Vol. 10, No. 01, January 2026, pp. 36-46

INTERNATIONAL JOURNAL OF INNOVATION IN ENTERPRISE SYSTEM

Journal homepage: https://ijies.sie.telkomuniversity.ac.id
e-ISSN: 2580-3050, DOI: https://doi.org/10.25124/ijies.v10i01.9808

Design and Functional Verification of a Two-Stage Smart Helpdesk for Student Information-System Support Using Forward Chaining and Content-Based Article Retrieval

Fahmi Reza Ferdiansyah[1]*, Heri Purwanto[2], Rikky Wisnu Nugraha[3], Rudy Sofian[4], Gunadhi Pratama[5]

[1] Informatic
Ekuitas University, Bandung, West Java
fahmi.reza@ekuitas.ac.id

[2] Information System
Sangga Buana University, Bandung, West Java
heri.purwanto@usbypkp.ac.id

[3] Information System
Widyatama University, Bandung, West Java
rikky.nugraha@widyatama.ac.id

[4] Information System
Universitas Informatika Dan Bisnis Indonesia, Bandung, West Java
rudysofian@unibi.ac.id

[5] Informatic Engineering
Institut Digital Ekonomi LPKIA, Bandung, West Java
gunadhi@lpkia.ac.id

*Corresponding Author

ARTICLE INFO

Article history:
Received 12 August 2026
Accepted 16 October 2026
Published 20 October 2026

Keywords: Content-based filtering; expert system; forward chaining; helpdesk; smart helpdesk.

ABSTRACT

Manual handling of student information-system complaints at Institut Digital Ekonomi LPKIA can delay assistance, while a rule-based helpdesk may fail to return useful guidance when user-reported indications do not completely match the knowledge base. This study aims to design, implement, and functionally verify a two-stage smart helpdesk that maintains support in both matched and unmatched cases. In the first stage, forward chaining evaluates Boolean indications against IF-THEN rules to map known cases to predefined problem codes and solutions. In the second stage, unresolved cases are redirected to content-based article retrieval, where candidate articles are filtered by problem category, matched using user-entered keywords, and ordered using average user ratings. The web prototype was implemented using Laravel and MySQL and examined through white-box and functional verification of the two control-flow paths. The verified workflow returned the corresponding diagnosis for complete rule matches and activated article retrieval when no complete rule was satisfied. The main contribution is the integration of deterministic diagnosis and a recommendation fallback in one higher-education helpdesk workflow, preventing unresolved cases from ending without guidance. Because no independent labeled test set was available, statistical diagnostic accuracy and recommendation relevance are reserved for future evaluation.

This is an open access article under the CC BY-NC-SA license

1. INTRODUCTION

Helpdesk services in higher education are expected to provide timely and consistent support for users of institutional information systems. At Institut Digital Ekonomi LPKIA, complaints related to academic and supporting information systems are still handled manually by staff from the Information Systems Department. This process can delay responses and lead to inconsistent resolution when the information provided by students is incomplete. Previous smart-helpdesk studies have used forward chaining with certainty factors[1], Naive Bayes classification [2], backward chaining [3], and forward-chaining chatbots [4]. In parallel, content-based recommendation studies have used item attributes, user preferences, and text representations to retrieve relevant items[5]–[9]. Foundational and survey literature further characterizes content-based recommendation and the broader design space of recommender systems [10]–[12].

Despite these developments, a rule-based diagnosis can fail to produce a conclusion when the facts reported by a user do not satisfy a complete rule or when the case falls outside the available knowledge base. In that situation, a helpdesk should still provide useful support rather than terminate the interaction without guidance. This creates a practical research gap: combining deterministic problem diagnosis with content-oriented article retrieval as a fallback within one higher-education helpdesk workflow.

The specific objective of this study is to design and implement a smart helpdesk for Institut Digital Ekonomi LPKIA that (1) diagnoses known student information-system problems using forward chaining over a predefined knowledge base and (2) provides category- and keyword-based article recommendations when a direct diagnosis is not obtained. The study focuses on the operational integration of these two mechanisms rather than proposing a new machine-learning algorithm.

Briefly, the contributions of this paper are: (1) an integrated two-stage smart-helpdesk workflow that connects rule-based diagnosis to a recommendation fallback; (2) a knowledge base linking student problem indications, diagnostic rules, problem codes, and corresponding solutions across several campus information systems; and (3) a web-based implementation in which unresolved cases are redirected to relevant support articles instead of producing no assistance.

2. PROPOSED METHOD

2.1 Research Positioning and State of the Art

Related helpdesk research shows several ways to automate support. Partha et al. combined forward chaining with certainty factors for IT-helpdesk problem solving [1]; Suryono et al. used a Naive Bayes classifier to route smart-helpdesk complaints [2]; Hariona applied backward chaining to e-government helpdesk incidents [3]; and Bambang et al. implemented a forward-chaining chatbot for helpdesk and course-reservation services [4]. Comparative and diagnostic expert-system studies also demonstrate the use of deterministic forward-chaining reasoning [13], [14]. Recommendation-system studies, meanwhile, use item or text attributes to retrieve relevant items [5]–[12], [15]. For text retrieval, classical term weighting, probabilistic ranking, and sentence-embedding approaches provide useful baselines for lexical and semantic matching [16]–[18]. These studies establish the individual diagnostic and recommendation approaches, but they do not address the same operational gap considered here: maintaining useful helpdesk support when a complete rule-based diagnosis cannot be produced. The present study therefore positions forward chaining as the primary mechanism for known problems and category/keyword article retrieval as a fallback for unresolved cases. The contribution is the operational integration of these two stages in a higher-education helpdesk, not a new inference or learning algorithm.

