

JUVOTEK JURNAL
Jurnal Vokasi Teknik | *J. Vok. Tek.*
https://mentech.id/jurnal/index.php/juvotek | e-ISSN 3025-1125
DOI: https://doi.org/10.53905/f2afxw50
Original Research Article

Development and Functional Evaluation of a Web-Based Semester Learning Plan Application Using an Agile Approach

Santika Thaymbola*[1], Gian Inggit Juniarsono[1], Septhia Sufendi[1], Junaidi[1], Muhammad Reza Ramadhan[1]

[1]Informatics Engineering, Samarinda State Polytechnic, Samarinda, Indonesia.

Received: january 19, 2026| Accpeted: March 25, 2026 | Published: April 01, 2026

ABSTRACT

Semester Learning Plans (Rencana Pembelajaran Semester, RPS) are essential instruments for aligning course activities, learning outcomes, instructional materials, schedules, and assessment in Indonesian higher education. However, fragmented document-based workflows may reduce accessibility, consistency, feedback quality, and updating efficiency. This study aimed to develop and functionally evaluate a web-based RPS application for lecturers, students, and administrators. The study employed an Agile software development approach comprising requirements analysis, system design, iterative implementation, testing, evaluation, and refinement. Requirements were obtained from lecturers and administrators in the Informatics Engineering Study Program at Samarinda State Polytechnic. The system was modeled using a context diagram, a level-1 data flow diagram, and an entity-relationship diagram. Django was used for the back end, while HTML, CSS, and JavaScript supported the web interface. Functional verification combined white-box testing of internal program logic and black-box testing of observable system behavior. The application implemented authentication, role-based access, course selection, learning-material management, feedback, timetable display, and dynamic navigation. All eight white-box test cases were successfully completed. Black-box testing covered 14 functional scenarios, of which 13 passed and one failed because the system permitted submission of an empty feedback comment, resulting in a descriptive pass rate of 92.9%. PDF files could be previewed and downloaded, whereas DOC and PPT files were available only for download. The application met most specified functional requirements and provides a structured foundation for digital RPS management. However, the findings demonstrate functional feasibility rather than improved usability or learning effectiveness because no controlled user study, usability scale, or performance benchmark was conducted. Future development should strengthen server-side validation, assess usability and security, integrate learning analytics, and evaluate interoperability with institutional learning management systems.

Keywords:

agile software development; black-box testing; django; higher education; semester learning plan; white-box testing.

Article Info

Check for updates

EDITED BY
Assoc. Prof. Dr. Herli Pardilla
Universitas Muhammadiyah Sumatera Utara, Indonesia.

COPYRIGHT©2026 (Author)
Santika Thaymbola
Gian Inggit Juniarsono
Septhia Sufendi
Junaidi
Muhammad Reza Ramadhan

***CORRESPONDENCE**
Santika Thaymbola
santikaadser@gmail.com

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

To cite this article:
Thaymbola, S., Juniarsono, G. I., Sufendi, S., Junaidi, J., & Ramadhan, M. R. (2026). Development and Functional Evaluation of a Web-Based Semester Learning Plan Application Using an Agile Approach. *Jurnal Vokasi Teknik (JuVoTek)*, 4(01), 1-9. https://doi.org/10.53905/f2afxw50

INTRODUCTION

The digital transformation of higher education extends beyond online teaching and learning platforms to encompass the administrative and curricular instruments that organize instruction. In Indonesia, the Semester Learning Plan, commonly referred to as the Rencana Pembelajaran Semester, functions as a comprehensive course-level roadmap that systematically links learning outcomes, specific topics, learning activities, detailed schedules, necessary learning resources, and various assessment methods (Darwis et al., 2024). This document also typically outlines criteria, indicators, and weighting for evaluation, ensuring a structured approach to achieving predefined learning objectives (Ulum, 2023, p. 346). Because lecturers and students depend on this detailed document throughout a semester, the accessibility, consistency, and updateability of RPS information directly affect the organization, quality, and effectiveness of teaching and learning activities (Darwis et al., 2024).

Conventional RPS management commonly relies on static files, fragmented communication channels, and manual document distribution (Naufal, 2023). Such practices often impose a significant administrative burden on lecturers (Yunanto et al., 2021) and can lead to several critical issues, including version

inconsistencies, delayed access to revised materials, and a disconnect between the formal learning plan and the dynamic learning resources and feedback generated during implementation (Nurtanto et al., 2021, p. 346). These challenges can ultimately hinder the effectiveness of the learning process. Therefore, web-based RPS systems have been developed to centralize course documentation, facilitate easier access, and dynamically support the management of lecture instruments (Aziira et al., 2023; Darwis et al., 2024; Firdaus et al., 2023; Sutrisno et al., 2023). Related research also indicates that digitally organized learning plans can significantly support curriculum implementation and foster structured independent learning, adapting to the evolving educational landscape (Ishaq & Rufaidah, 2019; Wijaya et al., 2023).

