International Journal of Applied Science and Technology Application

Vol. 1, No. 2, September 2026, pp. 80-91
e-ISSN: 3124-9132
DOI: https://doi.org/10.62712/ijapset.v1i2.11

Automation of Teaching Quality Survey Processing for Lecturers via Integrated TOPSIS Calculation

Ade Rizka[1]*, Virdyra Tasril[2], Indri Sulistianingsih[3], Fristi Riandari[4], Ramadhanu Ginting[5]

[1,2,3]Software Engineering Technology Study Program, Department of Computer Engineering and Informatics, Politeknik Negeri Medan, Medan, Indonesia
[4,5]Informatics Management Study Program, Department of Computer Engineering and Informatics, Politeknik Negeri Medan, Medan, Indonesia
Author's Email: [1*]aderizka@polmed.ac.id, [2]virdyratasril@polmed.ac.id, [3]indrisulistianingsih@polmed.ac.id, [4]fristiriandari@polmed.ac.id, [5]ramadhanuginting@polmed.ac.id

Article Info

Article history:

Received: July 31, 2026
Accepted: August 30, 2026
Published: September 30, 2026

Keywords: *lecturer competency evaluation; data validation; prototype model; black-box testing; System Usability Scale*

ABSTRACT

Faculty teaching quality in a vocational study program was previously monitored manually, which risked calculation errors and failed to prevent duplicate evaluations. This research presents a prototype application that integrates student evaluations, data validation, and calculations based on the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) into a single system. The development followed a prototyping model and used four faculty competencies as evaluation criteria. The system enforces rules requiring one evaluation per faculty member, at least one entry for each competency, and a minimum number of evaluators. Black-box testing across ten scenarios confirmed that all functions operate as intended. System-generated calculations matched manual calculations up to four decimal places. An implementation involving 485 evaluations provided by 100 students regarding six faculty members revealed that "professional competence" received the lowest scores across all faculty. A System Usability Scale assessment yielded an average score of 94.42 of 13 respondents. This system enables study programs to process faculty evaluations quickly and accurately, providing a foundation for professional development training.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Corresponding Author:

Ade Rizka
Department of Computer Engineering and Informatics, Politeknik Negeri Medan
Medan, Indonesia
Email: aderizka@polmed.ac.id

1. INTRODUCTION

The quality of learning in higher education is strongly influenced by lecturers' competence. According to Decree No. 399/P/2021 of the Minister of Education, Culture, Research, and Technology [1], lecturer competence comprises four aspects: pedagogical, professional, personal, and social. Students interact most frequently with lecturers during the learning process; therefore, student assessments are widely used to evaluate teaching. These evaluations can reveal a lecturer's strengths and weaknesses, enabling study programs to determine appropriate professional development or support.

In a previous study [2], the Diploma IV (D4) Software Engineering Technology (TRPL) study program at the Politeknik Negeri Medan evaluated lecturer competence using the TOPSIS method. That study relied on simulated data, and all calculations were performed manually. Wilantara, Wasilah, and Triloka (2024) observed

a similar situation, where they calculated teaching evaluations manually in Excel without applying weights to the assessment criteria [3]. Processing data via spreadsheets carries risks of errors and data loss risks that users often fail to recognize [4]. Furthermore, manual processing cannot prevent duplicate assessments or ensure that enough students evaluate each lecturer. This research addresses how to automate the processing of lecturer competence evaluations while ensuring data validation and accurate computational results.

Information technology can streamline lecturer evaluation processing, making the procedure faster and more efficient [5]. While collecting assessments online can expedite the process [6], the outcome depends on student participation. Dzakadzie and Quansah (2023) found that only 48.4% of expected responses were collected; as participation decreases, the reliability of the evaluation results also declines [7]. Consequently, a minimum threshold for the number of students required to evaluate each lecturer is necessary [8]. To generate rankings, the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method compares each alternative against both a positive ideal solution and a negative ideal solution [9]. This method is increasingly used in education [10], for instance, to rank lecturers' teaching effectiveness at vocational higher education institutions [11] and to assess lecturer performance using an extended fuzzy logic approach [12].

In a study by Adigun et al. (2022), the authors noted that student evaluations of lecturers are often questionable because students may be forced or may fabricate responses [13]. The proposed solution was a web-based system that 60 students tested by evaluating six lecturers. The results indicated that the system is viable and can distinguish between fabricated and genuine evaluations. In a study by Bata (2025), users without a technical background struggled to apply the TOPSIS method; consequently, they developed a web-based TOPSIS tool to address various use cases [14]. The tool's rankings matched those from manual calculations, and it achieved a System Usability Scale (SUS) score of 76.5, placing it in the "good" category. However, Adigun et al.'s system does not use a ranking method, while Bata's tool is designed for general purposes and does not process lecturer evaluation questionnaires.

In Do, Tran, and Tran's (2024) study, assessing faculty performance requires objective weighting of criteria [15]. The study uses the Fuzzy Analytic Hierarchy Process (AHP) to determine criterion weights and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) to rank faculty members based on expert evaluations. Consequently, the faculty evaluation process becomes more objective, helping higher education leadership enhance educational quality. In Hayati, Syaputra, and Eirlangga (2023), the evaluation team faced challenges processing large datasets of faculty information to identify top-performing instructors; therefore, they used the TOPSIS method [16]. As a result, the study selected three faculty members for recognition, identifying research, teaching, and attendance as the most influential criteria. Both studies relied on expert or institutional data rather than student evaluations.

In Yan and Li (2025), the reliability of faculty teaching ability scores must be verified before using them for ranking [17]. The study analyzed evaluations from 568 students across 16 classes taught by 16 faculty members. The findings indicate that a minimum of 30 student evaluators is required to ensure the reliability of the assessment scores. Dzakadzie (2026) similarly recommends approximately 13 to 25 student evaluators for smaller class sizes [8]. However, these recommendations regarding the number of evaluators have not yet been formalized as a standard rule within faculty evaluation processing systems.