Table 1 - State-of-the-Art Positioning of the Proposed Smart Helpdesk Workflow

Reference	Approach	Position relative to this study
[1] Partha et al. (2021)	Forward chaining + certainty factor in IT-helpdesk	Rule-based helpdesk with confidence handling; no article-recommendation fallback.
[5] Suryono et al. (2017)	Naive Bayes smart-helpdesk classification	Learns ticket classes from complaint data; different from deterministic diagnosis.
[6] Hariona (2021)	Backward chaining for e-government helpdesk	Goal-driven expert-system reasoning; no content recommendation stage.
[8] Bambang et al. (2021)	Forward-chaining chatbot/helpdesk	Rule-based conversational support; no separate article-retrieval fallback.
[9]-[15]	Content-based recommendation using item/text features	Establish content-based recommendation principles and item/text representations; do not perform helpdesk diagnosis.
Present study	Forward chaining + category/keyword article retrieval	Two-stage workflow: diagnose known cases first; retrieve support articles when diagnosis is unresolved.

*Corresponding author: fahmi.reza@ekuitas.ac.id

The comparison shows that existing helpdesk studies generally emphasize a single diagnostic or classification mechanism, while recommendation studies focus on item retrieval. This research connects both functions in one operational sequence so that a failed rule match does not end the support process.

2.2 Research Phases

The research was organized into five phases. Each phase addresses a specific problem and produces an output required by the next phase, thereby maintaining a coherent relationship between the helpdesk problem, the selected method, and the implemented solution.

Table 2 - Research Phases, Addressed Problems, and Outputs

Phase	Problem addressed	Procedure and output
1. Problem identification and knowledge acquisition	Manual handling and recurring student information-system problems.	Identify helpdesk categories, problem types, indications, solutions, and article resources using institutional domain knowledge.
2. Knowledge-base construction	User symptoms must be represented consistently.	Encode problem codes E001-E011, indication codes I001-I022, and IF-THEN rules linking indications to diagnoses and solutions.
3. Forward-chaining diagnosis	Known problems require deterministic conclusions from user-reported facts.	Collect Boolean responses, evaluate rule antecedents, and return the corresponding problem code when all conditions are satisfied.
4. Recommendation fallback	Some inputs do not satisfy a complete diagnostic rule.	Use the selected category and user keywords to retrieve matching support articles; use average user ratings as an additional ordering signal.
5. Implementation and verification	The workflow must operate as an integrated web service.	Implement with Laravel and MySQL; verify diagnostic and fallback control-flow paths using functional/white-box examination.

The phases are sequential but connected: knowledge acquired in Phase 1 is formalized in Phase 2; Phase 3 uses that knowledge for diagnosis; Phase 4 handles unresolved cases; and Phase 5 integrates and verifies the complete workflow.

2.3 Forward-Chaining Diagnostic Phase

The diagnostic input consists of the problem category selected by the student and Boolean answers to the indication questions. The inference engine evaluates the IF-THEN rules stored in the knowledge base. A diagnosis is produced only when all antecedents of a rule are satisfied. For example, IF I001 is TRUE AND I002 is TRUE THEN E001 maps the corresponding indications to the Student Data Error in SiAkad problem. The same procedure is applied to the remaining rules. If no rule is completely satisfied, the system does not force a diagnosis; control is transferred to the article-recommendation phase.

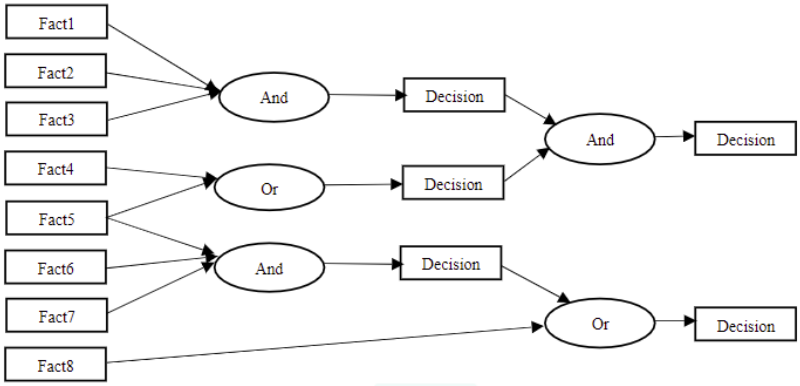


Figure 1 - Forward-Chaining Inference Pattern Used to Evaluate the Rule Base

2.4 Content-Based Article Recommendation Phase

The recommendation phase is activated after an unresolved diagnostic result. The implementation first reads the selected solution category and the sentence entered by the user, then splits the sentence into a keyword array. Candidate articles are restricted to the selected category, and article content is searched using string matching for the entered keywords. For matched articles, the system calculates the average user rating and uses the rating as an additional ordering signal. If no article contains the supplied keywords, no recommendation is returned for that query. Because this implementation relies on explicit article content, category information, and keyword matching rather than a learned user profile, it should be interpreted as a lightweight content-based retrieval mechanism.