Nevertheless, existing studies often emphasize system construction and functional description without integrating a clear user-feedback mechanism or reporting the relationship between system requirements, architectural models, implementation outputs, and multiple testing perspectives (Alí et al., 2017, p. 230; Yusop et al., 2024, p. 14675). In addition, claims that an application improves learning efficiency are frequently presented without direct measurements of task completion time, usability, user satisfaction, learning outcomes, or comparative performance (Bellamy, 1988, p. 307; Wang et al., 2022, p. 7510). This creates an evidentiary gap between successful software operation and demonstrated educational impact.

Software quality evaluation is consequently essential. Black-box testing examines whether visible functions produce the expected outputs under specified input conditions, whereas white-box testing evaluates internal control flow, logic paths, and implementation structures (Raihan & Voutama, 2023; Sie et al., 2022). Combining both perspectives is particularly relevant for an academic information system because a feature may appear operational to users while still containing weak internal logic, or conversely, internal logic may be correct even though validation and interface behavior remain inadequate (Bostrom et al., 2009, p. 27; Kriswibowo et al., 2023).

The present study was undertaken to develop a web-based RPS application for the Informatics Engineering Study Program at Samarinda State Polytechnic using an Agile development approach. The study sought to: (1) identify the principal functional and non-functional requirements of lecturers, students, and administrators; (2) design and implement an integrated web-based system for RPS-related learning-material management, feedback, scheduling, and role-based access; (3) evaluate internal program logic through white-box testing; and (4) verify user-visible functionality through black-box testing. The study does not test causal effects on learning outcomes; rather, it establishes the functional readiness and limitations of the prototype as a basis for subsequent usability and educational-effectiveness research.

MATERIALS AND METHODS

Study Design and Setting

This study employed an applied software-development design. The development setting was the Informatics Engineering Study Program at Samarinda State Polytechnic, Samarinda, Indonesia. The unit of development was a web-based RPS application designed for three principal roles: lecturers, students, and administrators.

The available source manuscript reports that requirements were identified through surveys and interviews with lecturers and administrators. However, the number, demographic characteristics, recruitment procedure, question items, response scale, and dates of data collection were not documented. Accordingly, the requirements findings are treated as qualitative development inputs rather than statistically generalizable user-research results.

Agile Development Procedure

Table 1. Operational Activities Across the Agile-Based Application Development Stages

Stage	Operational activity
Requirements analysis	Identify functional requirements, non-functional requirements, user roles, expected inputs and outputs, access restrictions, and anticipated feedback pathways.
Planning and system modeling	Translate requirements into a context diagram, level-1 DFD, ERD, interface wireframes, and a prioritized implementation backlog.
Iterative implementation	Develop back-end and front-end components in short cycles, integrate completed functions, and refine features based on identified issues.
Verification and validation	Use white-box testing to inspect internal program logic and black-box testing to verify observable system behavior against expected results.
Evaluation and refinement	Document failed or incomplete behaviors, apply targeted corrections where reported, and identify requirements for subsequent releases.

Requirements Specification

Table 2. Functional Requirements and Acceptance Criteria for the Web-Based Semester Learning Plan Application

Requirement	Acceptance criterion
Authentication	Validate registered credentials and prevent unauthorized access.
Role-based access	Display functions according to lecturer, student, or administrator privileges.
Course selection	Allow users to select the relevant course before accessing course-specific functions.
Learning-material management	Allow authorized lecturers to upload, organize, edit, view, and distribute course materials.
File handling	Accept PDF, DOC, and PPT files; provide preview where supported and download access.
Feedback	Enable authenticated users to post comments and support two-way communication.
Timetable	Display dates and scheduled academic events dynamically.
Navigation	Provide consistent transitions between login, home, course, material, feedback, and schedule pages.

The non-functional requirements identified in the source manuscript were usability or convenience, performance, and security. Because no formal

thresholds were specified, these requirements were not evaluated quantitatively in the current study.

System Architecture and Data Modeling

The system design used three complementary models. The context diagram defined the system boundary and interactions among lecturers, students, administrators, and the application. The level-1 DFD decomposed the principal processes for managing materials, feedback, and learning information. The ERD represented the relationships among user, course, class, material, schedule, and feedback-related entities

.

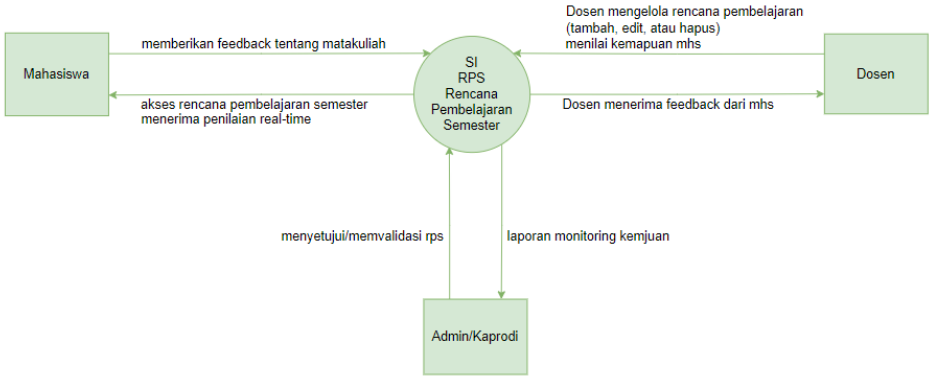


Figure 1. Context diagram of the web-based RPS application.