While existing research suggests that web-based systems and the TOPSIS method can facilitate faculty evaluation, previous studies have typically addressed only specific segments of the evaluation process. This research's novelty lies in integrating several components student evaluation collection, data validation, a transparent TOPSIS calculation process (with verifiable stages), and accuracy testing into a single, unified system; this integrated approach has not been observed in the reviewed literature. In this study, we developed a prototype application to process faculty competency evaluations for the D4 Software Engineering Technology (TRPL) study program at Politeknik Negeri Medan, covering the entire workflow from questionnaire completion to presenting rankings and graphical data. The system implements rules that allow one assessment per lecturer, require at least one statement per competency, and enable configuration of a minimum number of evaluators. The prototype was validated through black-box testing, comparison of system-generated results with manual calculations, and a usability assessment using the System Usability Scale (SUS). The system is intended to help academic departments process lecturer evaluations quickly and accurately, supporting professional development aimed at improving instructional quality.

2. RESEARCH METHOD

This study employs a prototype development model. The prototype model was selected because it allows rapid system development, user testing, and subsequent refinement based on feedback [18]. The research process consists of requirements analysis, system design, prototype construction, testing, and refinement. The construction and testing phases can be iterated until the system meets the users' requirements. Figure 1 illustrates the complete research process.

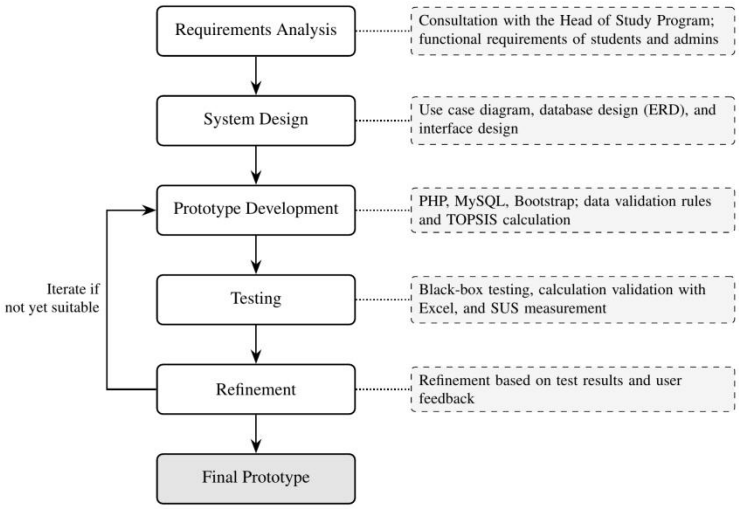


Figure 1. Research stages.

2.1. Requirements Analysis

The team conducted requirements analysis in consultation with the Head of the Study Program. The system has two user types: students and the study program administrator. Students provide their personal information and evaluate the lecturers who taught them during the current semester. The administrator manages lecturer data, criteria weights, questionnaire items, and questionnaire results. The administrator also performs TOPSIS calculations and views the ranking results, including their corresponding graphs. Table 1 presents the system's functional requirements.

Table 1. System functional requirements.

Actor	Functional Requirement
Student	Enter personal details (student ID number, name, class, semester, study program, department).
Student	Select a lecturer from the list and complete 40 questionnaire items.
Student	View the list of lecturers who have been evaluated and those who have not.
Admin	Log in to the system and change the password.
Admin	Manage lecturer data, criteria weights, and questionnaire items.
Admin	Import questionnaire data from a Google Form CSV file.
Admin	Review, accept, or delete flagged responses.
Admin	View each stage of the TOPSIS calculation, rankings, and charts, and download the results.

2.2. Research Instruments and Data

The research instrument consists of a questionnaire containing 40 statements covering four lecturer competencies [1]. Each competency comprises 10 statements, developed based on the specific details of the lecturer competencies. This instrument improves on a previous version [2] by refining items that measured the same behaviors across two competencies. Students select the statements that apply to the lecturer being evaluated, with a requirement to choose at least one statement for each competency. Each selected statement is assigned a value of 0.5, resulting in a score range of 0.5 to 5 for each competency. We collected data from approximately 100 students in the D4 TRPL (Software Engineering Technology) study program, who evaluated six lecturers via a dedicated application and a Google Form.

A student's competency score for a given lecturer is calculated using Equation (1). The values in the decision matrix represent the average scores from all students who evaluated that particular lecturer, as shown in Equation (2).

$$S_{mij} = 0{,}5 \times c_{mij} \tag{1}$$

$$x_{ij} = \frac{1}{n_i} \sum_{m=1}^{n_i} S_{mij} \tag{2}$$

In the equation, c represents the number of items checked by student m for lecturer i regarding competency j, while n is the number of students who evaluated lecturer i. The criteria and weights

employed are presented in Table 2. We determined these weights through discussions with the Head of the Study Program and then normalized them so their sum equals 1. All criteria are of the "benefit" type, meaning that higher values indicate better performance.

Table 2. Criteria and weighting.

Code	Competency	Number of items	Weight	Normalized weight
C1	Pedagogical	10	4	0.2500
C2	Professional	10	5	0.3125
C3	Personality	10	4	0.2500
C4	Social	10	3	0.1875

2.3. System Design

The system design includes process, database, and interface design. The use case diagram in Figure 2 illustrates the interactions between students, the administrator, and the system. The database comprises seven tables: admin, student, lecturer, criteria, item, response, and response_detail, as shown in Figure 3. The response table stores a single evaluation provided by a student for a lecturer, while the response_detail table stores the checked status of each item within that evaluation. The system is built with PHP, MySQL, and Bootstrap and runs on a local XAMPP server.