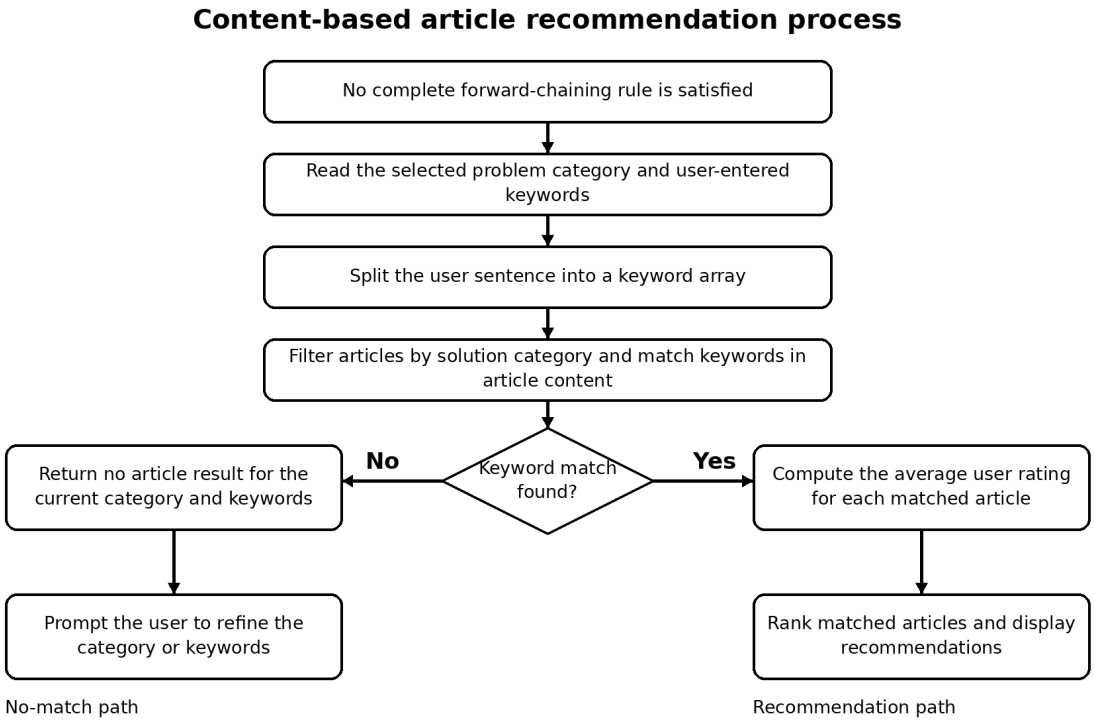


Figure 2 - Content-Based Article Recommendation Process Used in the Smart Helpdesk

2.5 System Implementation and Verification

The integrated system was implemented as a web application using the Laravel framework and a MySQL database. Verification focuses on the control flow of the implemented workflow: a complete rule match must return the corresponding diagnostic code and solution, whereas an incomplete match must activate the recommendation path. White-box examination is used to inspect these branches and the transitions between diagnosis and recommendation. The available study data do not include an independent labeled test set; therefore, the evaluation in this paper is functional rather than a statistical estimate of diagnostic accuracy or recommendation precision.

3. RESULTS

3.1 Knowledge-Base Implementation

The results are reported against the study objective: implementing a two-stage helpdesk that provides a rule-based diagnosis for known cases and a content-oriented recommendation fallback for unresolved cases. The resulting knowledge base contains 11 problem types (E001-E011), while the rule set references indication codes I001-I022.

Table 3 - Knowledge Base - Problem Types

No	Error Code	Error Name
1	E001	Student Data Error in SiAkad
2	E002	Invalid Login – SiAkad
3	E003	Parent Login – SiAkad
4	E004	Invalid KRS in SiAkad
5	E005	Invalid Password Reset – SiAkad
6	E006	Invalid Login – eLearning
7	E007	Invalid Login – Fellow
8	E008	Invalid Login – LUB
9	E009	LUB – Invalid Lecturer & Course Selection
10	E010	Inactive e-Proposal Account
11	E011	e-Proposal – Submission Disabled

After the problem types were encoded, representative indication questions were organized as shown in Table 4. These Boolean indications provide the facts evaluated by the forward-chaining rules.

Table 4 - Knowledge Base - Problem Indications (Excerpt)

No	Indication Code	Indication Description
1	I001	Is there an error in the student's name?
2	I002	Is there an error in the parent's name in the student data?
3	I003	Do you see a message stating "Invalid username or password" on the login page?
4	I004	Have you updated your SiAkad account password?
5	I005	Is there no NIK recorded in the parent's (mother or father) data?
6	I006	Have you logged in using a parent's NIK in SiAkad?
7	I007	Have you not paid the semester tuition?
…	…	…

Note: Table 4 shows the indication entries available in the manuscript excerpt. The implemented rule base refers to indication codes I001-I022; the full operational knowledge base should be provided as supplementary material for reproducibility.

