable, secure, and efficient operation, the research proceeds to completion. Final documentation includes comprehensive records of system design decisions, implementation details, testing procedures, performance measurements, and findings regarding the feasibility of blockchain-based voting for the evaluated scenario.

2.3 Performance Measurement and Analysis

System performance evaluation focuses on measuring transaction efficiency during voting operations conducted on the Ethereum Sepolia Testnet (Syaifudin et al., 2024). The evaluation quantifies computational costs and processing times associated with blockchain-based voting to assess economic feasibility and operational responsiveness.

Transaction cost measurement examines the amount of gas consumed when executing voting operations and the corresponding fees paid in ETH. Gas represents the computational effort required to execute smart contract functions, and the cost is determined by multiplying gas usage by the current gas price. The total transaction cost for a single voting operation is calculated using Equation (1):
The total transaction cost is calculated using Equation (1):

$$C_{avg} = G_{used} \times G_{price} \quad (1)$$

where C_{avg} represents the transaction cost in ETH, G_{used} is the amount of gas consumed, and G_{price} is the gas price per unit consisting of the base fee and priority fee.

To characterize typical transaction costs accounting for variations in network conditions, the average cost across multiple transactions is calculated. Performance data is collected from 50 independent voting transactions executed during the testing phase, providing sufficient statistical basis to identify stable patterns while accounting for fluctuations in network congestion and gas price dynamics. The average transaction cost is calculated using Equation (2):

$$\bar{C} = \frac{\sum_{i=1}^{n} C_i}{n} \quad (2)$$

where $\bar{C}$ denotes the average transaction cost (ETH), Ci represents the transaction fee of the i-th transaction, and n refers to the total number of transactions tested. The summation $\sum_{i=1}^{n}$ Ci indicates the total transaction cost accumulated from i = 1 to n. These calculated values are subsequently analyzed to evaluate transaction efficiency and consistency of the proposed blockchain-based e-voting system

Beyond average values, additional statistical measures are calculated to assess performance consistency. Standard deviation is computed to quantify the degree of variation in transaction costs, with lower deviation indicating more predictable and stable cost behavior. Minimum and maximum values are identified to understand the range of possible costs under different network conditions. These metrics collectively provide a comprehensive view of transaction cost characteristics for the blockchain-based voting system.

Transaction confirmation time is measured as the elapsed duration between vote submission by the user and the inclusion of the transaction in a confirmed blockchain block. This metric affects user experience by determining how quickly voters receive confirmation that their votes have been recorded. Confirmation times are influenced by factors

Copyright © 2026 the author, Page 1255
This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

including network congestion, the priority fee offered with the transaction, and the block production rate of the Ethereum network.

The collected performance data undergoes quantitative descriptive analysis to identify patterns and evaluate system efficiency. Cost patterns are examined to determine whether gas consumption remains consistent across transactions or exhibits significant variations. Time patterns are analyzed to assess whether confirmation latency remains within acceptable bounds for practical voting applications. Results are interpreted in the context of the specific use case scenario organizational or community-level voting—to evaluate whether the observed performance characteristics support viable system deployment in such environments.

3. RESULTS AND DISCUSSION

This research was conducted through several sequential stages, as illustrated in Figure 1. The process begins with a requirements analysis stage to identify system needs and security objectives. This is followed by system design, where the architecture, smart contract structure, and interaction flow are defined. The implementation stage involves developing the smart contract using Solidity and integrating it with a web-based frontend using Next.js and Web3.js. Next, the system is tested through functional black-box testing to validate core functionalities. Finally, the evaluation stage analyzes transaction costs and confirmation times, and the results are documented for discussion and conclusion.

Although the smart contract logic was designed to be streamlined in order to reduce gas consumption, security considerations were not compromised. Standard Solidity security practices were applied, including the use of state-change-before-external-call patterns to prevent re-entrancy attacks and built-in overflow protection provided by Solidity version ^0.8.0.

3.1 System Implementation Results

The implementation results indicate that the proposed blockchain-based e-voting system consists of three main components: the frontend application, the smart contract, and the Ethereum blockchain network. The overall system architecture and interaction flow are illustrated in Figure 3, which demonstrates how users interact with the smart contract through the frontend application.

