

Real-Time Counting Web Application for DPD IMM Lampung Elections Using Laravel

Muhammad Abdurasyid Dzakirah¹, Adi Wibowo²

^{1,2} Program Studi Sistem dan Teknologi Informasi, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Kotabumi, Indonesia

ARTICLE INFO	ABSTRACT
Article history: Received Sep 29, 2026 Revised Oct 03, 2026 Accepted Oct 09, 2026 Keywords: E-Voting; Laravel Framework; Laravel Reverb; Real-Time Counting; Websocket.	Muhammadiyah Student Association (IMM) Lampung at the Regional Deliberation forum is still conducted conventionally using paper ballots, making printing, casting, and counting relatively time-consuming and prone to recording errors, while live result information is not available to participants. This study aims to design and build a Laravel 12-based Real-Time Counting web application equipped with Laravel Reverb/WebSocket to support the election of the Chairperson and Field Coordinator of DPD IMM Lampung. The system was developed using the waterfall method, modeled with Unified Modeling Language (UML), and supported by a MySQL database. The implementation integrates voter-right validation using membership card numbers and tokens, digital ballot casting, anti-double-vote enforcement through a database unique constraint, real-time result broadcasting through the VoteCasted event, and a re-voting mechanism for tied results. Black Box Testing and User Acceptance Testing (UAT) were defined as evaluation procedures for assessing functional suitability and user acceptance. The article presents the testing scenarios and UAT indicators used in the evaluation. The implemented system provides the functional basis for replacing the manual ballot workflow and presenting election results more quickly and transparently. The evaluation included 14 Black Box Testing scenarios and UAT with 20 respondents (15 voters/participants and 5 committee/administrator users). The UAT obtained a total score of 706 out of 800, equivalent to 88.25%, categorized as Very Good. The Black Box table documents the 14 functional scenarios and expected outputs; quantitative pass/fail execution totals were not separately recorded in the source testing table. <i>This is an open access article under the CC BY-NC license.</i>



Corresponding Author:

Muhammad Abdurasyid Dzakirah,
Sistem dan Teknologi Informasi/ Fakultas Teknik dan Ilmu Komputer,
Universitas Muhammadiyah Kotabumi/UMKO
Jalan Hasan Kepala Ratu No. 1052, 34517 Sindangsari Indonesia.
Email: rasidktb12@gmail.com

1. INTRODUCTION

The development of information technology has accelerated digital transformation across various sectors, including education and student organizations. Digital transformation supports administrative efficiency, transparency, accountability, and improved decision-making through web-based information systems (Adham, 2024). One organization conducting periodic leadership elections is the Muhammadiyah Student Association (IMM). The election of the General Chairperson of the Regional Leadership Board (DPD IMM) is conducted through the Regional

Deliberation (Musyawarah Daerah Musyda), which requires a democratic, transparent, honest, and fair process.

Based on interviews with the DPD IMM Lampung organizing committee, elections are traditionally conducted through consensus and paper-ballot voting when consensus cannot be reached. This conventional process requires manual ballot distribution, voting, collection, verification, and counting, which can increase processing time, human error, and committee workload. Each participant has two voting rights: one for the General Chairperson and one for the Head of Division. If a tie occurs, voting must be repeated until the 50% + 1 majority requirement is achieved. The deliberation involves six Branch Leadership Boards, with each branch delegating seven voting representatives.

Web-based e-voting can integrate voter registration, voting-right validation, digital voting, data storage, and result recapitulation into a single system. Recent e-voting literature also emphasizes the importance of security, privacy, transparency, and verifiability in electronic election systems (Kho et al., 2022) ; (Vladucu et al., 2023) . WebSocket technology further enables election results to be updated in real time without requiring page refreshes (Kadim et al., 2023) . However, previous studies have generally focused on digitalizing voting processes without integrating real-time counting, hierarchical voting rights, and automated tie-resolution mechanisms. The research gap is therefore not simply the absence of an e-voting application, but the lack of an integrated workflow that simultaneously accommodates organization-specific voting rights, two election stages, database-level duplicate-vote prevention, real-time result broadcasting, and automated tie resolution. The innovation of the present study lies in translating these DPD IMM Lampung election rules into one Laravel-based workflow rather than introducing Laravel Reverb/WebSocket as a standalone technology.

Therefore, this study develops a Real-Time Counting Web Application for the Election of the General Chairperson of DPD IMM Lampung using Laravel, incorporating Laravel Reverb/WebSocket. The system integrates KTA and token-based voter validation, two types of voting rights, digital ballots, real-time result broadcasting, double-vote prevention, and a re-election mechanism for tied candidates. This integration is expected to accelerate vote counting, reduce recapitulation errors, and improve transparency and accountability in the Regional Deliberation process.

An information system consists of interconnected people, procedures, hardware, software, databases, and communication networks that collect, process, store, and distribute information to support organizational activities (Adham, 2024). E-voting utilizes electronic technology to support election processes, including voter validation, voting, vote storage, counting, and result reporting. Real-time counting enables vote results to be updated immediately through bidirectional communication technologies such as WebSocket (Bakhtiar et al., 2023) ; (Nurdiansyah et al., 2024).

The system uses Laravel, a PHP framework based on the Model-View-Controller (MVC) architecture, together with Laravel Reverb as the WebSocket server and Laravel Echo on the client side. MySQL is used as the relational database, while UML is employed to model system structure and processes. Development follows the Waterfall method, consisting of requirements analysis, system design, implementation, testing, and deployment and maintenance (Nurdiansyah et al., 2024). System quality is evaluated using Black Box Testing with equivalence partitioning and User Acceptance Testing (UAT) using a Likert-scale questionnaire (Akbar, 2022); (Ansfridus et al., 2024); (Novita & Helena, 2021) ; (Rosmiati, 2021) ; (Wulandari et al., 2022).