The indication codes were then combined into the expert-system rules shown in Table 5. Each rule maps a specific combination of Boolean facts to one problem code.

Table 5 - Knowledge Base - Expert System Rules

No	Rule
1	IF I001 is TRUE AND I002 is TRUE THEN E001
2	IF I002 is FALSE AND I003 is TRUE AND I004 is TRUE THEN E002
3	IF I003 is FALSE AND I005 is TRUE AND I006 is FALSE THEN E003
4	IF I001 is FALSE AND I007 is TRUE AND I008 is TRUE THEN E004
5	IF I004 is FALSE AND I009 is FALSE AND I010 is TRUE THEN E005
6	IF I011 is TRUE AND I012 is TRUE THEN E006
7	IF I013 is TRUE AND I014 is TRUE THEN E007
8	IF I015 is TRUE AND I016 is FALSE THEN E008
9	IF I015 is FALSE AND I017 is TRUE AND I018 is TRUE THEN E009
10	IF I019 is TRUE AND I020 is TRUE THEN E010
11	IF I019 is FALSE AND I021 is TRUE AND I022 is TRUE THEN E011

All antecedents in a rule must be satisfied before the corresponding error code is returned. This strict condition prevents the system from assigning a diagnosis from a partial match. When no complete rule is satisfied, the unresolved case is transferred to the recommendation module instead of forcing a closest diagnostic code. Figure 3 summarizes the implemented rule paths.

*Corresponding author: fahmi.reza@ekuitas.ac.id

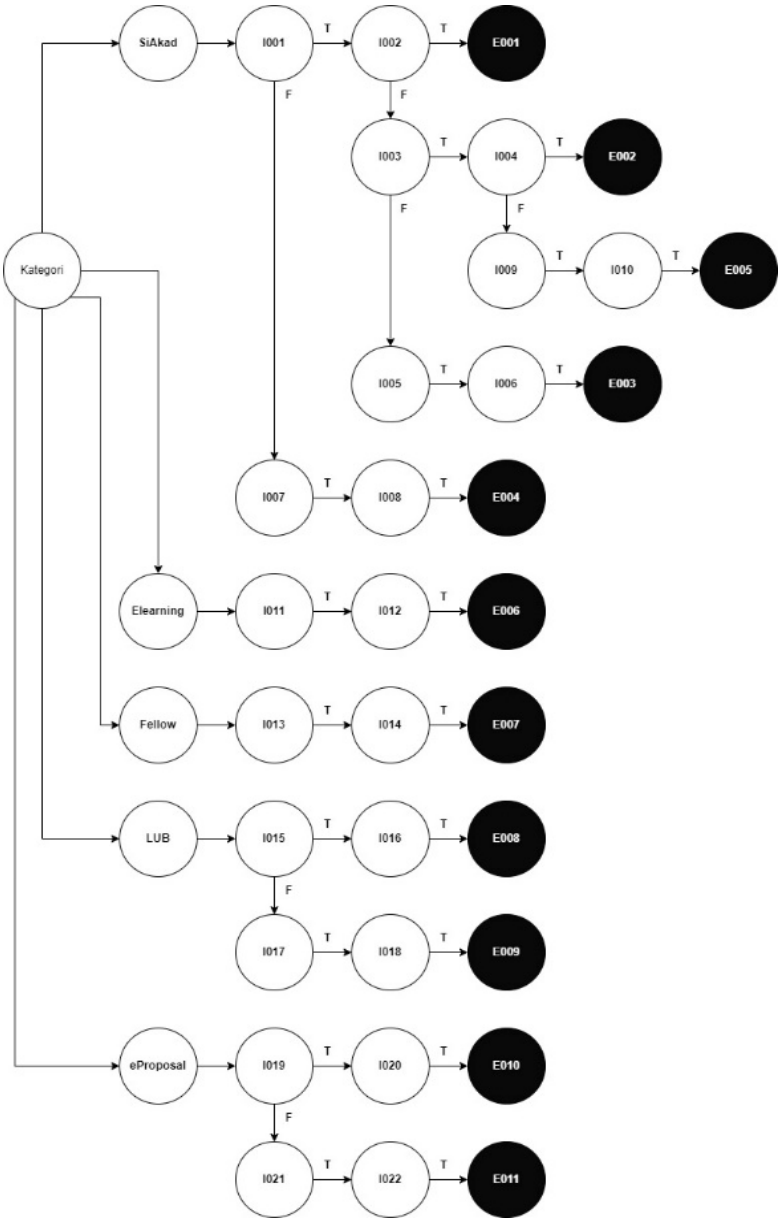


Figure 3 - Forward-Chaining Rule Flow for the Implemented Helpdesk Knowledge Base

3.2 Recommendation and Fallback Output

The recommendation module is triggered when the diagnostic rules do not yield a conclusion. In this path, the system restricts candidate articles to the selected problem category, matches the user's keywords against article content, calculates the average rating of matched articles, and presents the resulting articles in ranked order. This behavior is visible in the article-list and rating interfaces shown later in this section.