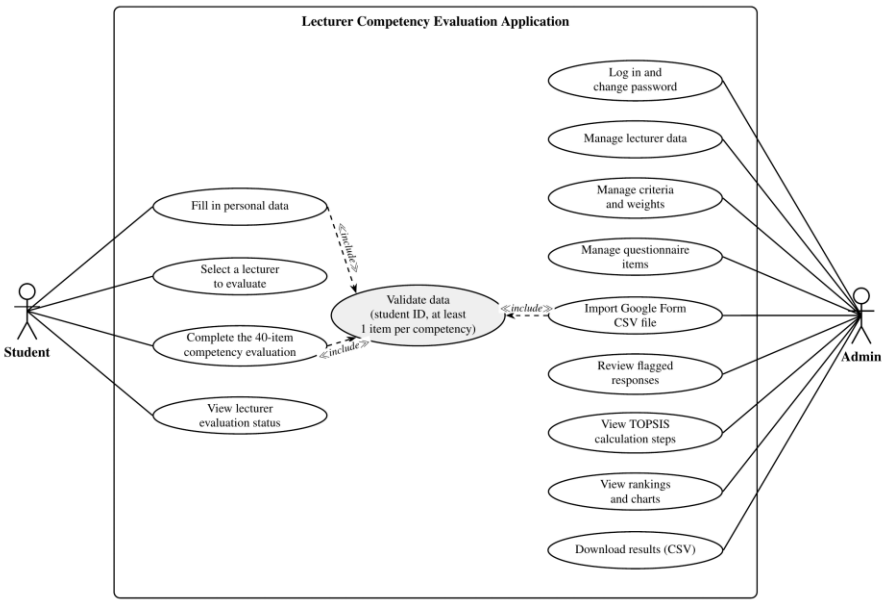


Figure 2. Use case diagram.

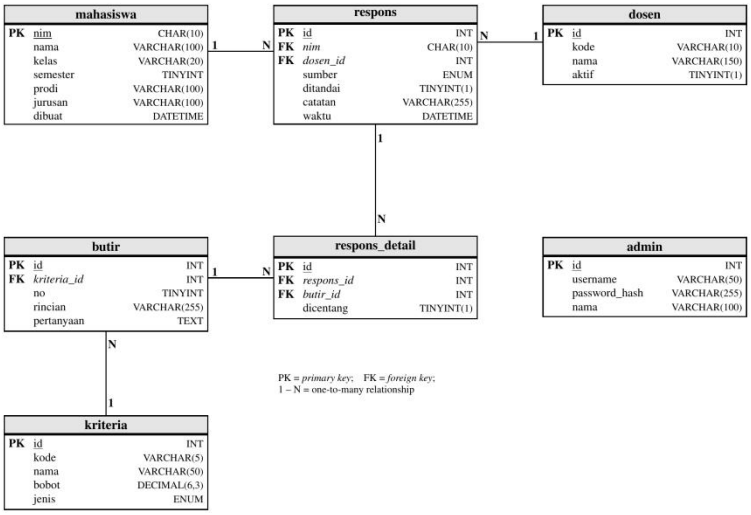


Figure 3. Database design (ERD).

2.4. Data Validation Rules

The system implements validation rules to ensure reliable processed data, since students are the primary source of inconsistency in teaching evaluations [19]. First, the student ID number (NIM) must consist of 10 digits. Second, a single student ID may only be used to evaluate a specific lecturer once. Third, at least one item must be selected for each competency. Fourth, lecturers who receive fewer than the minimum required student evaluations are excluded from the TOPSIS calculation and reported separately [8]. Imported responses that fail to meet these rules will be flagged and only included after being reviewed by an administrator.

2.5. TOPSIS Calculation

The system's TOPSIS calculation follows six stages [9]. The first stage involves constructing a decision matrix using equation (2). The second stage normalizes the matrix using equation (3). The third stage entails calculating the weighted matrix using equation (4). The fourth stage involves determining the positive and negative ideal solutions using equations (5) and (6). The fifth and sixth stages calculate each lecturer's distance from the ideal solutions using equations (7) and (8), then compute the preference value using equation (9).

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \quad (3)$$

$$y_{ij} = w_j \, r_{ij} \quad (4)$$

$$A^+ = \left(\max_i y_{i1}, \max_i y_{i2}, \ldots, \max_i y_{in}\right) \quad (5)$$

$$A^- = \left(\min_i y_{i1}, \min_i y_{i2}, \ldots, \min_i y_{in}\right) \quad (6)$$

$$D_i^+ = \sqrt{\sum_{j=1}^{n} \left(y_{ij} - y_j^+\right)^2} \quad (7)$$

$$D_i^- = \sqrt{\sum_{j=1}^{n} \left(y_{ij} - y_j^-\right)^2} \quad (8)$$

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad (9)$$

The lecturer with the highest V value is ranked first. The system displays the results of each stage, allowing the administrator to review them. The system can also download the results from each stage as a CSV file. These files are used to compare the system's output with manual calculations during testing.

2.6. System Testing

System testing is conducted in three ways. The first approach is functional testing using the black-box method, incorporating equivalence partitioning and boundary value analysis techniques [20], [21]. Test scenarios are developed for each function covering both valid and invalid inputs as shown in Table 3. The second approach validates calculation accuracy by comparing the system's results with manual calculations performed in Microsoft Excel. The data used for this purpose include data from a previous study [2] and data from the questionnaires in this study. We consider the results accurate if the values at every stage of the calculation as well as the final rankings match those from the manual calculations.

Table 3. Black-box testing scenarios.

No.	Function	Scenario	Expected Result
1	Personal data	Student ID (NIM) has fewer or more than 10 digits.	Rejected with an error message.