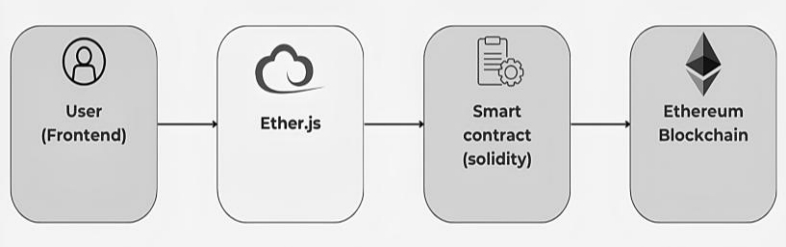

Figure 3. System implementation based on the proposed design in the methodology section

Figure 3 illustrates the interaction flow between the user interface and the Ethereum blockchain within the proposed e-voting system. The frontend allows users to perform voting actions, which are transmitted to the blockchain network through the Ether.js library as the communication layer. Ether.js handles transaction signing and submission, enabling secure interaction between the application and the deployed smart contract. The smart contract, implemented using Solidity, executes the voting logic and enforces predefined rules such as vote validation and data integrity. All validated transactions are then permanently recorded on the Ethereum blockchain, ensuring decentralized storage, immutability, and transparent verification of voting results.

During the implementation phase, users access the system via a web browser and connect their digital wallets using MetaMask. The frontend application, developed using the Next.js framework, provides essential features such as wallet authentication, candidate list presentation, vote submission, and real-time visualization of voting results. Voting transactions are executed through a Solidity-based smart contract deployed on the Ethereum Sepolia Testnet. All voting records are stored permanently on the blockchain, ensuring transparency and immutability, and can be independently verified through the Etherscan platform.

Initially, as described in the methodology section, the interaction between the frontend and the smart contract was planned using the Web3.js library. However, during implementation, Web3.js was replaced with Ethers.js due to its lighter package size, clearer documentation, and better compatibility with the Next.js framework. This modification did not alter the system architecture but significantly improved development efficiency and application stability. The user interface of the implemented e-voting system is presented in Figure 4.

Copyright © 2026 the author, Page 1256
This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

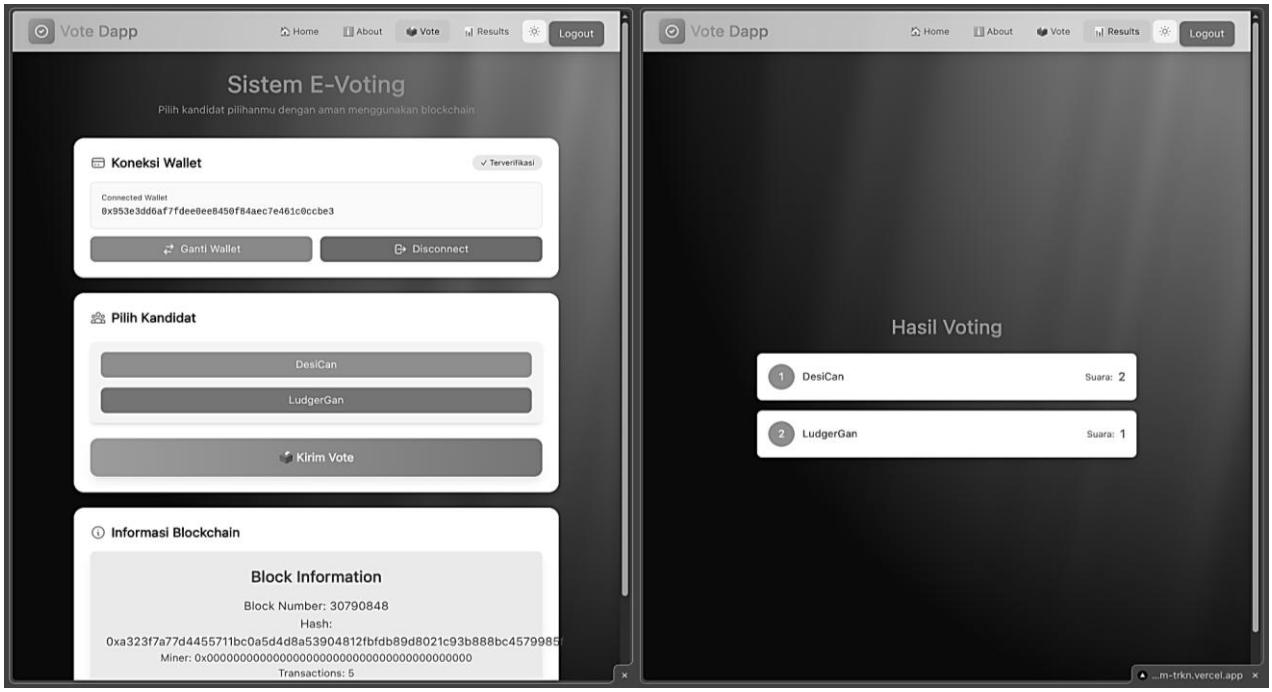

Figure 4. User interface of the blockchain-based e-voting application.