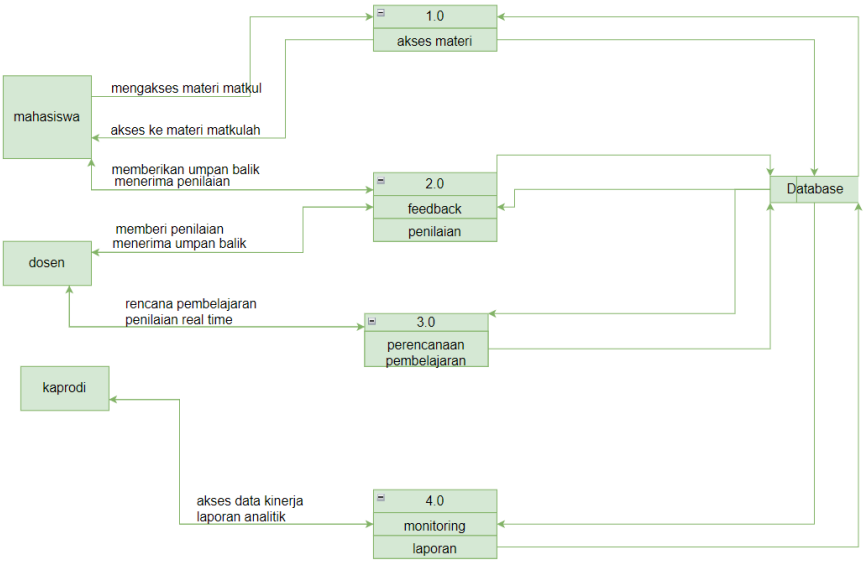


Figure 2. Level-1 data flow diagram of the RPS application.

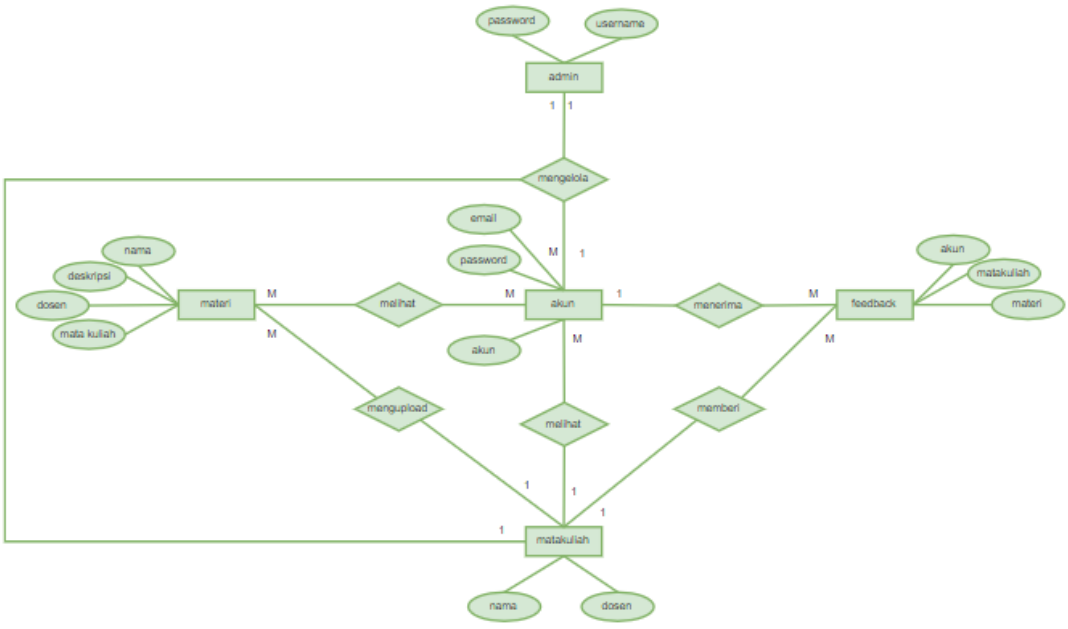


Figure 3. Entity-relationship diagram of the RPS application.

Technology Stack and Implementation Rules

Table 3. Technical Components and Implementation Configuration of the Web-Based Semester Learning Plan Application

Component	Implementation
Back end	Django web framework.
Front end	HTML, CSS, and JavaScript.
Supported upload formats	PDF, DOC, and PPT.
Preview behavior	PDF preview and download; DOC and PPT download without browser preview.
Access control	Role-based menus and authentication before protected functions.
Feedback workflow	Authenticated users submit comments that become visible within the feedback function.