2. RESEARCH METHOD

The research was conducted at the Secretariat of the Regional Leadership Board of the Muhammadiyah Student Association (Dewan Pimpinan Daerah Ikatan Mahasiswa Muhammadiyah DPD IMM) of Lampung, which served as the location for initial data collection through interviews with the organizing committee of the Regional Deliberation (Musyawarah Daerah). Data were collected using three methods: direct interviews with the committee to obtain information regarding the existing election mechanism, challenges encountered, and organizational regulations concerning voting rights, invalid ballots, and the resolution of tied results; observation of the Regional Deliberation process, including the number of Branch Leadership Boards and their delegates; and a literature review of relevant journals, books, and official documentation related to

Laravel, Laravel Reverb, and MySQL technologies, particularly in the context of e-voting and real-time counting.

Based on the results of the interviews and observations, system requirements were classified into functional and non-functional requirements. The functional requirements include validation of participants' voting rights using their Membership Identification Card (KTA) number and token; provision of digital ballots according to the currently active election stage, with a mechanism to prevent voters from casting more than one vote within the same stage (**anti-double-vote**); storage of votes, including the identification of invalid votes; real-time broadcasting of vote-count updates to the public dashboard; management of administrator/committee, Branch Leadership Board, voter, and candidate data for each round; management of election rounds; provision of a re-election mechanism specifically for tied candidates; and presentation of result recapitulations that can be printed as official election minutes.

The non-functional requirements include short-interval (near real-time) result broadcasting through WebSocket technology; security mechanisms involving Cross-Site Request Forgery (CSRF) protection and hashed credentials; a simple and responsive user interface; portability as a web-based application built using Laravel 12; and compatibility with MySQL and PHP 8.2 or later.

2.1 Research Position and Related Studies

Previous studies demonstrate the feasibility of web-based e-voting, Laravel-based election applications, and WebSocket-based result reporting. Table 1 positions the present study in relation to these works and highlights the combination of voting-right validation, real-time counting, duplicate-vote prevention, and tie-resolution addressed by the proposed application.

Table 1. Previous Studies and the Position of the Present Study

Study	Focus	Limitation	Position of the Present Study
(Nurdiansyah et al., 2024)	Web-based e-voting, Waterfall, Django/MySQL	No real-time functionality; results are presented after recapitulation	Result broadcasting using WebSocket
(Sika et al., 2024)	Laravel-based student e-voting	Single voting right; no automated tie-handling mechanism	Two-stage election and re-voting mechanism
(Ardiansyah et al., 2023)	Muhammadiyah e-voting based on Laravel + RFID	Focused on authentication; results are not broadcast in real time	Real-time counting using Reverb/WebSocket
(Paskalis Bachri, 2024)	Laravel-based student-organization e-voting	No real-time functionality and no hierarchical voting rights	Integration of voting rights, real-time counting, and tie handling

The contribution of this study is not the use of Laravel Reverb alone. Rather, it is the integration of voter-right validation, digital voting, duplicate-vote prevention, real-time result broadcasting, and tie-resolution within one election workflow. The design is specifically adapted to the characteristics of the DPD IMM Regional Deliberation, making the organization's election rules part of the system requirements.

2.2 System Development Method

The system was developed using the Waterfall method, which consists of five stages: requirements analysis, system design, implementation, testing, and deployment and maintenance. The system design stage employed Unified Modeling Language (UML), including a use case diagram illustrating the interactions of three main actors: Voter, who can validate voting rights, complete the digital ballot, and view preliminary results; Admin/Committee, who can manage Branch Leadership Board data, voters, candidates, and election rounds, as well as print result recapitulations; and Public, who can access the real-time results dashboard without authentication, as illustrated in Figure 1. UML is widely used to represent system structure and interaction in information-system design (Setiaji & Sastra, 2021); (Siking et al., 2023). The Waterfall method was selected because the main election rules and functional requirements were identified through interviews and observation before implementation, while the development scope was clearly structured into requirements, design, implementation, testing, and deployment/maintenance. This sequential structure also supports documentation and traceability of organization-specific rules such as voting rights, election-round status, and the 50% + 1 tie-resolution requirement. It is not claimed to be superior to iterative methods in general; the choice is based on the relatively defined scope of this prototype.

The design also includes an activity diagram describing the voting process and tie-result handling, a class diagram illustrating the data structures and relationships among the entities *Branch*, *Voter*, *ElectionRound*, *Candidate*, *Vote*, and *User*, and a sequence diagram depicting interactions among objects during the real-time result update process, beginning with vote storage by the *VotingController* and continuing with the broadcasting of the *VoteCasted* event through Laravel Reverb to subscribed clients.

Table 2. System Requirements

Category	Main Requirements
Functional	KTA number and token validation; digital ballot; anti-double-vote mechanism; storage of valid/invalid votes; real-time counting; management of committee, branches, voters, candidates, and election rounds; re-election for tied results; result recapitulation/election minutes.
Non-functional	Near real-time communication through WebSocket; CSRF protection and credential hashing; simple and responsive interface; portability based on Laravel 12; compatibility with PHP 8.2+ and MySQL.

The requirements analysis served as the basis for mapping system features into the use case, activity diagram, database structure, and interface design. Thus, the functional requirements extend beyond the voting process itself and encompass the entire election management cycle, from voter registration to result documentation.

2.3 Voting Process Design

The voting process design begins with voter identity validation. The voter enters the KTA number and token provided by the committee. The system matches these two data elements against the *voters* table and checks whether voting rights for the currently active election type are still available. If the credentials are valid, the voter is directed to the digital ballot. If the credentials are invalid or the voting right has already been used, the system denies access to the voting stage. This approach transfers the previously manual physical KTA verification process into a structured validation mechanism within the application.

The voting-rights model was designed to accommodate two election stages, namely the General Chairperson and Head of Division elections. The *has_voted_ketua_umum* and *has_voted_ketua_bidang* status fields in the voter data are used to help control voting rights at the application level. At the database level, a unique constraint on the combination of *election_round_id*, *voter_id*, and *type* provides an additional layer of protection, ensuring that a voter cannot cast more than one vote for the same election type and round. The combination of application-level and database-level controls is important because validation does not rely solely on the user interface.