The module therefore extends system coverage for unresolved cases, but its relevance mechanism is based on category filtering and string matching rather than semantic similarity or a learned user profile. The result should be interpreted as content-oriented retrieval with rating-based ordering.

*Corresponding author: fahmi.reza@ekuitas.ac.id

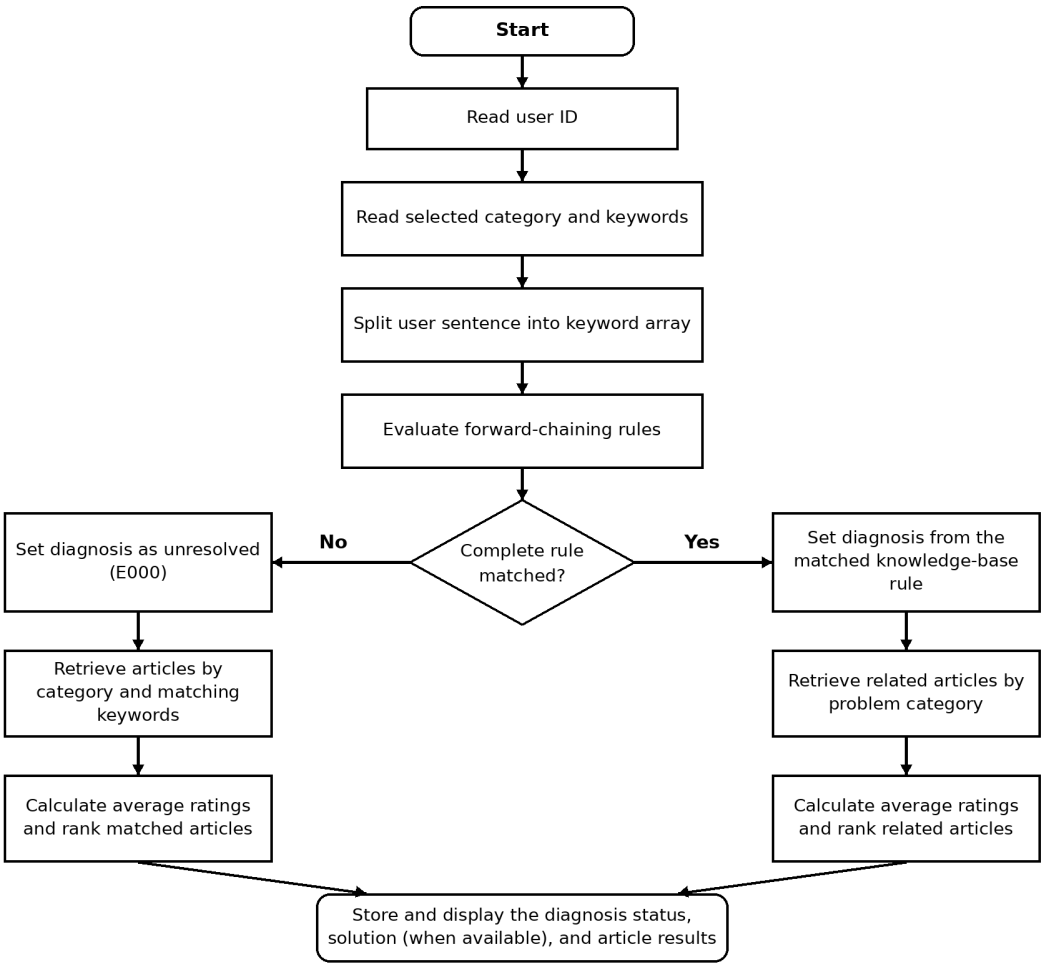


Figure 4 - Integrated Diagnostic and Fallback Workflow

3.3 Functional Verification

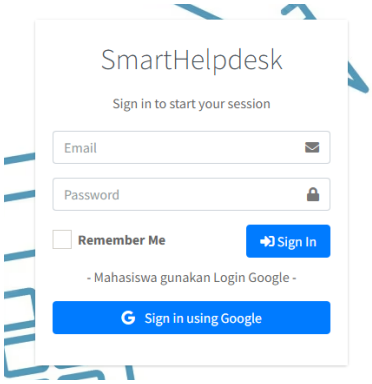
The integrated workflow was examined using white-box testing of the diagnostic control flow. The verification objective was to confirm two core behaviors: (1) a complete rule match returns the corresponding diagnostic result and related support information, and (2) a case without a complete rule match is redirected to the article-recommendation path. Figure 4 represents these alternative branches. The system was implemented with Laravel and MySQL. Because the manuscript does not provide an independent labeled dataset or a quantitative recommendation benchmark, the evidence supports functional correctness of the workflow but not a statistical claim of diagnostic accuracy, recommendation precision, or efficiency.