Procedure and Data Analysis

White-box testing examined internal logic associated with authentication, menu selection, course selection, learning-material management, feedback access, timetable operation, navigation, and class-dropdown behavior. A test case was categorized as successful when the programmed path executed and produced the intended internal state or transition.

Black-box testing evaluated the application from the user's perspective using valid, invalid, empty, authenticated, unauthenticated, supported-file, and unsupported-file scenarios. Expected and actual behaviors were compared and classified as Pass or Fail. Results were summarized using frequencies and percentages. The descriptive pass percentage was calculated as passed scenarios divided by all tested scenarios multiplied by 100. No inferential statistical test was appropriate because the observations were software test cases rather than participant-level measurements.

RESULTS

Requirements Analysis

The requirements-analysis stage identified three principal needs: centralized learning-plan and material management, a direct feedback mechanism, and digital access to learning information. These needs were translated into role-specific functions for lecturers, students, and administrators. The design also incorporated convenience, performance, and security as non-functional priorities, although the source data did not contain measurable acceptance thresholds for these dimensions.

Interface Design and Implementation

The implemented prototype provided a material-input page for lecturers, a material-list page for students, and a feedback page for two-way communication. Course selection preceded lecturer material entry. Uploaded files were validated by extension, and users accessed protected functions after authentication.



Figure 4. Interface design for the learning-material input page.

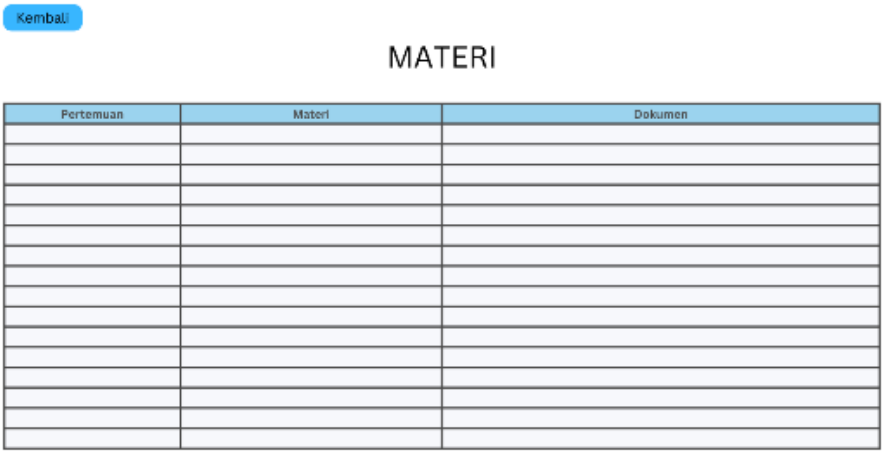


Figure 5. Interface design for the learning-material list page.

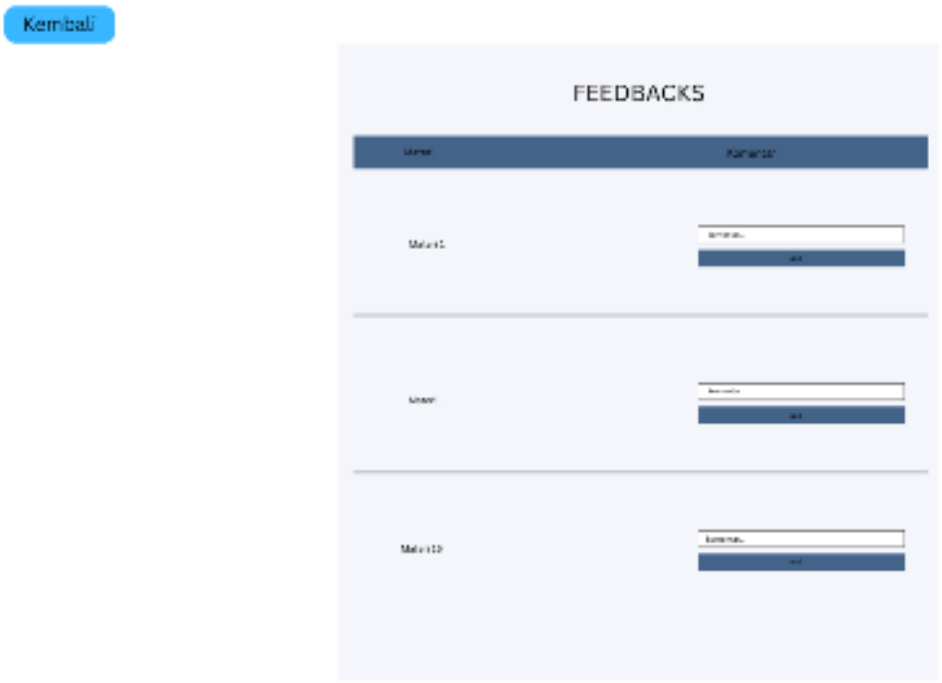


Figure 6. Interface design for the feedback page.