The election-round design allows the committee to create, open, and close election stages in a controlled manner. The status of each round is classified as **draft**, **open**, or **closed**, enabling the system to determine when digital ballots can be made available. Once a round is closed, the results can be recapitulated and published. If two candidates obtain the same highest number of votes, the system provides a mechanism for creating a **re-election round** focused specifically on the tied candidates. This mechanism translates the organization's regulations concerning the resolution of tied results into an application-level business process. The re-voting mechanism is necessary because the DPD IMM Lampung rule requires a repeated vote when candidates obtain the same highest result and a 50% + 1 majority has not been achieved. Integrating re-voting into the application prevents the committee from having to recreate the election stage manually, preserves the tied-candidate scope, and keeps the resolution process traceable within the same election workflow.

2.4 Real-Time Communication

On the communication side, the selection of WebSocket technology is based on the need to deliver result updates to multiple clients without relying on periodic polling. Laravel Reverb functions as the WebSocket server, while Laravel Echo is used on the browser side. The *VoteCasted* event carries the calculated results retrieved again from the database after the vote-storage transaction has been successfully completed. Therefore, the broadcast data are not

derived from temporary values on the user interface but are recalculated based on the stored database records, thereby improving result consistency.

2.5 User Interface Design

The user interface is divided into three user groups. The voter interface is designed to support a short and straightforward process consisting of validation, candidate selection, and confirmation. The public interface focuses on result readability through bar charts, total numbers of valid and invalid votes, and a WebSocket connection status indicator. Meanwhile, the admin/committee interface focuses on data management and election-round control, including branch management, voter management, candidate management, opening and closing election rounds, result recapitulation, and handling of tied results. This separation prevents operational functions from being mixed with monitoring functions.

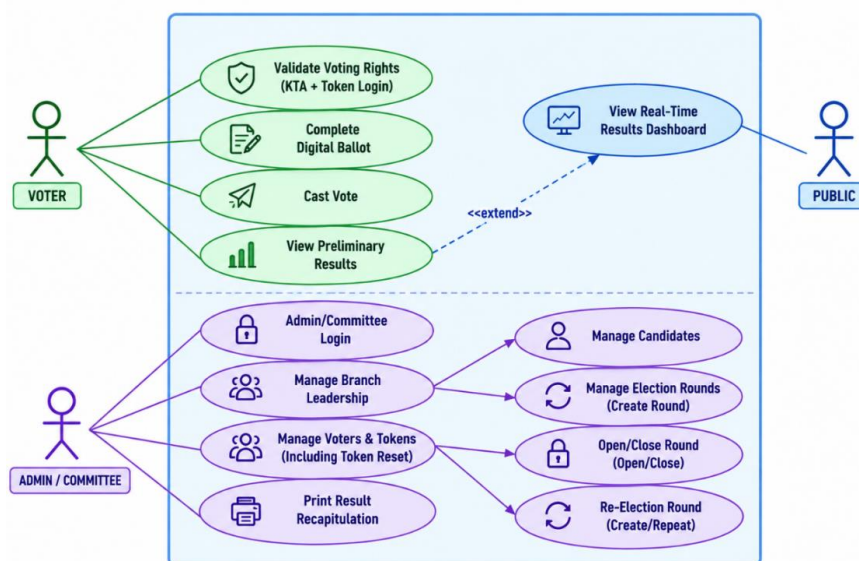


Figure 1. System Use Case Diagram

2.5 Database Design

The database was designed using MySQL with six main interrelated tables, namely *users*, *branches*, *voters*, *election_rounds*, *candidates*, and *votes*, as illustrated in Figure 2. The unique columns in the *voters* table (*cta_number* and *token*) ensure that each voter has a distinct identity and credential, while the composite unique constraint in the *votes* table (*election_round_id*, *voter_id*, and *type*) serves as the primary mechanism for preventing **double voting** at the database level. Consequently, the prevention mechanism remains effective even when simultaneous requests occur on the application side.

The interface design follows the principles of simplicity and ease of use, considering that system users have diverse technical backgrounds. The interface is divided into three groups of pages: voter pages for voting-right validation, digital ballots, and confirmation; public pages for the real-time results dashboard; and admin/committee pages for the dashboard, branch management, voter management, candidate management, election-round management, and result recapitulation. The interface uses the Tailwind CSS framework to ensure a consistent and responsive appearance across various screen sizes.

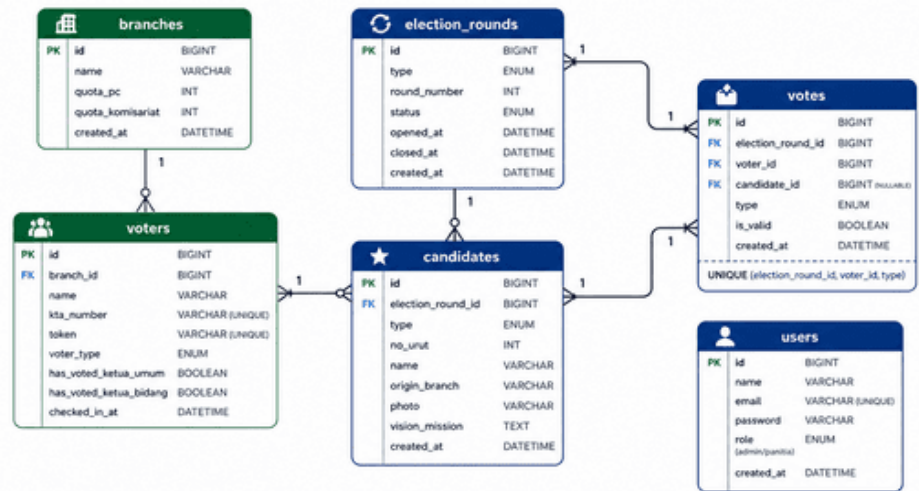


Figure 2. System Entity Relationship Diagram (ERD)

2.6 Real-Time Counting Implementation

The real-time counting technology was implemented using Laravel Reverb, which operates as a separate WebSocket server on the backend, and the Laravel Echo library, which operates on the client side. The implementation workflow is as follows: when a vote is successfully stored, the controller invokes the `broadcast()` function for the `VoteCasted` event; the event carries result data that are recalculated directly from the database to ensure accuracy; Laravel Reverb receives and broadcasts the event to all clients subscribed to the `results.{type}` channel; and on the client side, Laravel Echo, which is connected to Reverb, receives the broadcast data and immediately updates the total vote counts and bar charts using JavaScript without requiring the page to be reloaded.