Figure 4 presents the user interface of the developed blockchain-based e-voting application. The interface allows users to connect their digital wallet, select a preferred candidate, and submit their vote through a simple and structured layout. After the voting process is completed, the system displays real-time voting results, showing the number of votes received by each candidate. Additional blockchain-related information, such as block data and transaction details, is provided to support transparency and verifiability. This interface is designed to ensure usability while maintaining secure interaction with the underlying Ethereum blockchain through smart contract execution

3.2 Functional Testing Results

Functional testing was conducted using the Blackbox Testing method to verify that each system function operated according to the specified requirements without examining the internal source code. Similar testing approaches have been widely adopted in previous blockchain-based e-voting studies to validate voting functionality and to prevent multiple voting attempts by the same participant (Mukherjee et al., 2023).

The functional testing results for the main features of the proposed system are summarized in Table 1. The tested features include wallet login, voting execution, voting result visualization, and data security against duplicate voting attempts.

Table 1. Functional testing results of the e-voting system

No	Tested Feature	Test Scenario	Expected Result	Actual Result	Status
1	Wallet Login	User connects MetaMask to the application	Wallet connected and address displayed on the UI	As expected	Passed
2	Voting	User selects a candidate and submits a transaction	Transaction recorded on the blockchain	As expected	Passed
3	Voting Results	User views election results	Vote counts displayed according to on-chain data	As expected	Passed
4	Data Security	Duplicate voting attempt by the same wallet	System rejects the second vote	As expected	Passed

As shown in Table 1, all primary system functions operated as expected. The system successfully authenticated user wallets, recorded voting transactions on the blockchain, accurately displayed voting results based on on-chain data, and effectively prevented duplicate voting attempts from the same wallet address. These results confirm that the proposed e-voting system meets its functional requirements and ensures secure and reliable voting execution.

3.3 Gas Fee Performance Evaluation

Gas fee performance evaluation is a critical aspect in assessing the feasibility of blockchain-based e-voting systems, particularly when deployed on public Ethereum networks (Rathee et al., 2021). Several previous studies have

This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

highlighted that transaction efficiency and smart contract design directly influence gas consumption and system scalability (Sandeep & Chandra Satapathy, 2021).

In this study, gas fee performance evaluation was conducted to measure the transaction cost efficiency of the proposed e-voting system deployed on the Ethereum Sepolia Testnet. A total of six voting transactions were executed using six different wallet accounts. Detailed transaction parameters, including gas usage, gas price, and total gas cost, are presented in Table 2.

Table 2. Voting transaction measurement results on the Sepolia network

No	Wallet Account	Gas Limit	Gas Used	Base Fee (GWEI)	Priority Fee (GWEI)	Gas Price (ETH)	Total Gas Cost (ETH)	Transaction Status
1	0xBc30d...484dA	61,27	60,447	0.000000009	1.5	1.5×10^{-9}	0.000091	Confirmed
2	0x953E3...CcBe3	61,27	60,447	0.000000009	1.5	1.5×10^{-9}	0.000091	Confirmed
3	0xDF386...e15F8	61,27	60,447	0.000000009	1.5	1.5×10^{-9}	0.000091	Confirmed
4	0xbd5b0...3e6BF	61,27	60,447	0.000004956	1.5	$1.500004956 \times 10^{-9}$	0.000091	Confirmed
5	0x0fEdF...aDCC3	95,605	94,647	57.623.524.427	1.5	$59.123524427 \times 10^{-9}$	0.005596	Confirmed
6	0x50896...64b58	78,438	77,547	0.000005891	1.5	$1.500005891 \times 10^{-9}$	0.000116	Confirmed

Based on Table 2, all voting transactions were successfully confirmed on the Sepolia Testnet, indicating stable network interaction and correct smart contract execution.

3.4 Transaction Cost Analysis

The total transaction cost recorded was 0.006076 ETH, resulting in an average transaction cost of approximately 0.001013 ETH per transaction. It should be noted that the performance evaluation in this study was conducted using six voting transactions on the Sepolia test network. This number is sufficient to provide an initial overview of gas consumption and confirmation time behavior; however, it is not intended to represent large-scale voting conditions. Therefore, the results should be interpreted as a preliminary evaluation. Future work will involve increasing the number of transactions (e.g., $n \geq 30$) to allow statistical analysis such as standard deviation measurement and to better account for network fee volatility in public blockchain environments.