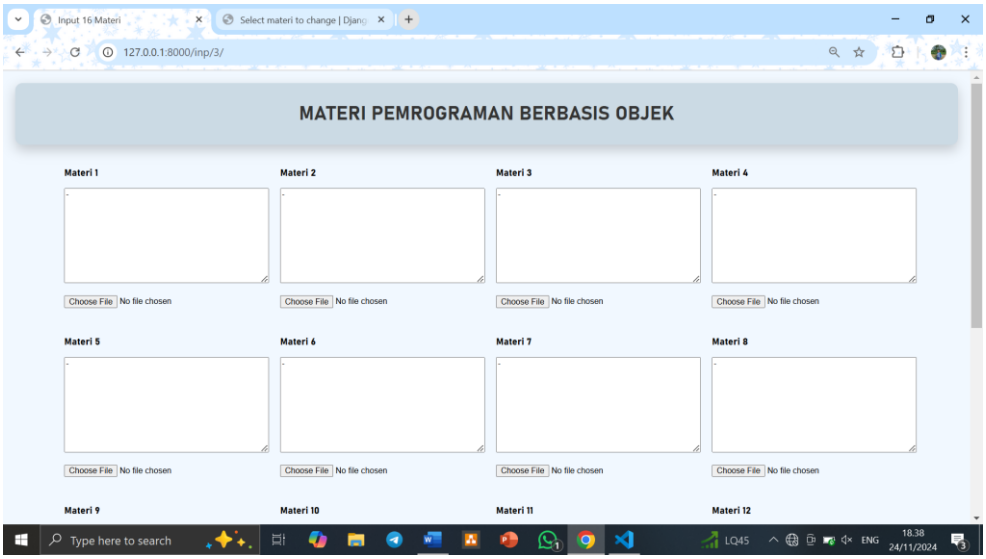


Figure 7. Implemented learning-material input page.

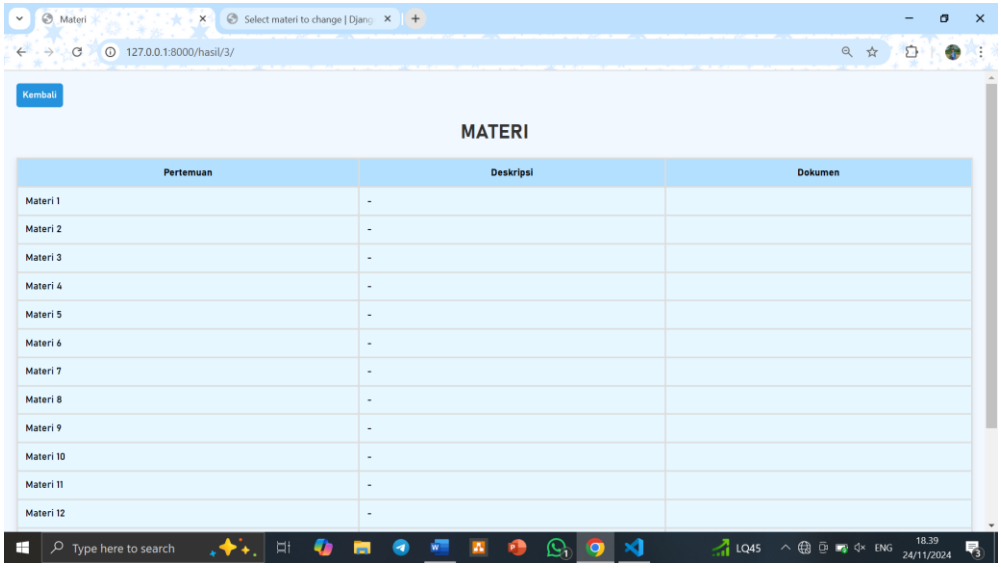


Figure 8. Implemented learning-material list page.