The system quality was subsequently evaluated using two testing methods: Black Box Testing with the equivalence partitioning technique applied to the system's main functions, including voting-right validation, digital ballot completion and storage, anti-double-vote mechanisms, real-time result updates, administrator login, and data management through the admin panel; and User Acceptance Testing (UAT) using a five-point Likert-scale questionnaire distributed to voter and committee respondents to assess ease of use, functional suitability, result presentation speed, and overall user satisfaction. The use of equivalence partitioning for functional testing is consistent with previous Black Box Testing studies (Ansfridus et al., 2024) ; (Rosmiati, 2021) ; (Wulandari et al., 2022).

3. RESULTS AND DISCUSSIONS

The Real-Time Counting Web Application for the Election of the General Chairperson of DPD IMM Lampung was implemented using PHP 8.2, Laravel 12, Laravel Reverb as the WebSocket server, MySQL 8 as the database, Tailwind as the CSS framework, and Laravel Echo and Chart.js on the client side. The program code follows the standard Laravel directory structure. The `app/Models` directory contains six model classes *Branch*, *Voter*, *ElectionRound*, *Candidate*, *Vote*, and *User* representing the database structure shown in Figure 2. The `app/Http/Controllers` directory contains controllers for voter and administrator authentication, the `VotingController` for the voting process, the `ResultController` for presenting results, and five controllers for managing committee-related data. The `app/Events` directory contains the `VoteCasted` class, which serves as the core component of the real-time counting mechanism, while the `app/Http/Middleware` directory contains `EnsureVoterAuthenticated`, which restricts access to the voting page to validated voters only.

The database structure shown in Figure 2 was implemented using Laravel's migration feature, making the database schema version-controlled. Relationships between tables were established using foreign key constraints, while initial data including Branch Leadership Board data, committee accounts, sample voters, and candidates were provided through the `DatabaseSeeder` class. This enables the system to be immediately tested or simulated.

3.1 User Interface Implementation

On the interface side, the voter page provides a form for validating the KTA number and token against the data stored in the voters table. Once successfully validated, the voter is redirected to the digital ballot page, which displays the candidates participating in the currently active election stage, including their serial numbers, branch affiliations, and summaries of their visions and missions. A confirmation page is displayed after the vote has been submitted.

The public page consists of a real-time results dashboard, as shown in Figure 3. The dashboard can be accessed without authentication and is designed to be displayed on a large screen in the deliberation hall. It presents bar charts showing the number of votes obtained by each candidate, the total number of valid and invalid votes, and an indicator showing the status of the active WebSocket connection. Because the public dashboard is intentionally accessible without authentication, its design must expose only aggregate election information needed for transparency. The dashboard does not provide voter identity, KTA/token credentials, or an individual voter's candidate choice. Nevertheless, public exposure creates a security consideration because result timing and aggregate vote changes are visible to any connected client. Therefore, access to voting and administrative functions remains authenticated, while the public channel should be limited to result data and protected against unauthorized publication of confidential records. The WebSocket connection indicator should be understood as a connectivity indicator, not as proof of confidentiality or cryptographic security.

The committee accesses the system through an administrator login page that uses Laravel's built-in session-based authentication. After authentication, administrators are directed to a summary dashboard displaying statistics on registered voters, the number of voters who have checked in, and the status of each election round. Administrators can also manage Branch Leadership Board data, voter data and their access tokens, candidate data for each round, and result recapitulations that can be printed as official election minutes. The system also provides a button for automatically creating a re-election round specifically for candidates who obtain tied votes in accordance with the 50% + 1 vote provision.

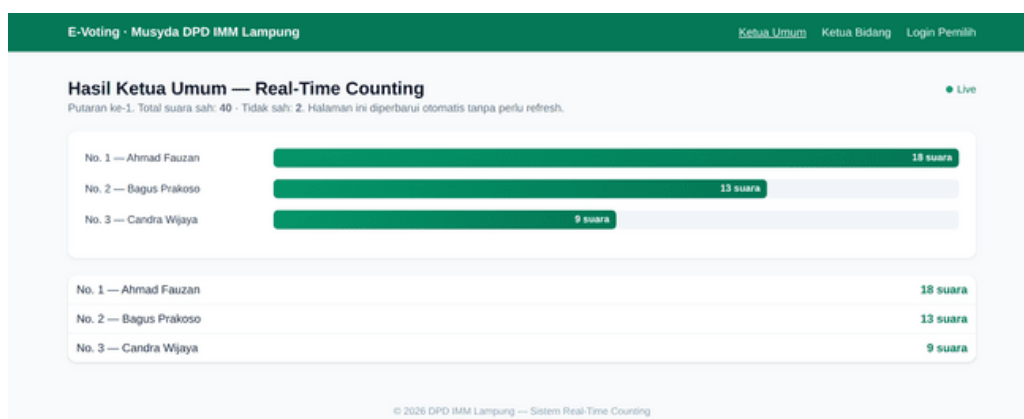


Figure 3. Real-Time Results Dashboard

3.2 Real-Time Counting Implementation

The real-time counting feature represents the core implementation addressing the research problem concerning the automatic updating of voting results. In the VotingController, each vote submitted by a voter is stored in the database through a transaction using `DB::transaction`. The transaction includes both the insertion of a new record into the votes table and the updating of the voter's "already voted" status in the voters table. Consequently, both operations either succeed or fail together, preventing inconsistent data states. Immediately after the transaction is successfully completed, the system invokes: `broadcast(new VoteCasted($round))`; The event class implements the `ShouldBroadcastNow` interface, allowing the event to be broadcast immediately without passing through a queue. The `broadcastOn()` method determines the public channel results.{type}, while the `broadcastWith()` method recalculates the number of valid votes obtained by each candidate

directly from the database before transmitting the data. Therefore, the data received by clients always reflect the current database state.

On the client side, the results dashboard loads the Laravel Echo library, which connects to Reverb using the credentials specified in the configuration file. When an update event is received, the JavaScript script on the page immediately updates the total number of votes, the bar chart, and the vote count displayed for each candidate without requiring a page refresh.