3.5 Discussion

The results indicate that the proposed blockchain-based e-voting system is capable of achieving consistent and relatively low transaction costs under Ethereum testnet conditions. As observed in Transactions 1–4 and 6, the gas costs remained stable at approximately 0.0001 ETH, which reflects efficient execution of the smart contract and minimal computational overhead during the voting process. This consistency demonstrates that the core voting functions—wallet validation, vote submission, and result recording—are implemented with a streamlined logic that avoids unnecessary gas consumption. However, the outlier identified in Transaction 5 highlights the inherent dependency of transaction costs on dynamic network conditions, particularly fluctuations in the base fee on public Ethereum networks.

When compared to several existing Ethereum-based e-voting implementations reported in the literature, the proposed system exhibits improved transaction cost efficiency. Previous studies have shown that more complex smart contract structures, additional cryptographic verification steps, and multi-layer validation mechanisms often lead to increased gas consumption (Hajian Berenjestanaki et al., 2024). In contrast, the proposed system adopts an optimized smart contract design with a focus on essential voting operations, which contributes to lower and more predictable transaction costs (Gadekallu et al., 2022). This design choice represents a practical trade-off between functionality and efficiency, especially for small- to medium-scale voting scenarios.

Furthermore, the findings support prior research that suggests the adoption of Layer-2 blockchain solutions, such as Polygon, as a promising approach to further reducing transaction fees while preserving security and transparency (Leinweber et al., 2024). By offloading transaction execution from the Ethereum main layer, Layer-2 solutions can significantly enhance scalability and cost efficiency. Therefore, future research may focus on migrating the proposed e-voting system to a Layer-2 environment and evaluating its performance under larger voter populations and real-world deployment conditions.

4. CONCLUSION

This study describes the design and testing of a decentralized e-voting system built using Ethereum smart contracts, with a focus on transaction costs, confirmation times, and system transparency. The results of the experiment show that the proposed system is quite efficient in terms of gas and has an average transaction confirmation time of around 12 seconds on the Sepolia testnet. This level of latency is suitable for small to medium-scale elections, such as internal

Copyright © 2026 the author, Page 1258
This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

organizational elections, academic voting, or DAO-based governance. However, for large-scale elections with a large number of participants, scalability issues remain a challenge and require integration with Layer-2 solutions or alternative blockchain architectures. In terms of transparency, this system enables publicly verifiable audit trails through immutable blockchain records, thereby increasing trust in the election process. However, balancing voter privacy with transparency remains a challenge that has not been fully resolved. Current implementations already separate voter identities from the votes cast, but advanced privacy protection mechanisms such as zero-knowledge proofs have not yet been implemented and are an important direction for future development. Overall, the results of this study show that blockchain-based e-voting systems are already suitable for use in controlled environments, but further improvements are needed before they can be implemented on a large scale and in applications that are highly sensitive to privacy issues.

REFERENCES

Alvi, S. T., Uddin, M. N., Islam, L., & Ahamed, S. (2022). DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system. *Journal of King Saud University - Computer and Information Sciences*, *34*(9), 6855–6871. https://doi.org/10.1016/j.jksuci.2022.06.014

Aziz, M. J. A., & Shukur, Z. (2021). *Blockchain for Electronic Voting System—Review and Open Research Challenges*. https://www.mdpi.com/1424-8220/21/17/5874

Ethereum gas and fees: Technical overview. (2025). Ethereum.Org. https://ethereum.org/developers/docs/gas/

Gadekallu, T. R., Huynh-The, T., Wang, W., Yenduri, G., Ranaweera, P., Pham, Q.-V., Costa, D. B. da, & Liyanage, M. (2022). *Blockchain for the Metaverse: A Review* (No. arXiv:2203.09738). arXiv. https://doi.org/10.48550/arXiv.2203.09738

Hajian Berenjestanaki, M., Barzegar, H. R., El Ioini, N., & Pahl, C. (2024). Blockchain-Based E-Voting Systems: A Technology Review. *Electronics*, *13*(1), 17. https://doi.org/10.3390/electronics13010017

Hu, B., & Huang, H. (2025). Design of a secure electronic voting system based on zero-knowledge proof and blockchain technology. *Journal of Computational Methods in Sciences and Engineering*, 14727978251361854. https://doi.org/10.1177/14727978251361854