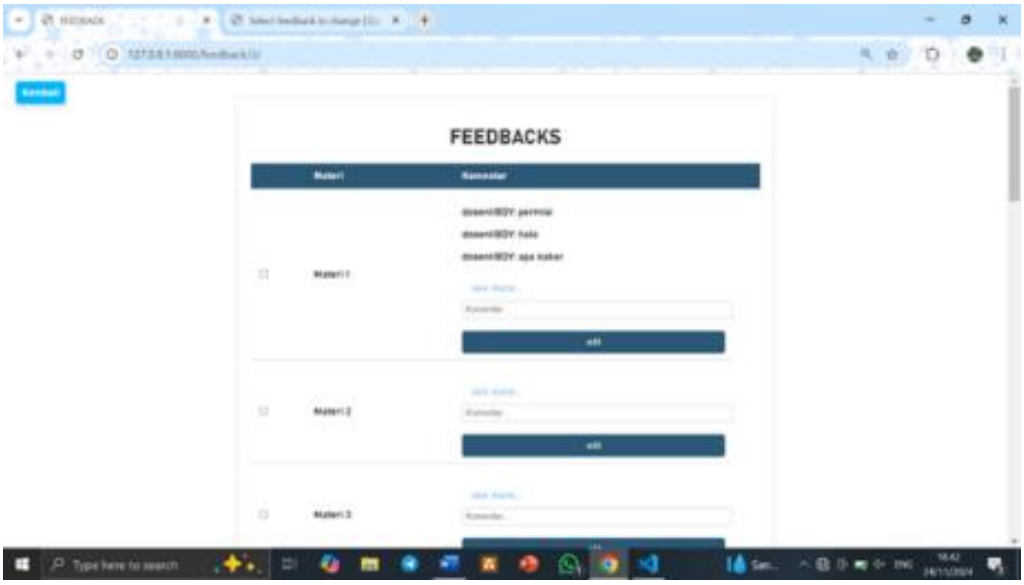


Figure 9. Implemented feedback page.

White-Box Testing Results

Table 4. White-box testing results

Test case	Status	Observation
Login status	Pass	Credential validation and redirection operated as intended.
Role-based menu options	Pass	Menus were restricted according to user role.
Course selection	Pass	Course selection path executed correctly.
Material management	Pass	Material-management logic executed successfully.
Feedback access	Pass	Authenticated users accessed the feedback function; the login modal was rarely visible because authentication preceded access.
Timetable	Pass	Timetable logic executed correctly.
Navigation	Pass	Page transitions followed the expected paths.
Class dropdown	Pass	The dropdown appeared only after students had been assigned to classes, consistent with the programmed condition.

Note. All eight white-box cases were reported as successful (8/8; 100%). The brief appearance of the login modal and conditional display of the class dropdown were interpreted as implementation behaviors rather than failed logic paths.

Black-Box Testing Results

Table 5. Black-box testing results

Module	Scenario	Status	Actual result
Login form	Valid credentials	Pass	Successful authentication and redirection to the main page.
Login form	Invalid or empty credentials	Pass	Access was denied and the login page remained active or was reloaded.
Home page	Role-based options	Pass	Lecturers could access student data; students could not access lecturer-only options.
Home page	Logout	Pass	Users were redirected to the login page.
Home page	Navigation menu	Pass	Menus operated according to the system design.
Activity page	Navigate to material input	Pass	Lecturers selected a course before entering the upload page.
Activity page	Back navigation	Pass	The back function returned users to course selection.
Material upload	Upload completed	Pass	The system redirected users to the material list.
Material upload	Unsupported format	Pass	Unsupported file formats were rejected.
Material access	Preview and download	Pass	PDF was previewable and downloadable; DOC/PPT were downloadable only.
Feedback	Valid comment	Pass	The comment was posted and visible.
Feedback	Empty comment	Fail	A blank comment could be submitted; required-field validation was absent or insufficient.
Feedback	Unauthenticated submission	Pass	Authentication was required before comment submission.
Schedule	Dynamic display	Pass	Dates and events were displayed dynamically.

Thirteen of the 14 black-box scenarios passed, producing a descriptive pass rate of 92.9%. The sole failed scenario concerned empty-comment submission. This result indicates that the major workflows were operational but input validation in the feedback module remained incomplete.

Table 6. Consolidated functional testing summary

Testing method	Passed	Failed	Total	Pass rate
White-box	8	0	8	100.0%
Black-box	13	1	14	92.9%
Combined descriptive total	21	1	22	95.5%

The combined percentage is a descriptive summary only. It should not be interpreted as a standardized software-quality score because white-box and black-

box test cases assess different constructs and were not weighted by risk or severity.

Significant Findings

1. Role-based authentication and authorization functioned across the tested navigation paths.
2. The application integrated course selection, learning-material management, feedback, and timetable functions within one web-based environment.
3. File restrictions operated as designed, but browser preview was limited to PDF files.
4. The most important defect was the acceptance of an empty feedback comment, demonstrating the need for both client-side and server-side validation.
5. The available evidence establishes functional feasibility, not increased lecturer efficiency, student comprehension, usability, security, or learning effectiveness.

DISCUSSION

Interpretation of the Findings

The results indicate that the prototype successfully translated most identified requirements into functioning web features. Complete success across the reported white-box cases suggests that the principal internal control paths were implemented coherently. The black-box results further show that authentication, role separation, navigation, material handling, feedback access, and timetable display behaved as expected in nearly all scenarios.

The empty-comment failure is technically minor in comparison with authentication or authorization defects, but it is important for data quality and communication integrity. Blank records can reduce the interpretability of feedback histories, create unnecessary database entries, and complicate downstream analytics. Corrective action should therefore include required-field constraints in the interface, server-side validation in Django forms or serializers, trimming of whitespace-only input, informative error messages, and regression testing after remediation.