No.	Function	Scenario	Expected Result
2	Personal data	Student ID (NIM) has 10 digits, and data is complete.	Added to the lecturer list.
3	Personal data	Re-entering an already registered Student ID (NIM).	Uses the previously stored name.
4	Evaluation	One competency left unchecked.	Rejected; the empty competency is highlighted.
5	Evaluation	At least one competency is checked.	Evaluation is saved.
6	Evaluation	Evaluating the same lecturer a second time.	Rejected; lecturer is marked as already evaluated.
7	Admin login	Incorrect password.	Rejected with an error message.
8	CSV import	Row contains an invalid Student ID (NIM) or an unknown lecturer.	Row fails and is reported.
9	CSV import	Response with one or more competencies left unchecked.	Response is imported and marked accordingly.
10	TOPSIS results	Lecturer has fewer evaluators than the minimum threshold.	Excluded from calculation; displayed separately.

The third method involves measuring the system's ease of use using the System Usability Scale (SUS) [22]. The SUS consists of 10 statements, each rated on a scale from 1 to 5. The SUS questionnaire is completed by 10 students and 3 administrators after they have finished assigned tasks within the system. The SUS score is calculated using equation (10), where s represents the respondent's answer to the k statement. The final score is the average of all respondents' scores, ranging from 0 to 100.

$$SUS = 2{,}5 \times \left[\sum_{k\ ganjil}(s_k - 1) + \sum_{k\ genap}(5 - s_k)\right] \quad (10)$$

3. RESULTS AND DISCUSSION

3.1. System Implementation

The prototype consists of two sections: a student interface and an admin interface. The system's design incorporates Politeknik Negeri Medan's colors and logo to ensure easy recognition by users. The student interface is used to enter personal information and evaluate lecturers. In contrast, the admin interface manages data, reviews responses, and displays the results of the TOPSIS calculations. Each page is described below.

The personal information page is the first screen students encounter, as shown in Figure 4. Students must enter their student ID number (NIM), name, class, semester, study program, and department. The system will reject any NIM that does not consist of exactly 10 digits. If a NIM is already registered, the system retrieves the associated name to ensure that a single NIM cannot be linked to different names. After validating the personal information, the system directs the student to the lecturer list page.

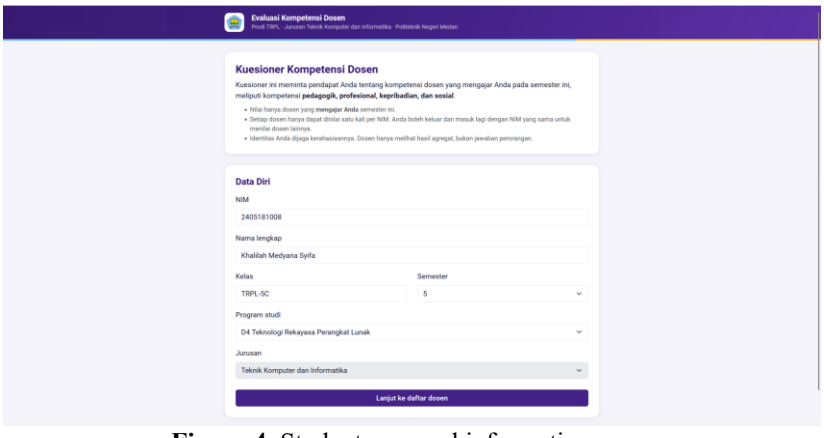


Figure 4. Student personal information page.

The lecturer list page displays all lecturers eligible for evaluation and their evaluation status, as shown in Figure 5. Lecturers who have already been evaluated are marked as such and cannot be evaluated again. Students may log out and log back in using the same NIM to continue evaluating other lecturers.

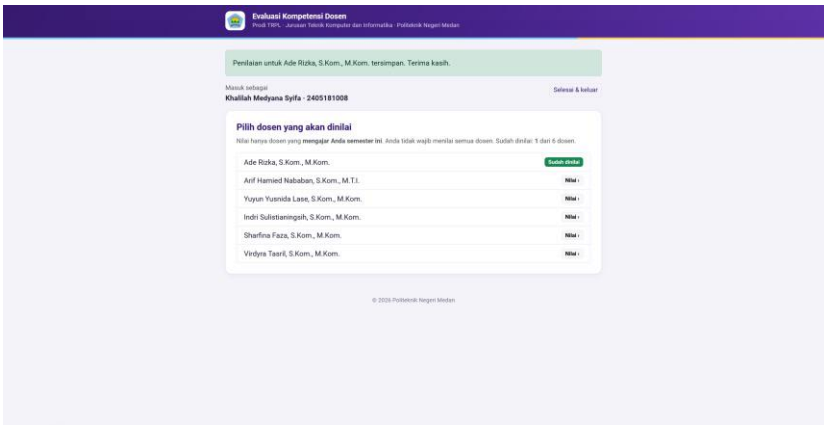


Figure 6. Lecturer list page.

The evaluation page presents 40 statements categorized into four competencies, as shown in Figure 6. For each competency, the system displays the number of selected items. If any competency remains incomplete, the system rejects the submission and highlights the incomplete section in red. Before saving the evaluation, the system requests confirmation, as submitted evaluations cannot be modified.

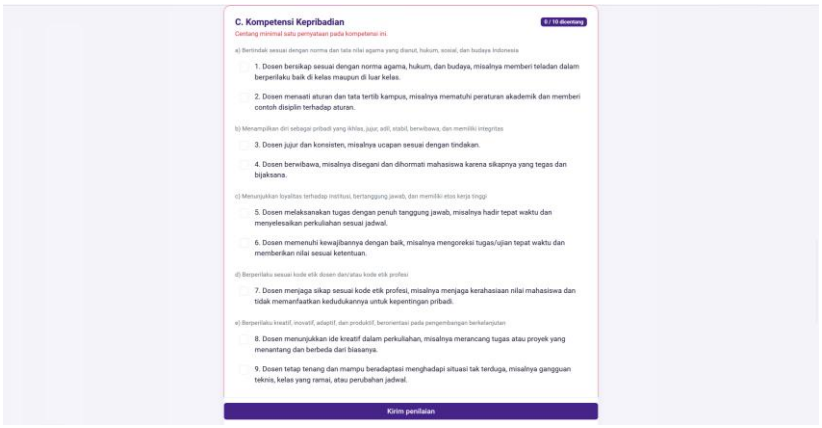


Figure 6. Lecturer evaluation page with competency validation.