Jain, A. K., Kalra, S., Kapoor, K., & Jangra, V. (2023). Blockchain-Based Secure E-voting System Using Aadhaar Authentication. In H. K. Thakkar, M. Swarnkar, & R. S. Bhadoria (Eds.), *Predictive Data Security using AI: Insights and Issues of Blockchain, IoT, and DevOps* (pp. 89–103). Springer Nature. https://doi.org/10.1007/978-981-19-6290-5_5

Jayakumari, B., Sheeba, S. L., Eapen, M., Anbarasi, J., Ravi, V., Suganya, A., & Jawahar, M. (2024). E-voting system using cloud-based hybrid blockchain technology. *Journal of Safety Science and Resilience*, *5*(1), 102–109. https://doi.org/10.1016/j.jnlssr.2024.01.002

Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani-Hani, A. (2021). Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Networking and Applications*, *14*(5), 2901–2925. https://doi.org/10.1007/s12083-021-01127-0

Kim, H., & Kim, D. (2024). Optimal Gas Fee Minimization in DeFi: Enhancing Efficiency and Security on the Ethereum Blockchain. *IEEE Access*, *12*, 173810–173823. https://doi.org/10.1109/ACCESS.2024.3495714

Leinweber, M., Willig, J., & Schoenfeld, S. A. (2024). *Mastering Crypto Assets: Investing in Bitcoin, Ethereum and Beyond*. John Wiley & Sons.

Mukherjee, A., Majumdar, S., Kolya, A. K., & Nandi, S. (2023). *A Privacy-Preserving Blockchain-based E-voting System* (No. arXiv:2307.08412). arXiv. https://doi.org/10.48550/arXiv.2307.08412

Ohize, H. O., Onumanyi, A. J., Umar, B. U., Ajao, L. A., Isah, R. O., Dogo, E. M., Nuhu, B. K., Olaniyi, O. M., Ambafi, J. G., Sheidu, V. B., & Ibrahim, M. M. (2024). Blockchain for securing electronic voting systems: A survey of architectures, trends, solutions, and challenges. *Cluster Computing*, *28*(2), 132. https://doi.org/10.1007/s10586-024-04709-8

Rathee, G., Iqbal, R., Waqar, O., & Bashir, A. K. (2021). On the Design and Implementation of a Blockchain Enabled E-Voting Application Within IoT-Oriented Smart Cities. *IEEE Access*, *9*, 34165–34176. https://doi.org/10.1109/ACCESS.2021.3061411

Sandeep, K. P., & Chandra Satapathy, S. (2021). *Blockchain Technology: Applications and Challenges*. Springer Cham. https://link.springer.com/book/10.1007/978-3-030-69395-4

Singh, A., Ganesh, A., Patil, R. R., Kumar, S., Rani, R., & Pippal, S. K. (2023). Secure Voting Website Using Ethereum and Smart Contracts. *Applied System Innovation*, *6*(4), 70. https://doi.org/10.3390/asi6040070

Singh, I., Kaur, A., Agarwal, P., & Idrees, S. M. (2024). Enhancing Security and Transparency in Online Voting through Blockchain Decentralization. *SN Computer Science*, *5*(7), 921. https://doi.org/10.1007/s42979-024-03286-2

Syaifudin, Y. W., Prada Aprilia, S., Apriliyanto, S., Rizqiyatul Himmah, D., Siradjuddin, I., Sinal, M., & Aulia Rahmadani, A. (2024). Blockchain-Based E-Voting System: A Decentralized Approach on the Ethereum Private Network. *International Journal of Frontier Technology and Engineering*, *3*(1), 1–18. https://doi.org/10.33795/ijfte.v3i1.6095

This Journal is licensed under a Creative Commons Attribution 4.0 International License

TIN: Terapan Informatika Nusantara
Vol 6, No 8, January 2026, page 1250-1260
ISSN 2722-7987 (Media Online)
Website https://ejurnal.seminar-id.com/index.php/tin
DOI 10.47065/tin.v6i8.8997

Wang, B., Guo, F., Liu, Y., Li, B., & Yuan, Y. (2024). An efficient and versatile e-voting scheme on blockchain. *Cybersecurity*, *7*(1), 62. https://doi.org/10.1186/s42400-024-00226-8

Wang, X., Feng, T., Liu, C., & Fang, J. (2024). Multi party confidential verifiable electronic voting scheme based on blockchain. *Journal of Cloud Computing*, *13*(1), 160. https://doi.org/10.1186/s13677-024-00723-8

Wood, D. G. (2025). *ETHEREUM: A Secure Decentralised Generalised Transaction Ledger*.

Copyright © 2026 the author, Page 1260
This Journal is licensed under a Creative Commons Attribution 4.0 International License