The differentiated handling of file types is another relevant finding. Supporting preview for PDF while restricting DOC and PPT to download is operationally acceptable, but users may expect consistent preview behavior. Future implementations could convert office documents to PDF, use a secure document viewer, or explicitly communicate preview limitations before upload.

Comparison with Previous Studies

The system's centralized management of RPS-related information is consistent with prior web-based RPS and lecture-instrument applications designed to reduce fragmented administrative workflows (Aziira et al., 2023; Firdaus et al., 2023; Sutrisno et al., 2023)The current development extends this orientation by incorporating a feedback function and reporting both internal-logic and external-functional testing.

The black-box procedure is consistent with the functional-verification approach described by (Raihan & Voutama, 2023), in which test inputs are compared with expected observable outcomes. Likewise, the use of white-box testing reflects the concern for control-flow correctness emphasized by (Sie et al., 2022). The combination of these approaches strengthens functional confidence because it evaluates both program structure and visible behavior, although it does not replace usability, security, compatibility, or performance testing.

The findings also align conceptually with studies emphasizing that digitally structured learning plans can support access to course information and systematic learning administration (Ishaq & Rufaidah, 2019; Wijaya et al., 2023). However, the present study cannot conclude that the application improved student understanding or lecturer efficiency because those outcomes were not directly measured. Such claims require task-based comparisons, validated usability instruments, learning-outcome data, or longitudinal usage analytics.

Practical and Research Implications

For practice, the prototype offers a foundation for consolidating RPS documents, materials, communication, and schedules. Role-based access can reduce accidental exposure of lecturer-only functions, while centralized materials may improve document traceability. Before institutional deployment, the application should be subjected to security testing, backup and recovery testing, browser and device compatibility testing, accessibility review, and load testing under realistic concurrent use.

For research, the next development phase should distinguish software quality from educational effectiveness. Usability can be evaluated through completion time, error rate, learnability, the System Usability Scale, and qualitative interviews. Educational value can be examined through access logs, timeliness of material use, quality of feedback interactions, and course-level learning outcomes. A comparative or pretest-posttest design would be needed to support claims of improvement.

Limitations

1. The source manuscript did not report the number or characteristics of lecturers, administrators, or students involved in requirements elicitation or evaluation.
2. No validated usability instrument, user-satisfaction measure, task-completion time, response-time benchmark, or learning-outcome measure was provided.
3. Testing was reported as scenario outcomes without code coverage, cyclomatic complexity, severity classification, repeated trials, test environment specifications, or independent tester information.
4. Performance, cybersecurity, accessibility, database integrity, compatibility, and scalability were identified as relevant but were not empirically evaluated.

5. The study was conducted within one study program, limiting institutional generalization.

The evidence therefore supports prototype functionality, not causal claims regarding efficiency, comprehension, or educational effectivene.

CONCLUSION AND SUGGESTIONS

The study developed a web-based Semester Learning Plan application using an Agile approach and implemented authentication, role-based menus, course selection, learning-material management, feedback, timetable display, and structured navigation. The results reinforce the central proposition that an integrated web system can operationalize core RPS-management requirements: all eight reported white-box cases passed, and 13 of 14 black-box scenarios passed. The main unresolved defect was the acceptance of empty feedback comments. Thus, the prototype demonstrates strong functional readiness but not yet proven usability, efficiency, security, or educational impact. Its potential contribution lies in providing a centralized and extensible foundation for digital course-planning administration. The authors recommend that subsequent development add robust server-side input validation, document-preview enhancement, learning analytics, LMS interoperability, security and performance testing, accessibility compliance, and user-centered evaluation with lecturers and students. Authors submitting related future studies are encouraged to report participant characteristics, reproducible test protocols, quantitative usability indicators, software-quality metrics, and direct evidence of learning or administrative outcomes.

Acknowledgments

The authors acknowledge the support of the Informatics Engineering Study Program, Samarinda State Polytechnic, and the lecturers and administrators who contributed requirements and feedback during application development.

Conflict of Interest

The authors declare that they have no known financial, professional, or personal conflicts of interest that could have influenced the work reported in this article

REFERENCES

Alí, N., Baker, S., O'Crowley, R., Herold, S., & Buckley, J. (2017). Architecture consistency: State of the practice, challenges and requirements. *Empirical Software Engineering*, *23*(1), 224–258. https://doi.org/10.1007/s10664-017-9515-3

Aziira, A. H., Kamil, H., & Kartika, A. D. (2023). Pembangunan Sistem Informasi Pengelolaan Rencana Pembelajaran Semester (RPS). *Jurnal Nasional Teknologi dan Sistem Informasi*, *9*(2), 103–112. https://doi.org/10.25077/teknosi.v9i2.2023.103-112

Bellamy, R. (1988). Advances in human-computer interaction. *International Journal of Information Management*, *8*(2), 147–147. https://doi.org/10.1016/0268-4012(88)90025-4