The same mechanism is applied when the committee closes an election round. The VoteCasted event is also broadcast so that the final status and final results are transmitted to the public dashboard.

3.3 System Testing

System testing in this study was designed using Black Box Testing with the equivalence partitioning technique across fourteen scenarios. These scenarios cover voting-right validation under valid and invalid conditions, completion and storage of digital ballots, the anti-double-vote mechanism, real-time result updates on the dashboard when a new vote is submitted from another client, administrator login under valid and invalid credentials, management of Branch Leadership Board data, voters, candidates, and election rounds, handling of tied results through the creation of a re-election round, and printing of result recapitulations as official election minutes.

The User Acceptance Testing (UAT) instrument used a five-point Likert-scale questionnaire consisting of eight indicators covering interface usability, voting-right validation, digital ballot completion, result-update speed, data-management suitability, acceleration of result recapitulation, transparency, and overall user satisfaction. The Black Box Testing table presents the defined scenarios and expected outputs, while the UAT table presents the indicators used to assess usability, voting-right validation, ballot completion, result-update speed, data-management suitability, transparency, and overall satisfaction. For the UAT, 20 actual respondents were used, consisting of 15 voters/participants and 5 committee/administrator users. This composition represents both major user roles in the application. The study uses purposive user representation for formative acceptance evaluation, rather than claiming that 20 respondents constitute a statistically generalizable sample of all IMM members. The UAT produced 706 points from a maximum of 800 points, or 88.25%, categorized as Very Good. The 14 Black Box scenarios in Table 3 define the functional coverage; the source record does not provide a separate numerical pass/fail aggregate, so no unsupported pass-rate percentage is claimed. The workbook records the actual working data for all 20 respondents, including 15 Pemilih/Peserta and 5 Panitia/Admin, and the item-level totals produce an aggregate score of 706/800 (88.25%). These values are the quantitative UAT results used in the revised article.

Table 3. Black Box Testing Scenarios

No.	Function	Brief Scenario	Expected Output
1	Voter Login	Valid KTA number and token	Access to the digital ballot
2	Voter Login	Invalid KTA number/token	Access denied and an error message displayed
3	Voter Login	Empty KTA number or token	Required-field validation is displayed
4	Digital Ballot	Select a candidate and submit	Vote is stored and voting-right status is updated
5	Anti-Double-Vote	Submit a vote during an election stage that has already been used	System rejects repeated voting
6	Real-Time Counting	A new vote is submitted while another client's dashboard is open	Vote count/chart changes without refreshing
7	Administrator Login	Valid credentials	Access to the committee dashboard
8	Administrator Login	Invalid credentials	Remains on the login page and displays an error message
9	Branch Management	Add complete branch data	Data are stored and displayed
10	Voter Management	Register a voter	Voter data and access token are generated
11	Candidate Management	Add a candidate to an election round	Candidate appears in the corresponding round
12	Election Round Management	Open and close an election round	draft → open → closed status is stored
13	Result Recapitulation/Re-election	Two candidates obtain the same number of votes	Tie notification and re-election option are displayed
14	Print Recapitulation	Print election results	Recapitulation can be printed as official

election minutes

The table presents the functional testing scenarios and their expected outputs, covering the principal voter, administrator, election-round, result, and re-election functions of the application.

3.3 User Acceptance Testing

Table 4. UAT Indicators

No.	UAT Indicator
1	The application interface is easy to understand and use (usability).
2	Voting-right validation using the KTA number and token is easy to perform.
3	Completing and submitting the digital ballot is easy to understand.
4	Voting results on the dashboard are updated rapidly in real time without refreshing the page.
5	The branch, voter, candidate, and election-round management features meet user requirements.
6	The system helps accelerate result recapitulation compared with the manual process.
7	The system improves transparency in the election process for Regional Deliberation participants.
8	Users are generally satisfied with the application.

The UAT instrument consists of eight indicators covering usability, voting-right validation, digital ballot completion, real-time result updates, data-management suitability, result-recapitulation support, transparency, and overall satisfaction. These indicators provide the basis for evaluating user acceptance of the developed application. Quantitative UAT result based on the actual working data. The evaluation involved 20 respondents, comprising 15 voters/participants and 5 committee/administrator users. Using eight indicators and a five-point Likert scale, the maximum score was $20 \times 8 \times 5 = 800$. The obtained score was 706, giving $(706/800) \times 100\% = 88.25\%$, which falls in the Very Good acceptance category (81–100%). The detailed respondent-level scores and item-level recapitulation are recorded in the UAT workbook.

Table 5. Actual UAT Results

No.	UAT Indicator	Total Score	Maximum Score	Percentage
1	Interface usability	89	100	89.00%
2	Ease of voting-right validation using KTA and token	86	100	86.00%
3	Ease of completing and submitting the digital ballot	88	100	88.00%
4	Speed of real-time result updates without refresh	91	100	91.00%
5	Suitability of branch, voter, candidate, and election-round management	85	100	85.00%
6	Acceleration of result recapitulation compared with the manual process	88	100	88.00%
7	Transparency of the election process	89	100	89.00%
8	Overall user satisfaction	90	100	90.00%
	Total / overall	706	800	88.25%

The item-level totals above are calculated directly from the respondent-level worksheet: for each indicator, the frequencies of STS, TS, CS, S, and SS are weighted 1–5. Across all eight indicators, the actual working data yield 706 points from a maximum of 800 points. Actual group breakdown: 15 Pemilih/Peserta obtained 528 points out of 600 (88.00%), while 5 Panitia/Admin obtained 178 points out of 200 (89.00%). The combined score is 706 out of 800 (88.25%), categorized as Very Good.

3.4 Discussion

The main contribution is the integration of organizational election rules and technical real-time mechanisms into one workflow: KTA/token validation controls voting rights, separate status fields and a database unique constraint reduce duplicate voting, VoteCasted with Reverb/Echo distributes recalculated aggregate results, and a dedicated re-election round implements the tie rule. This combination directly addresses the operational characteristics of the DPD IMM Lampung Regional Deliberation and is the principal difference from prior systems that generally address only subsets of these functions.