Table 6 - Functional Verification Scenarios and Criteria

Scenario	Input condition	Verification criterion and available evidence
Rule-matched diagnosis	All antecedents of one predefined rule are satisfied.	The corresponding E-code, indication/solution path, and related support output are produced. The implemented result interface is illustrated in Figure 7.
Unresolved diagnosis	No complete predefined rule is satisfied by the reported facts.	The diagnostic status remains unresolved and the workflow continues to category/keyword article retrieval rather than terminating without guidance. The fallback output is illustrated in Figure 8.
Article rating and ordering	Retrieved articles have one or more submitted user ratings.	The system calculates the average rating and uses it as an ordering signal for the displayed article results; the rating interface is illustrated in Figure 9.

These scenarios constitute a functional control-flow experiment. They verify that the implemented branches behave as designed, but they do not constitute a statistical benchmark of diagnostic accuracy or recommendation relevance.

a



b

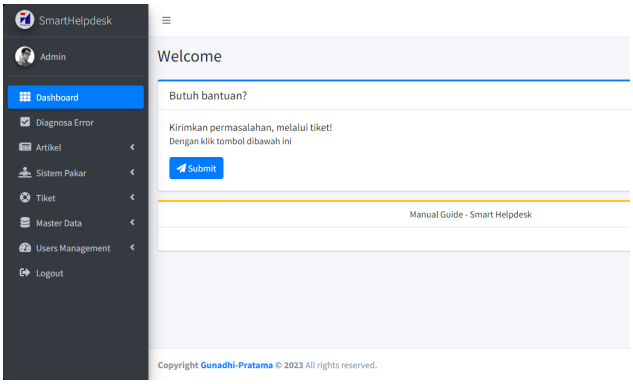


Figure 5 – (a) Login Interface; (b) Dashboard Interface

Figure 5 presents the login and dashboard interfaces. Administrator accounts use previously registered credentials, while student users can register and sign in using Gmail. After authentication, users access role-specific dashboard menus.

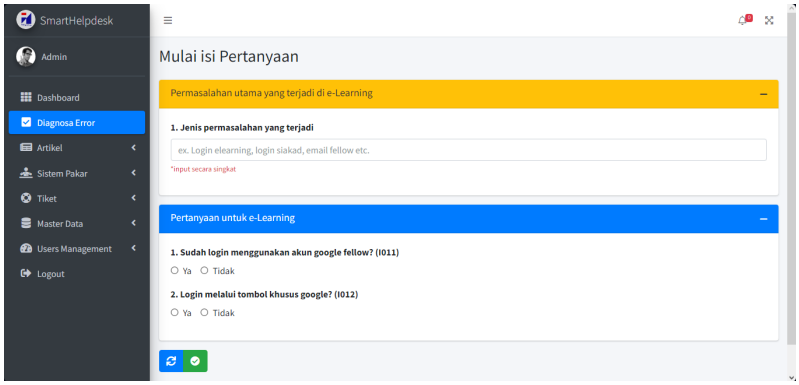


Figure 6 - Problem-Diagnosis Interface

Figure 6 presents the problem-diagnosis interface, where students answer indication questions that supply facts to the forward-chaining rules.

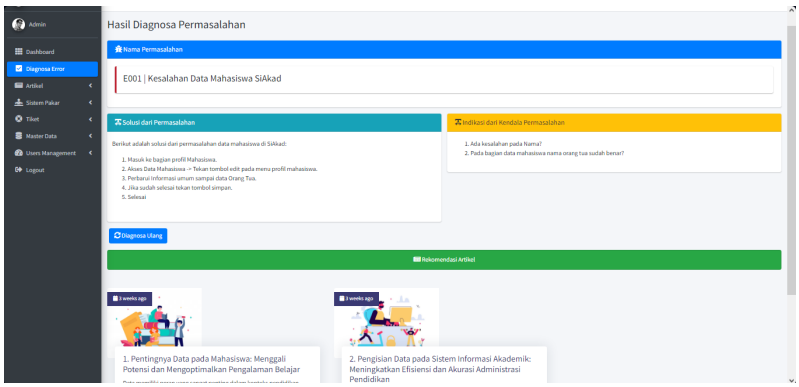


Figure 7 - Diagnosis-Results Interface

Figure 7 presents the diagnostic-result interface. For an identified problem, the system displays the problem name, relevant indications, a solution, and related articles. For an unresolved problem, the workflow switches to article retrieval using the selected category and entered keywords; matching articles are then ordered using their user ratings.

*Corresponding author: fahmi.reza@ekuitas.ac.id

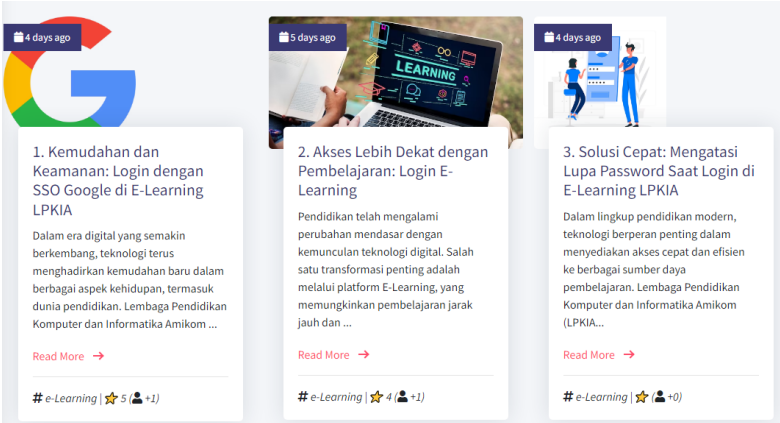


Figure 8 - Article-list interface for unresolved diagnoses

Figure 8 displays the article list returned for an unresolved diagnosis after category and keyword filtering.