Bostrom, R. P., Gupta, S., & Thomas, D. (2009). A Meta-Theory for Understanding Information Systems Within Sociotechnical Systems. *Journal of Management Information Systems*, *26*(1), 17–48. https://doi.org/10.2753/mis0742-1222260102

Darwis, D. (Darwis), Yusran, Y. (Yusran), Sau, T. (Tenri), & Yani, A. (Ahmad). (2024). RENCANA PEMBELAJARAN SEMESTER (RPS). In *Neliti*. https://repository.penerbitwidina.com/publications/568705/rencana-pembelajaran-semester-rps

Firdaus, T. A., Putra, R. H., Arifandi, F., Anam, M. K., & Lathifah, L. (2023). Implementasi Sistem Rencana Pembelajaran Semester Berbasis Web Untuk Mempermudah Proses Pembelajaran. *Jurnal Teknoinfo*, *17*(1), 156–156. https://doi.org/10.33365/jti.v17i1.2348

Ishaq, I., & Rufaidah, S. (2019). Analisis Belajar Mandiri Mahasiswa Melalui Tugas Terstruktur Terhadap Capaian Pembelajaran Mata Kuliah Aplikasi Komputer (Analisis Penerapan Kkni Dan Rencana Pembelajaran Semester). *Jurnal Sains dan Pendidikan Fisika*, *15*(1). https://doi.org/10.35580/jspf.v15i1.9412

Kriswibowo, R., Prayogo, J. S., Febriana, R. W., & Alia, P. A. (2023). Implementasi Black Box Testing dan Acceptance Testing Fitur SKKM pada Cybercampus.uam.ac.id Universitas Anwar Medika. *Jurnal Informatika Universitas Pamulang*, *8*(4), 561–567. https://doi.org/10.32493/informatika.v8i4.36904

Naufal, M. D. (2023). Rancang Bangun E-RPS Prodi Teknik Informatika IIB Darmajaya Berbasis Web. *ETNIK Jurnal Ekonomi Dan Teknik*, *2*(10), 869–879. https://doi.org/10.54543/etnik.v2i10.234

Nurtanto, M., Kholifah, N., Masek, A., Sudira, P., & Samsudin, A. (2021). Crucial problems in arranged the lesson plan of vocational teacher. *International Journal of Evaluation and Research in Education (IJERE)*, *10*(1), 345–345. https://doi.org/10.11591/ijere.v10i1.20604

Raihan, H., & Voutama, A. (2023). Pengujian Black Box Pada Aplikasi Database Perguruan Tinggi dengan Teknik Equivalence Partition. *Antivirus jurnal ilmiah teknik informatika*, *17*(1), 1–18. https://doi.org/10.35457/antivirus.v17i1.2501

Sie, J. B. L., Musdar, I. A., & Bahri, S. (2022). Pengujian White Box Testing Terhadap Website Room Menggunakan Teknik Basis Path. *KHARISMA Tech*,

17(2), 45–57. https://doi.org/10.55645/kharismatech.v17i2.235

Sutrisno, D., Dewi, A. Y., & Rosyadi, I. (2023). Rancang Bangun Aplikasi Instrumen Perkuliahan Pada Fakultas Teknik dan Ilmu Komputer UMPP Berbasis WEB. *Jurnal Surya Informatika*, *13*(1), 52–60. https://doi.org/10.48144/suryainformatika.v13i1.1440

Ulum, M. (2023). Pengembangan Kurikulum Mata Kuliah Umum Bahasa Indonesia. *Teks Jurnal Penelitian Bahasa Sastra dan Pengajarannya*, *8*(2), 340–351. https://doi.org/10.26877/teks.v8i2.17179

Wang, W., Lin, Y., & Lu, H.-E. (2022). Exploring the effect of improved learning performance: A mobile augmented reality learning system. *Education and Information Technologies*, *28*(6), 7509–7541. https://doi.org/10.1007/s10639-022-11487-6

Wijaya, A. B., Santiko, I., Suliswaningsih, S., Hamdi, A., Anuggilarso, L. R., & Zettira, S. B. Z. (2023). Model Rencana Pembelajaran Semester (RPS) Menggunakan Konsep Kurikulum Mandiri Pada Mobile Learning. *CogITo Smart Journal*, *9*(1), 145–155. https://doi.org/10.31154/cogito.v9i1.446.145-155

Yunanto, A. A., Rochimah, S., Arifiani, S., & Dhuha, M. S. (2021). *Design and Implementation of Rencana Pembelajaran Semester (RPS) Document Generation System*. 125–130. https://doi.org/10.1109/icamimia54022.2021.9807731

Yusop, N. S. M., Rosli, M. M., Farid, N. F., Nazri, N. A. S. M., Jamil, N., & Ramli, M. I. (2024). Development of a Collaborative Intelligent Individual Education Program System using a Prototyping Approach. *Engineering Technology & Applied Science Research*, *14*(3), 14666–14676. https://doi.org/10.48084/etasr.7352