The implementation results show that the system integrates voting-right validation, digital voting for the General Chairperson and Head of Division, and result recapitulation, providing a functional alternative to paper ballots. Vote results are updated immediately through the VoteCasted event after votes are stored. The system also prevents double voting through application-level `has_voted` status and database-level unique constraints and provides a re-election mechanism for tied results. The quantitative UAT result of 88.25% provides supporting evidence that the integrated workflow is accepted by the participating users. However, the result does not by itself demonstrate cryptographic security, ballot secrecy, or scalability. Black Box Testing provides functional conformity for the defined scenarios, while the absence of a separate numerical pass/fail execution record limits the strength of any quantitative claim about functional success.

The evaluation design combines functional Black Box Testing with UAT to assess both system functions and user acceptance. The Black Box scenarios and UAT indicators are aligned with the functional and non-functional requirements identified during the requirements analysis. The system also has limitations, including the absence of blockchain and advanced cryptographic mechanisms and the lack of large-scale load testing. These limitations are relevant because recent e-voting research identifies privacy, verifiability, integrity, and scalability as important requirements for more advanced election systems (Pereira et al., 2023).

a. Software Architecture

The Laravel Model-View-Controller (MVC) architecture separates data management, interface presentation, and request control. This structure supports modular development, including voter, candidate, election-round, and result modules, while real-time broadcasting is separated from the voting process. This architecture also facilitates future maintenance when election rules change. The selection of Laravel is consistent with prior web-application development studies using the framework (Kadim et al., 2023) ; (Sinlae et al., 2024) .

b. Data Integrity

Data integrity is maintained through DB::transaction, which treats vote storage and voting-right updates as a single operation. If an operation fails, the transaction can be rolled back to prevent inconsistent data. The database-level unique constraint provides additional protection against duplicate votes, including nearly simultaneous requests. The security implication of the public dashboard is therefore a balance between transparency and confidentiality. Aggregate counts can be public, but voter credentials, voter identity, and individual vote selections must remain outside the public channel. The present implementation supports this separation at the interface level, but stronger security assurance would require additional security testing and, for higher-stakes deployment, advanced cryptographic mechanisms and formal privacy/verifiability analysis.

c. Transparency

The public dashboard can be accessed without authentication, allowing participants to monitor results without having permission to modify election data. The WebSocket connection indicator shows whether the update channel is active. However, this indicator does not guarantee system security or data integrity, which remain areas for further development. The most important contribution for the IMM Lampung election process is the conversion of organization-specific election rules into an executable, integrated digital workflow. The system does not merely digitize ballot casting; it connects eligibility validation, two voting rights, duplicate-vote prevention, real-time aggregate counting, election-round control, tie resolution, and official result recapitulation. The

practical value is reduced manual handling and faster visibility of aggregate results while retaining the committee's control over election rounds.

d. Research Contribution

The research contributes at both technological and business-process levels by integrating Laravel 12, MySQL, Laravel Reverb, Laravel Echo, and Chart.js with DPD IMM Lampung election rules, including KTA and token validation, two voting rights, and tie resolution. Thus, the system functions not only as a digital voting application but also as an integrated election management system covering validation, voting, real-time result broadcasting, round management, and result documentation. The priority improvements are (1) stronger cryptographic protection and privacy-preserving mechanisms for vote integrity and ballot secrecy, (2) formal security testing including vulnerability assessment and penetration testing, (3) large-scale load/stress testing of Laravel Reverb and the database under concurrent voting and dashboard connections, and (4) stronger verifiability and audit mechanisms. These improvements are more important for production deployment than adding additional user-interface features.

e. Recommendations for Further Research

Future research should conduct broader validation with voters and committee members using the defined Black Box Testing and UAT procedures, including respondent numbers, scores, percentages, and acceptance categories. Load testing should also be performed to evaluate simultaneous WebSocket connections, broadcasting response time, and system stability under higher transaction loads. Future development may also examine privacy-preserving and scalable e-voting approaches reported in recent literature (Emami Neishabouri et al., 2023). These evaluations would provide stronger evidence of system effectiveness and readiness for actual Regional Deliberation implementation. A Progressive Web App is also a relevant direction for extending web access to mobile users (Apriyanti & Ramli, 2025).

Table 5. Comparison between the Conventional Mechanism and the Developed System

Aspect	Conventional Mechanism	Developed System
Voter Validation	Physical KTA verification	KTA number + token
Voting	Paper ballot	Digital ballot
Double-Vote Prevention	Manual verification	Voting-right status + unique constraint
Vote Counting	Manual counting and committee recapitulation	Data calculated from the database
Result Presentation	After recapitulation	Real-time through Reverb/WebSocket
Tied Results	Manual re-voting	Dedicated re-election round for tied candidates
Documentation	Manual preparation	Recapitulation can be printed as official election minutes

The comparison demonstrates that the main contribution of the system lies in the integration of the voting process with real-time result presentation and the management of election rules. The table is not intended as a quantitative measurement of processing time but rather as a mapping of the transition from a manual process to a computerized process.

f. Database Implementation

The database implementation uses Laravel migrations to produce a version-controlled database structure, enabling schema changes to be managed in a more structured manner. Foreign key relationships connect branches with voters, election rounds with candidates, and voters and candidates with vote records. The seeder is used to provide initial data, including Branch Leadership Board data, committee accounts, sample voters, and candidates, allowing the system to be immediately used in a testing or simulation environment. This structure demonstrates that the database serves not only as a storage mechanism but also as an integral component of the validation and integrity mechanisms of the voting process. The relational-database approach is consistent with the use of MySQL in web-based information-system development (Dachi & Suhada, 2025).

g. Voter Page Implementation

The implementation of the voter page demonstrates a simpler workflow than the manual paper-ballot process. After successful validation, the voter only sees the candidates participating in the active round, together with their serial numbers, branch affiliations, and summaries of their visions and missions. After the vote is submitted, the system displays a confirmation page. This sequence reduces the committee's need to physically distribute, collect, and inspect paper ballots. From a business-process perspective, the digitalization also allows voting-right status to be recorded directly in the database.