The admin dashboard displays a summary, including the number of valid responses, the total number of students, flagged responses, and data sources, as shown in Figure 7. The dashboard also shows the number of evaluators for each lecturer and how many more are needed to meet the minimum threshold. This information helps the admin identify lecturers who do not yet meet the requirements for inclusion in the calculations.

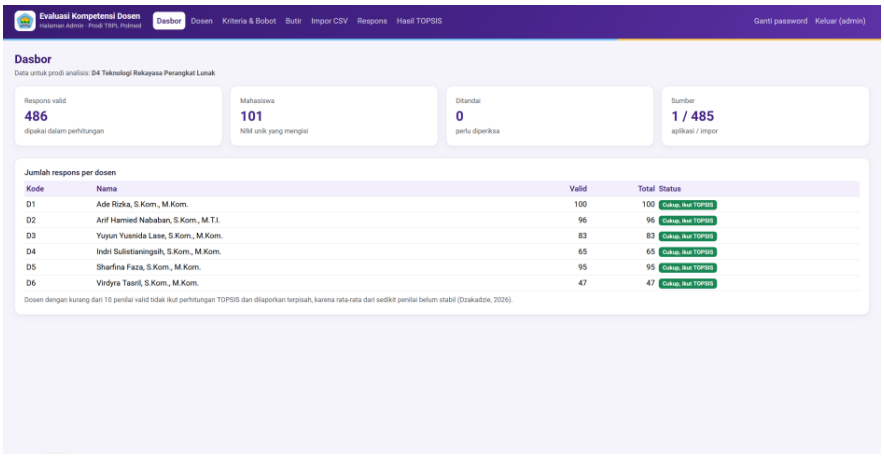


Figure 7. Admin dashboard page.

The import page is used to upload questionnaire data from a Google Forms CSV file, as shown in Figure 8. Admins can run a test mode first without saving the data. The system reports how many rows were

successfully processed, failed, or flagged, along with the reasons. Flagged responses are excluded from calculations until they are reviewed and accepted by the admin.

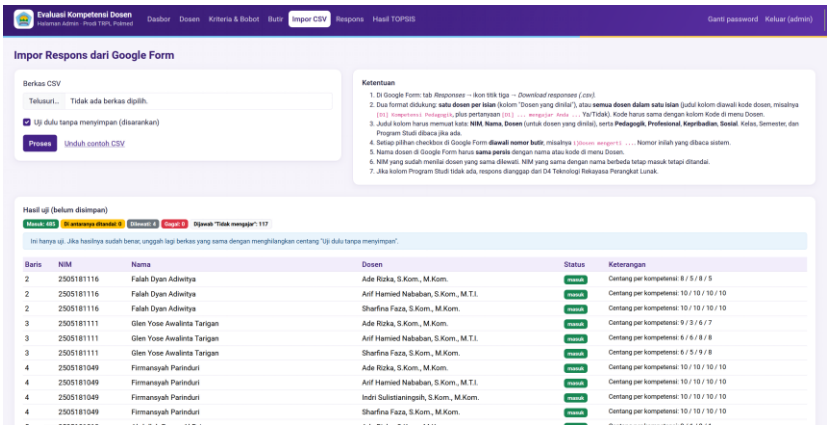


Figure 8. Questionnaire data import page.

The results page displays each stage of the TOPSIS calculation from the decision matrix to the final preference values and lecturer rankings as shown in Figure 9. The results are also presented as charts, including a graph of preference values and a breakdown of scores for each competency. Lecturers who have not met the minimum number of evaluators are listed in a separate table. All calculation results can be downloaded as a CSV file.

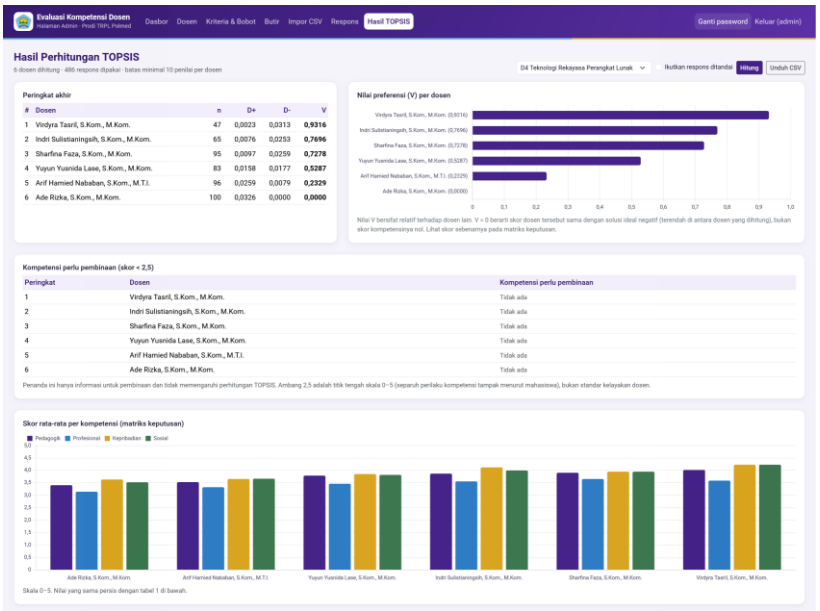


Figure 8. TOPSIS calculation results page.