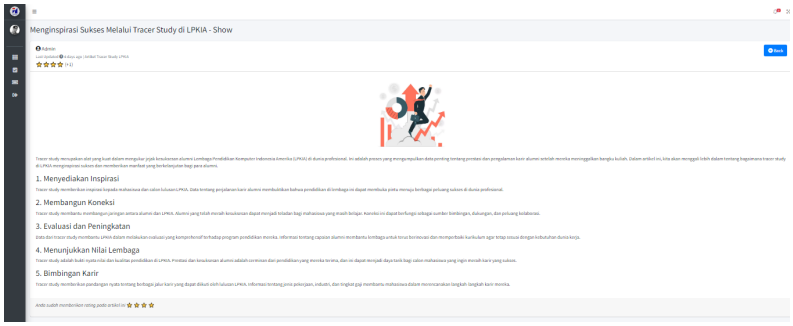


Figure 9 - Article-rating interface

Figure 9 shows the article-rating interface. Each user can rate an article once, and the average of submitted ratings is used as an ordering signal for matched articles.

4. DISCUSSION

The study objective was to maintain useful helpdesk support in two situations: when the available facts are sufficient for a predefined diagnosis and when they are not. The implementation addresses the first situation with forward chaining and the second with article retrieval. This directly responds to the operational problem identified at LPKIA: incomplete or non-matching user information should not necessarily end the support interaction.

Relative to previous smart-helpdesk work, the main contribution is the workflow integration rather than a new algorithm. Forward chaining with certainty factors has been used to provide IT-helpdesk solutions [1], Naive Bayes has been used for complaint classification [2], backward chaining has been applied to e-government helpdesk cases [3], and forward chaining has been implemented in a helpdesk chatbot [8]. The present system adds a separate fallback stage that retrieves support articles when its deterministic rules cannot produce a conclusion.

The recommendation mechanism should also be interpreted according to its actual implementation. Established content-based filtering approaches represent item content and use content attributes, user profiles, or similarity information to identify relevant items [7]–[12]. In this study, the implemented module instead uses problem categories, keyword string matching, and user-rating order. This design is transparent and easy to implement, but it can miss semantically related articles when users employ vocabulary that does not literally occur in the article content. Classical TF-IDF-style term weighting, BM25 ranking, and sentence-embedding retrieval provide progressively stronger baselines that can be used to test whether the fallback mechanism benefits from improved lexical or semantic matching[16]–[18] .

The current verification demonstrates the functioning of the two control-flow paths and the corresponding user interfaces. However, white-box and functional verification alone cannot establish real-world diagnostic accuracy, recommendation relevance, service-response efficiency, or user satisfaction. Those claims require an independent set of helpdesk cases, expert-labeled expected diagnoses, relevance judgments for recommended articles, and appropriate quantitative metrics.

*Corresponding author: fahmi.reza@ekuitas.ac.id

Established recommender-system evaluation literature emphasizes matching evaluation measures to the user task [19], while ranked retrieval quality can be assessed with graded-gain measures such as NDCG [20].
Accordingly, the main contribution of the paper is a practical two-stage architecture for a higher-education helpdesk: rule-based diagnosis is used for known cases, while unresolved cases are redirected to a content-oriented support path. The study should therefore be read as an implementation and workflow-integration contribution, with quantitative effectiveness evaluation remaining an important next step.

5. CONCLUSION

This study was motivated by two related problems at Institut Digital Ekonomi LPKIA: student helpdesk requests are handled manually, and a purely rule-based diagnosis can provide no useful output when reported indications do not completely satisfy a predefined rule. The study objective was therefore to implement an integrated workflow that handles both conditions. The implemented system addresses known cases by applying forward chaining to the knowledge base and returning the corresponding problem code and solution when a complete rule is satisfied. For unresolved cases, the workflow does not force a diagnosis; instead, it redirects the user to support articles filtered by problem category and user-entered keywords and ordered by average user rating. Functional and white-box verification confirms that these two control-flow branches operate as designed.

The main contribution of this paper is a practical two-stage helpdesk architecture that links deterministic rule-based diagnosis with a content-oriented fallback in a single web application. Its contribution lies in workflow integration and continuity of assistance rather than in proposing a new inference or machine-learning algorithm. Accordingly, the present evidence supports the functional behavior of the integrated workflow, but it does not yet establish statistical diagnostic accuracy, recommendation relevance, response-time improvement, or user satisfaction.