h. Public Dashboard Implementation

The public dashboard is the component that most clearly distinguishes this system from e-voting applications that display results only after the recapitulation process. When a new vote is successfully stored, the VoteCasted event is broadcast to the channel corresponding to the election type. The browser, which has subscribed through Laravel Echo, receives the event and updates the total vote count and Chart.js visualization without requiring a page refresh. In the context of the Regional Deliberation, the dashboard can be displayed on a large screen in the deliberation hall, allowing participants to obtain information about the progress of the results directly.

i. Tie-Resolution Mechanism

The tie-resolution mechanism also provides added value in terms of process management. Under the manual mechanism, a tied result requires the committee to conduct another election and prepare the voting process again. In the developed system, a tie can be identified when an election round is closed, after which the committee is provided with an option to create a dedicated re-election round for the tied candidates. Thus, the 50% + 1 vote rule is not merely a written regulation but is translated into an integral part of the application's workflow.

j. Alignment between Requirements and Implementation

When compared with the requirements identified through interviews and observations, the implemented features cover the main requirements: voter validation, two types of voting rights, double-vote prevention, candidate and election-round management, result recapitulation, a real-time dashboard, and tied-result handling. This demonstrates alignment between the requirements analysis and system implementation. However, this functional alignment still needs to be supported by complete Black Box Testing results and user acceptance through UAT. Because the result table in the source thesis remains empty, this article explicitly distinguishes between features that have been implemented and quantitative testing results that are not yet available.

k. Research Limitations

The absence of advanced cryptography means that the current study should not be interpreted as providing end-to-end cryptographic guarantees of ballot secrecy, coercion resistance, or independently verifiable election integrity. The absence of large-scale load testing also means that the observed real-time behavior cannot be generalized to a large number of simultaneous voters and public dashboard clients. These limitations define the boundary between a functional prototype and a production-ready election platform.

The limitations of the research should be considered when the system is intended for use in an actual election forum. The system has not implemented blockchain or advanced cryptographic mechanisms; therefore, the study does not yet address cryptographic guarantees of vote integrity. Accordingly, deployment at real-world scale should be preceded by controlled performance testing using a participant count and concurrent-connection profile representative of the intended election. The results should report response time, throughput, error rate, WebSocket connection stability, and database behavior under peak load.

In addition, large-scale load testing has not been conducted to determine the capability of Laravel Reverb to handle a large number of simultaneous WebSocket connections. Therefore,

deployment at a real-world scale should initially be preceded by simulation and performance testing adjusted to the number of participants.

I. Technical Real-Time Workflow

Technically, the real-time workflow demonstrates a clear relationship between database transactions and WebSocket communication. Votes are stored within a transaction together with the update of voter status. Once the transaction is successfully completed, the VoteCasted event is broadcast through Reverb to a channel based on the election type. Laravel Echo on the browser receives the event and updates the total vote count and Chart.js visualization. This approach eliminates the need for periodic polling and enables result changes to be delivered to connected clients. When an election round is closed, the same event is used to distribute the final status, ensuring that the dashboard remains consistent with the database state.

4. CONCLUSION

The evaluation framework comprised 14 Black Box Testing scenarios and UAT with 20 respondents. The UAT score of 706/800 (88.25%) indicates a Very Good level of acceptance among the participating users. The study's principal contribution is the integrated implementation of DPD IMM Lampung's voting rules with real-time result broadcasting and automated tie-resolution in one election workflow. The findings support functional feasibility and user acceptance, but do not establish cryptographic security or large-scale scalability.

This study resulted in a Real-Time Counting Web Application for the Election of the General Chairperson of DPD IMM Lampung, developed using Laravel 12. The system includes voting-right validation, digital voting for two election stages, candidate and voter management, and result recapitulation. The Real-Time Counting feature was implemented using Laravel Reverb/WebSocket through broadcasting of the VoteCasted event, enabling the public dashboard to receive result updates without requiring a page refresh. The system also implements double-vote prevention through a unique constraint and provides a re-election mechanism for tied results in accordance with the 50% + 1 vote provision. Black Box Testing and User Acceptance Testing (UAT) are included as the evaluation procedures for the developed application. The Black Box scenarios cover the main system functions, while the UAT indicators assess usability, functional suitability, real-time result presentation, transparency, and overall user satisfaction. Before actual deployment, the system should undergo controlled pilot testing and large-scale performance testing rather than relying only on a development simulation. The administrator credentials should be changed before operational use, and future versions should add stronger authentication, cryptographic protection, security testing, and capacity evaluation.

Recommendations For DPD IMM Lampung, the system is recommended to undergo a small-scale simulation before being deployed for an actual Regional Deliberation (Musyawarah Daerah). The default administrator account credentials should also be changed immediately before the system is used in a real election. Further development is recommended to implement additional security mechanisms, such as two-factor authentication (2FA) for committee members/administrators and encryption of voting data; conduct performance testing (load testing or stress testing) to determine the capability of the Laravel Reverb WebSocket server to handle a large number of simultaneous connections and voting requests at a larger participant scale; develop a mobile application or Progressive Web App (PWA) version to provide voters with more flexible access to digital ballots; and add an automatic notification feature for voters who have not yet exercised their voting rights before the closing of an election round in order to increase participant engagement and voter participation.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to Universitas Muhammadiyah Kotabumi, particularly the Program Studi Sistem dan Teknologi Informasi, for providing academic support and resources throughout this research. The authors also extend their appreciation to the Regional Leadership Board of the Muhammadiyah Student Association (DPD IMM) Lampung and the organizing committee of the Regional Deliberation for providing the information, data, and cooperation required during the requirements analysis and system development process.

The authors also gratefully acknowledge all individuals who contributed directly or indirectly to the implementation, testing, and completion of this research. Their support, constructive feedback, and cooperation greatly contributed to the completion of this study.