3.2. Black-Box Testing Results

We conducted black-box testing based on the 10 scenarios presented in Table 3. Table 4 shows the test results.

Table 4. Black-box test results.

No.	Scenario	Expected Result	**Test Result**	**Conclusion**
1	Student ID (NIM) has fewer or more than 10 digits.	Rejected with an error message.	NIMs with 9 digits, 11 digits, or containing letters are rejected with the message "NIM must be a 10-digit number".	Successful
2	Student ID (NIM) has 10 digits, and data is complete.	Added to the lecturer list.	Data accepted; user redirected to the lecturer list page.	Successful
3	Re-entering an already registered Student ID (NIM).	Uses the previously stored name.	System uses the name stored previously.	Successful

4	One competency left unchecked.	Rejected; the empty competency is highlighted.	Submission rejected; the empty competency is outlined in red with a mandatory-field message.	Successful
5	At least one competency is checked.	Evaluation is saved.	Assessment saved; lecturer status updated to "assessed".	Successful
6	Evaluating the same lecturer a second time.	Rejected; lecturer is marked as already evaluated.	Rejected with the message "You have already assessed this lecturer".	Successful
7	Incorrect password.	Rejected with an error message.	Rejected with the message "Incorrect username or password".	Successful
8	Row contains an invalid Student ID (NIM) or an unknown lecturer.	Row fails and is reported.	Row is not saved and is included in the import report along with the reason.	Successful
9	Response with one or more competencies left unchecked.	Response is imported and marked accordingly.	Entry saved with a flagged status and excluded from calculations.	Successful
10	Lecturer has fewer evaluators than the minimum threshold.	Excluded from calculation; displayed separately.	Lecturer appears in the "insufficient data" table and is excluded from TOPSIS calculations.	Successful

All 10 scenarios yielded results consistent with expectations. These results show that the data validation rules outlined in subsection 2.4 function within the system for both application-entered and imported data. The system can also reject invalid inputs and provide clear feedback to the user. Consequently, the system includes only data that complies with the validation rules in the TOPSIS calculations.

3.3. Validation of Calculation Accuracy

The first validation used data from a previous study [2]. The system-generated preference values and rankings matched the results reported in that study. The second validation used the current study's questionnaire data by independently recalculating the TOPSIS stages, starting from the decision matrix produced by the system. The distance values ($D+$ and $D-$) and preference values (V) obtained from the recalculation matched the system's output to four decimal places, with a maximum discrepancy of 0.0001 due to rounding input values. The rankings derived from both sets of calculations were also identical. These results indicate that the system's TOPSIS calculations are accurate.

3.4. Application to Questionnaire Data

Questionnaire data imported from Google Forms yielded 485 valid responses, with no flagged entries or failed rows. We excluded 117 entries because students indicated the lecturer did not teach their class, and we omitted 4 entries because they were duplicate evaluations for the same lecturer. Including one response submitted via the application, we used 486 responses to evaluate six lecturers. Each lecturer received more than the minimum number of evaluations, ranging from 47 to 100 students. Table 5 presents the results of the TOPSIS calculations. Lecturer names have been anonymized using codes A1 through A6.

Table 5. TOPSIS calculation results for questionnaire data.

Code	n	C1	C2	C3	C4	$D+$	$D-$	V	Rank
A1	100	3,4000	3,1400	3,6300	3,5200	0,0326	0,0000	0,0000	6
A2	96	3,5208	3,3177	3,6510	3,6615	0,0259	0,0079	0,2329	5
A3	83	3,7831	3,4578	3,8494	3,8133	0,0158	0,0177	0,5287	4
A4	65	3,8615	3,5538	4,1154	3,9923	0,0076	0,0253	0,7696	2
A5	95	3,9000	3,6474	3,9421	3,9421	0,0097	0,0259	0,7278	3
A6	47	4,0106	3,5851	4,2234	4,2234	0,0023	0,0313	0,9316	1

The average scores for all lecturers range from 3.14 to 4.22 on a 5-point scale; thus, no competency falls below the 2.5 threshold that would trigger a need for professional development. Professional competency (C2) received the lowest scores across all lecturers, while personality (C3) and social (C4) competencies generally received the highest. These findings can guide the study program in prioritizing development efforts focused on professional competency. Lecturer A6 ranked first with a V-value of 0.9316, while lecturer A1 had a V-

value of 0. A V-value of 0 does not mean the lecturer's competency score is zero; rather, it indicates that their score was the lowest across all four competencies, aligning them with the negative ideal solution.

Although the score differences between lecturers are relatively small less than 0.7 points for any given competency the V-values span from 0 to 0.9316. This is because TOPSIS measures each lecturer's relative position against both the positive and negative ideal solutions. Consequently, interpret V-values in conjunction with the average scores for each competency to avoid misconstruing them as absolute performance metrics. Rankings that are close to one another such as those of A4 and A5 should also be interpreted with caution, as the number and composition of students evaluating each lecturer differ. The stability of these rankings warrants further investigation in future research.

3.5. Usability Testing Results

The System Usability Scale (SUS) assessment was conducted on October 16, 2026, using the developed prototype. The respondents included 10 students and 3 administrators from the D4 TRPL (Software Engineering Technology) study program. Before completing the questionnaire, the students performed three tasks: entering their personal information, evaluating a lecturer until the assessment was saved, and logging out and logging back in using the same student ID (NIM) to evaluate another lecturer. The administrator completed five tasks: accessing the admin page, modifying one lecturer's data, importing a CSV file in test mode, opening the TOPSIS results page, and downloading the CSV file containing the calculation results. All respondents completed every task; therefore, we used 13 responses in the calculations. Table 6 presents the SUS score calculation results, derived using equation (10).

Table 6. SUS score results.