6. FUTURE WORK

Future research should move from functional verification to quantitative and external evaluation. First, anonymized helpdesk logs should be collected across multiple academic periods and labeled independently by at least two domain experts, with disagreements reconciled and inter-rater agreement reported. A held-out test set should then be used to report diagnostic accuracy, precision, recall, F1-score, and per-problem performance for E001-E011. Second, article relevance should be judged independently and evaluated at fixed cutoffs using Precision@5, Recall@5, MRR, and NDCG@5, following established principles for recommender-system and ranked-retrieval evaluation [19], [20]. Third, the current category/keyword retrieval baseline should be compared under the same test queries with TF-IDF plus cosine similarity, BM25, and sentence-embedding retrieval [16]–[18], with statistical significance testing where appropriate. Fourth, unresolved production cases should be reviewed periodically to expand, validate, and version the knowledge base. Finally, a prospective user study should measure response time, task completion, user satisfaction, and escalation rate, followed by external validation at another higher-education institution to assess generalizability.

REFERENCES

[1] I. G. N. W. Partha, R. S. Hartati, and L. Jasa, “Rancang Bangun Sistem IT-Helpdesk Untuk Maintenance dan Tutorial Aplikasi Simplebiz Menggunakan Metode Forward Chaining dan Certainty Factor,” Maj. Ilm. Teknol. Elektro, vol. 20, no. 2, pp. 315–326, 2021, doi: 10.24843/MITE.2021.v20i02.P16.

[2] W. D. Suryono, R. Saptono, and W. Wiranto, “Implementasi Pengembangan Smart Helpdesk di UPT TIK UNS Menggunakan Algoritma Naive Bayes Classifier,” in Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2017, p. D-39-D-43. [Online]. Available: https://journal.uii.ac.id/Snati/article/view/8474

[3] P. Hariona, “Sistem Pakar dengan Metode Backward Chaining untuk Optimalisasi Layanan Helpdesk E-Government,” J. Inform. Ekon. Bisnis, vol. 3, no. 2, pp. 66–71, 2021, doi: 10.37034/infeb.v3i2.68.

[4] F. A. Bambang, Ronald, and Gunardi, “Rancang Bangun Chatbot Reservasi Kelas Kursus Pada Stephen Komputer Jambi Dengan Metode Forward Chaining,” J. Process., vol. 16, no. 2, pp. 70–79, 2021, doi: 10.33998/processor.2021.16.2.995.

[5] A. I. Putra and R. R. Santika, “Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering,” Edumatic J. Pendidik. Inform., vol. 4, no. 1, pp. 121–130, 2020, doi: 10.29408/edumatic.v4i1.2162.

[6] D. A. P. D. Pramesti and I. W. Santiyasa, “Penerapan Metode Content-Based Filtering dalam Sistem Rekomendasi Video Game,” J. Nas. Teknol. Inf. dan Apl., vol. 1, no. 1, pp. 229–234, 2022, doi: 10.24843/JNATIA.2022.v01.i01.p28.

*Corresponding author: fahmi.reza@ekuitas.ac.id

[7] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A Hybrid Approach using Collaborative Filtering and Content Based Filtering for Recommender System," J. Phys. Conf. Ser., vol. 1000, no. 1, p. 12101, 2018, doi: 10.1088/1742-6596/1000/1/012101.
[8] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," Expert Syst. Appl., vol. 89, pp. 404–412, 2017, doi: 10.1016/j.eswa.2017.08.008.
[9] A. E. Wijaya and D. Alfian, "Sistem Rekomendasi Laptop Menggunakan Collaborative Filtering dan Content-Based Filtering," J. Comput. Bisnis, vol. 12, no. 1, pp. 11–27, 2018, doi: 10.55281/jcb.v12i1.167.
[10] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Springer, 2011, pp. 73–105. doi: 10.1007/978-0-387-85820-3_3.
[11] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in The Adaptive Web: Methods and Strategies of Web Personalization, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., Springer, 2007, pp. 325–341. doi: 10.1007/978-3-540-72079-9_10.
[12] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Trans. Knowl. Data Eng., vol. 17, no. 6, pp. 734–749, 2005, doi: 10.1109/TKDE.2005.99.
[13] R. Apriliyani, "Comparison of Forward Chaining and Backward Chaining Methods in Expert System Applications for Learning Style Identification," J. Inform. dan Teknol. Komput., vol. 3, no. 2, pp. 84–92, 2022, doi: 10.55377/j-icom.v3i2.6290.
[14] E. Susanto, Gustientiedina, and M. Siddik, "Application of the Forward Chaining Method in Diagnosing Tomato Fever," J. Appl. Bus. Technol., vol. 5, no. 1, pp. 41–50, 2024, doi: 10.35145/jabt.v5i1.143.
[15] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Syst., vol. 46, pp. 109–132, 2013, doi: 10.1016/j.knosys.2013.03.012.
[16] S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," Found. Trends Inf. Retr., vol. 4, no. 1–2, pp. 1–174, 2009, doi: 10.1561/1500000019.
[17] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manag., vol. 24, no. 5, pp. 513–523, 1988, doi: 10.1016/0306-4573(88)90021-0.
[18] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." [Online]. Available: https://github.com/UKPLab/
[19] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 5–53, 2004, doi: 10.1145/963770.963772.
[20] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," ACM Trans. Inf. Syst., vol. 20, no. 4, pp. 422–446, 2002, doi: 10.1145/582415.582418.

*Corresponding author: fahmi.reza@ekuitas.ac.id