REFERENCES

- Akbar, B. B. (2022). Analisis Kepuasan Pengguna Sistem Informasi Manajemen Surat Menggunakan Metode End User Computing Satisfaction. In *Jurnal Pengembangan Sistem Informasi dan Informatika* (Vol. 3, Number 1), <https://doi.org/10.47747/jpsii.v3i1.613>.
- Ansfridus, M., Dwiandiyanta, B. Y., & Emanuel, A. W. R. (2024). Black box testing dengan metode equivalence partitioning pada Techno Expertise Academy (TEA) Astra Credit Companies. *KONSTELASI: Konvergensi Teknologi Dan Sistem Informasi*, 4(2). <https://doi.org/10.24002/konstelasi.v4i2.8996>
- Apriyanti, A., & Ramli, M. (2025). Pengembangan website Program Studi Teknik Informatika berbasis Progressive Web App (PWA). *Jurnal Ilmu Pengetahuan Dan Teknologi (IPTEK)*, 9(1). <https://doi.org/10.31543/jii.v9i1.383>
- Ardiansyah, A., Rosid, M. A., & Fitriani, A. S. (2023). Web-based Muhammadiyah e-voting application using the Laravel framework. *Procedia of Engineering and Life Science*, 4. <https://doi.org/10.21070/pels.v4i0.1419>
- Bakhtiar, H. Y., Dwiasnati, S., & Herfandi. (2023). Implementasi dashboard reporting WebSocket dengan event regression test berbasis web menggunakan algoritma FIFO. *Format: Jurnal Ilmiah Teknik Informatika*, 12(1), 71–80. <https://doi.org/10.22441/format.2023.v12.i1.009>
- Dachi, J. V., & Suhada, J. (2025). Implementasi basis data sederhana menggunakan MySQL/PostgreSQL. *Router: Jurnal Teknik Informatika Dan Terapan*, 3(1), 49–59. <https://doi.org/10.62951/router.v3i2.621>
- Emami Neishabouri, A., Yajam, H., Akhaee, M. A., & Asghari, R. (2023). A scalable decentralized privacy-preserving e-voting system based on zero-knowledge off-chain computations. *Journal of Information Security and Applications*, 79, 103645. <https://doi.org/10.1016/j.jisa.2023.103645>
- Fahmi, M., Manajemen, A., Keimigrasian, T., & Imigrasi, P. (2024). Analisis Implementasi Sistem Informasi: Studi Literatur Analysis Of Information System Implementation: Literature Review. *JTSI*, 5(1), 264–275. <https://doi.org/https://doi.org/10.35957/jtsi.v5i1.7815>
- Kadim, A. A., Hadjaratie, L., & Muthia. (2023). Implementasi framework Laravel dalam pembuatan sistem pencatatan notula berbasis website. *Jurnal Sistem Informasi Bisnis*, 13(1), 45–51. <https://doi.org/10.21456/vol13iss1pp45-51>
- Kho, Y.-X., Heng, S.-H., & Chin, J.-J. (2022). A review of cryptographic electronic voting. *Symmetry*, 14(5), 858. <https://doi.org/10.3390/sym14050858>
- Novita, D., & Helena, F. (2021). Analisis kepuasan pengguna aplikasi Traveloka menggunakan metode Technology Acceptance Model (TAM) dan End-User Computing Satisfaction (EUCS). *Jurnal Teknologi Sistem Informasi*, 2(1), 22–37. <https://doi.org/10.35957/jtsi.v2i1.846>
- Nurdiansyah, R., Suryawan, S. H., & Rahim, A. (2024). Pembuatan sistem aplikasi electronic voting (e-voting) berbasis web dengan pendekatan metode Waterfall. *Jurnal Informatika Terpadu*, 10(2), 112–121. <https://doi.org/10.54914/jit.v10i2.1283>
- Paskalis, Y. M. J. G., & Bachri, K. O. (2024). Student Organization Website with E-Voting Feature by Using Student Card Verification Concept Design. *Sinkron*, 8(3), 1505–1514. <https://doi.org/10.33395/sinkron.v8i3.13734>
- Pereira, B. M. B., Torres, J. M., Sobral, P. M., Moreira, R. S., Soares, C. P. de A., & Pereira, I. (2023). Blockchain-Based Electronic Voting: A Secure and Transparent Solution. *Cryptography*, 7(2). <https://doi.org/10.3390/cryptography7020027>
- Rosmiati. (2021). Analisis dan pengujian sistem menggunakan Black Box Testing Equivalence Partitioning. *Jurnal Sains Komputer Dan Teknologi Informasi*, 3(2), 56–63. <https://doi.org/10.33084/jsakti.v3i2.1932>
- Setiaji, & Sastra, R. (2021). Implementasi diagram UML (Unified Modelling Language) pada perancangan sistem informasi penggajian. *Jurnal Teknik Komputer*, 7(1), 106–111. <https://doi.org/10.31294/jtk.v7i1.9773>
- Sika, X., Pratama, Y., Riyadi, W., Kisbianty, D., & Zulia, R. (2024). Perancangan sistem e-voting Pemilu Raya Mahasiswa di Universitas Dinamika Bangsa. *Jurnal Ilmiah Media Sisfo*, 18(2), 291–303. <https://doi.org/10.33998/mediasisfo.2024.18.2.1990>
- Siking, A., Koniyo, M. H., & Yassin, R. M. T. (2023). Unified Modelling Language (UML) dalam perancangan sistem informasi pelayanan pengujian material berbasis web pada Dinas Pekerjaan Umum dan Penataan Ruang Provinsi Gorontalo. *Diffusion: Journal of Systems and Information Technology*, 3(2), 204–213. <https://doi.org/10.37031/diffusion.v3i2.20543>
- Sinlae, F., Irwanda, E., Maulana, Z., & Eka Syahputra, V. (2024). Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP. *Jurnal Siber Multi Disiplin*, 2(2), 119–132. <https://doi.org/10.38035/jsmd.v2i2.186>

- Vladucu, M. V., Dong, Z., Medina, J., & Rojas-Cessa, R. (2023). E-Voting Meets Blockchain: A Survey. *IEEE Access*, 11, 23293–23308. <https://doi.org/10.1109/ACCESS.2023.3253682>
- Wulandari, A. S., Saepudin, A., Kinanti, M. P., Sudesi, Z., Saifudin, A., & Yulianti. (2022). Pengujian aplikasi sistem informasi akademik berbasis web menggunakan metode Black Box Testing Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 5(2), 102–109. <https://doi.org/10.32493/jtsi.v5i2.17561>