Group	n	Mean	Standard Deviation	Minimum	Maximum
Students	10	93.00	4.22	87.50	100.00
Admin	3	99.17	1.44	97.50	100.00
All respondents	13	94.42	4.58	87.50	100.00

The average SUS score across all respondents is 94.42. This score exceeds the SUS benchmark of 68 [23] and falls into the A+ category on the Sauro and Lewis rating scale used by Suria (2024) [24]. The lowest recorded score was 87.50, meaning every respondent scored above that average benchmark. Although administrators' scores were higher than students', the small number of administrators (only three) makes it impossible to draw definitive conclusions about this difference. These scores are also higher than those reported for other web-based systems: a TOPSIS-based tool in a study by Bata (2025) with a score of 76.5 [14], a web-based learning platform with a score of 75 [25], and a web-based lecturer assessment application with a score of 82.5 [5]. However, as these systems involve different users and tasks, this comparison serves only as a general reference.

Table 7. Average responses and score contributions for each SUS statement (n = 13).

No.	Statement	Average Response	Score Contribution (0–4)
1	I would use the application frequently	4.69	3.69
2	The application is too complicated	1.08	3.92
3	The application is easy to use	4.77	3.77
4	I need help from others	1.08	3.92
5	The features are well-integrated	4.69	3.69
6	There are many inconsistencies	1.23	3.77
7	Most people would learn to use it quickly	4.77	3.77
8	The application is cumbersome	1.23	3.77
9	I feel confident when using it	4.69	3.69

The score contribution is calculated by subtracting 1 from the average response for odd-numbered statements and subtracting the average response from 5 for even-numbered statements, as shown in Table 7. The highest contributions are in statements 2 and 4 (3.92), indicating that respondents do not find the application complicated and do not need assistance to use it. The lowest contributions are found in statements 1, 5, and 9, with a value of 3.69. Among students, these three statements yielded a contribution of 3.6, suggesting that feature coherence and user self-confidence could still be improved. The low score for statement 1 may also reflect that lecturer evaluation is conducted only once per semester.

Respondents also suggested improvements to the application. Of the 13 respondents, six offered positive feedback or made no suggestions. Three respondents suggested making the application available to all lecturers and other study programs. Two respondents suggested adding photos or biographical information for the

lecturers, one respondent suggested making the application accessible online, and one respondent suggested improving the interface for mobile devices. These suggestions align with the system's current state, which runs on a local server and is restricted to the D4 TRPL (Software Engineering Technology) study program. Consequently, these suggestions will serve as a basis for improvements in the next development phase.

This SUS (System Usability Scale) evaluation has certain limitations. With only 13 respondents all from a single study program the results reflect usability for this initial user group. A larger sample size should validate the high and relatively uniform scores once the system is deployed online. Furthermore, any interface changes made during the next development phase should be re-evaluated using the SUS.

4. CONCLUSION

This research produced an application prototype for processing lecturer competency evaluations in the D4 TRPL study program at Politeknik Negeri Medan, covering the entire process from questionnaire completion to presenting lecturer rankings. The system implements data validation rules: a 10-digit student ID (NIM), one evaluation per lecturer, at least one item per competency, and a minimum number of evaluators. Black-box testing confirms that all 10 test scenarios yielded the expected results. The system's TOPSIS calculations align with manual recalculations based on both previous research data and the current study's questionnaire data, in terms of both preference values and final rankings. An analysis of 486 student responses on six lecturers revealed that "professional competency" received the lowest score, making it a priority area for professional development. Usability testing, measured via the System Usability Scale (SUS), yielded an average score of 94.42 from 13 respondents, falling into the "excellent" category.

This study has certain limitations: the system was hosted on a local server and tested by respondents from a single academic program. Future research should focus on deploying the system online and optimizing its interface for mobile devices. The system could also be expanded to other academic programs and enhanced with lecturer profiles, as users suggested. Furthermore, SUS testing should be repeated with a larger sample size once the system is deployed online.

5. ACKNOWLEDGEMENTS

The authors thank the Head of the D4 Software Engineering Technology program, the program's administrative staff, and the students who participated as respondents in this study. The authors also thank the Center for Research and Community Service (P3M) at Politeknik Negeri Medan for their support. This research was self-funded through the Independent Regular Research (PRM) scheme.

6. REFERENCES

[1] Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. (2021). Keputusan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi Nomor 399/P/2021 tentang Standar Pelayanan Minimum POLMED. Jakarta: Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi.

[2] A. Rizka *et al.*, "Analisis Metode TOPSIS Terhadap Kinerja Dosen Berdasarkan Preferensi Penilaian Mahasiswa," *Jurnal Nasional Teknologi Komputer*, vol. 5, no. 3, pp. 974–984, Oct. 2025.

[3] R. Pramodiyo Wilantara, Wasilah, and J. Triloka, "Sistem Pendukung Keputusan Penilaian Efektivitas Pemilihan Dosen Terbaik ITBA Dian Cipta Cendikia Bandar Lampung Menggunakan Metode AHP dan Topsis," *Jurnal JUPITER*, vol. 16, no. 2, pp. 425–435, Oct. 2024.

[4] L. Raković, M. Sakal, S. Dakić, and J. Đurković, "Spreadsheets: risk from the shadow," *Int. J. Bus. Inf. Syst.*, vol. 41, no. 1, pp. 1–19, 2022, doi: 10.1504/IJBIS.2022.126027.

[5] S. Aryza and Z. Lubis, "Optimizing Lecturer Performance Assessment through Web-Based Application Design and Deployment," *Journal of Computer Science, Information Technology and Telecommunication Engineering (JCoSITTE)*, vol. 7, no. 1, pp. 1028–1035, Mar. 2026, doi: 10.30596/jcositte.v7i1.26501.

[6] H. C. Wang, R. Ehianeta, and H. Sen Doong, "Students' online evaluation of teaching and system continuance usage intention: new directions from a multidisciplinary perspective," *Assess. Eval. High. Educ.*, vol. 48, no. 8, pp. 1295–1309, Nov. 2023, doi: 10.1080/02602938.2023.2199181.

[7] Y. Dzakadzie and F. Quansah, "Modeling unit non-response and validity of online teaching evaluation in higher education using generalizability theory approach," *Front. Psychol.*, vol. 14, 2023, doi: 10.3389/fpsyg.2023.1202896.

[8] Y. Dzakadzie, "Identifying optimal number of raters and items for dependable student evaluation of teaching: evidence from generalizability theory," *Front. Educ. (Lausanne).*, vol. 11, pp. 01–14, May 2026, doi: 10.3389/feduc.2026.1814494.

[9] V. Pandey, Komal, and H. Dincer, "A review on TOPSIS method and its extensions for different applications with recent development," *Soft comput.*, vol. 27, no. 23, pp. 18011–18039, Dec. 2023, doi: 10.1007/S00500-023-09011-0/METRICS.

[10] M. Maral and A. Özdemir, "A systematic review on multi-criteria decision-making methods in educational research," *Br. Educ. Res. J.*, vol. 51, no. 6, pp. 01–36, Jul. 2025, doi: 10.1002/berj.70002.

[11] Y. Zheng, "Evaluation of Online Teaching Effect of Vocational College Teachers Based on TOPSIS Technology and the Hierarchical Chi-Square Model," *International Journal of Emerging Technologies in Learning*, vol. 18, no. 15, pp. 161–173, 2023, doi: 10.3991/ijet.v18i15.42251.

[12] S. Broumi, R. Sundareswaran, M. Shanmugapriya, P. K. Singh, M. Voskoglou, and M. Talea, "Faculty Performance Evaluation through Multi-Criteria Decision Analysis Using Interval-Valued Fermatean Neutrosophic Sets," *Mathematics*, vol. 11, no. 18, pp. 1–21, Sep. 2023, doi: 10.3390/math11183817.

[13] J. O. Adigun *et al.*, "Development and Evaluation of a Web Based System for Students' Appraisal on Teaching Performance of Lecturers," *International Journal of Information Engineering and Electronic Business*, vol. 14, no. 1, pp. 25–36, Feb. 2022, doi: 10.5815/ijieeb.2022.01.03.

[14] J. V. M. Bata, "A General-Purpose Web-based TOPSIS tool for Accessible Multi-Criteria Decision Making," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 5, pp. 3994–4005, Oct. 2025, doi: 10.52436/1.jutif.2025.6.5.4988.

[15] Q. H. Do, V. T. Tran, and T. T. Tran, "Evaluating lecturer performance in Vietnam: An application of fuzzy AHP and fuzzy TOPSIS methods," *Heliyon*, vol. 10, no. 11, May 2024, doi: 10.1016/j.heliyon.2024.e30772.

[16] N. Hayati, A. E. Syaputra, and Y. S. Eirlangga, "SISTEM PENUNJANG PENGAMBILAN KEPUTUSAN DALAM PEMBERIAN REWARD DOSEN TERBAIK MENGGUNAKAN METODE TOPSIS," *Jurnal Informatika dan Komputer*, vol. 11, no. 2, pp. 246–253, Oct. 2023, doi: 10.35508/jicon.v11.i2.12390.

[17] D. Yan and G. Li, "Applying multivariate generalizability theory to compose and rank the evaluation scores of college teachers' teaching ability," *Sci. Rep.*, vol. 15, no. 1, pp. 1–13, Dec. 2025, doi: 10.1038/s41598-025-08550-w.

[18] E. Bjarnason, F. Lang, and A. Mjöberg, "An empirically based model of software prototyping: a mapping study and a multi-case study," *Empir. Softw. Eng.*, vol. 28, no. 5, pp. 1–47, Sep. 2023, doi: 10.1007/s10664-023-10331-w.

[19] F. Quansah, A. Cobbinah, K. Asamoah-Gyimah, and J. E. Hagan, "Validity of student evaluation of teaching in higher education: a systematic review," *Front. Educ. (Lausanne).*, vol. 9, pp. 01–12, Feb. 2024, doi: 10.3389/feduc.2024.1329734.

[20] N. Indah Chasanah *et al.*, "BLACK BOX TESTING WEBSITE RIMA BATIK DENGAN TEKNIK BOUNDARY VALUE DAN EQUIVALENCE PARTITIONING," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 6, pp. 12814–12820, Dec. 2024.

[21] A. Wisesa Bong, E. Safitri, and T. Hendry, "Analisis dan Pengujian Sistem Informasi Stok dengan Boundary Value Analysis pada PT ABC," *Journal Of Information System, Computer Science And Information Technology*, vol. 5, no. 1, pp. 217–231, Jun. 2024.

[22] P. Vlachogianni and N. Tselios, "Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review," *Journal of Research on Technology in Education*, vol. 54, no. 3, pp. 392–409, 2022, doi: 10.1080/15391523.2020.1867938.

[23] M. Hyzy *et al.*, "System Usability Scale Benchmarking for Digital Health Apps: Meta-analysis," *JMIR Mhealth Uhealth*, vol. 10, no. 8, Aug. 2022, doi: 10.2196/37290.

[24] O. Suria, "A Statistical Analysis of System Usability Scale (SUS) Evaluations in Online Learning Platform," *Journal of Information Systems and Informatics*, vol. 6, no. 2, pp. 992–1007, Jun. 2024, doi: 10.51519/journalisi.v6i2.750.

[25] A. Nfh and A. T. Wardani, "Evaluating Usability of a Website-Based Learning Platform for Basic Programming Using the SUS Method," *INTEC Journal: Information Technology Education Journal*, vol. 4, no. 1, pp. 45–50, Feb. 2025.