Jurnal Kecerdasan Buatan dan Teknologi Informasi
Vol. 5, No. 3, September 2026, pp. 742–752
e-ISSN: 2964-2922, p-ISSN: 2963-6191

Open access

Original Research

Implementation of a Web-Based Rule-Based System for Determining Goat Health Status from Clinical Symptoms and Feeding Patterns

Azka Naufal Elrumi [a], Mulkan Azhari [b*]

[a,b]Department of Information Systems, Faculty of Computer Science and Information Technology, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia

*Corresponding Author: mulkan@umsu.ac.id

ABSTRACT

Goat farmers frequently rely on direct observation and personal experience to assess animal health, while similar clinical signs and limited access to veterinary personnel can delay initial handling. This study develops a web-based expert system using a rule-based system and forward chaining to determine goat health status from clinical symptoms, feeding patterns, and additional observable conditions. Knowledge was represented as 45 IF–THEN rules stored in a MySQL database. The application was implemented with PHP and provides modules for consultation, health-status recommendations, administration of symptoms, feeding patterns, additional conditions, health categories, diagnostic rules, and diagnosis history. The inference process begins with facts selected by the user and matches them against the knowledge base until a conclusion is obtained. The system generates four health-status outcomes: healthy, unhealthy, sick, and indicated mild illness, together with initial handling recommendations. Black-box testing covered the public interface, consultation form, forward-chaining process, result page, and administrator functions. All tested functions produced the expected outputs. The implementation demonstrates that integrating clinical signs and feeding behavior within a transparent rule-based workflow can support structured initial health assessment for goats. The system is intended as decision support and does not replace examination by a veterinarian.

How to Cite:
A. N. Elrumi and M. Azhari, "Implementation of a Web-Based Rule-Based System for Determining Goat Health Status from Clinical Symptoms and Feeding Patterns," *J. Kecerdasan Buatan dan Teknol. Inf.*, vol. 5, no. 3, pp. 742–752, 2026.
Doi: https://doi.org/10.69916/jkbti.v5i3.598

KEYWORDS

Expert System
Forward Chaining
Goat Health
Rule-Based System
Web-Based System

ARTICLE INFO

Received: July 21, 2026
Accepted: August 28, 2026
Available Online: September 18, 2026

1 INTRODUCTION

Goat farming contributes to household income and animal-protein availability, but its productivity depends strongly on the health condition of the animals. Early disturbances are commonly indicated by changes in body temperature, appetite, activity, and physical appearance. Previous expert-system studies and feeding-behavior research show that these observable indicators can support structured preliminary assessment when direct examination is not immediately available [1–3].

Field assessment remains difficult because different disorders may produce similar symptoms and farmers often depend on personal experience. Forward-chaining studies in livestock and goat farming have demonstrated that facts obtained from users can be matched with stored rules to produce consistent conclusions, while animal-based health indicators provide an objective basis for organizing those facts [4–6].

A rule-based expert system represents specialist knowledge as explicit IF–THEN statements. This structure makes the reasoning path traceable and supports maintenance of the knowledge base because symptoms, feeding patterns, conditions, and conclusions can be modified independently. Reviews of goat welfare indicators and previous livestock expert systems also emphasize the value of combining observable behavior with formalized diagnostic rules [7–9].

The theoretical basis of the system follows knowledge-based artificial intelligence, decision-support architecture, and data-driven forward chaining. In addition to clinical signs, feeding management is included because reduced appetite, refusal of feed, and changes in consumption frequently accompany declining animal condition [10–13]. Earlier goat-disease applications mainly focused on identifying a specific disease from clinical symptoms, including web-based and forward-chaining implementations. These studies established the feasibility of rule-based diagnosis, but the integration of feeding patterns and additional observable conditions into a general health-status output remains limited [14, 15].

The primary novelty of this study lies in the paradigm shift from specific disease diagnosis to a holistic health-status assessment that simultaneously integrates three dimensions of observation: clinical symptoms, feeding patterns, and additional conditions. Unlike existing goat expert systems that predominantly focus on identifying specific disease names based solely on clinical signs, this research formulates a knowledge base of 45 explicit IF–THEN rules that map combinations of feeding behavior and physical conditions into four distinct health-status categories (healthy, unhealthy, sick, and indicated mild illness), complete with initial handling recommendations. This approach offers significant practical value for farmers in the field, as they are often more capable of recognizing changes in appetite and general animal behavior than independently diagnosing specific diseases. Consequently, this system serves as a more inclusive, transparent, and easily maintainable preliminary decision-support tool, bridging the gap between layperson observation and the need for structured livestock health management prior to professional veterinary intervention.

This study therefore implements a web-based rule-based system that determines goat health status from clinical symptoms, feeding patterns, and additional conditions. The objectives are to formalize the knowledge into 45 diagnostic rules, execute the rules through forward chaining, provide initial handling recommendations, and verify the public and administrative functions through black-box testing. The output is intended as an initial decision-support result rather than a substitute for veterinary examination.

2 RESEARCH METHODS

2.1 Research Design and Knowledge Source

The study used an applied system-development design. The research stages comprised problem analysis, literature-based knowledge acquisition, identification of clinical and feeding variables, construction of the rule base, database and interface design, implementation of the web application, and functional testing. The knowledge source consisted of the scientific references and rule combinations documented in the thesis, with the system focused on observable conditions that can be entered by farmers.

2.2 Input Variables and Knowledge Representation

The diagnostic facts were organized into three groups. Clinical symptoms represented body-temperature conditions; feeding patterns represented normal appetite, reduced appetite, or refusal of feed; and additional conditions represented active behavior, weakness, rapid breathing, passive behavior, or shivering. The conclusions comprised healthy, unhealthy, sick, and indicated mild illness, each linked to an initial handling recommendation.

The knowledge base contained 45 IF–THEN rules. Every rule combined one clinical-symptom value, one feeding-pattern value, and one additional condition with a health-status conclusion and recommendation. The rules were stored separately from consultation facts so that an administrator could revise the knowledge base without changing the consultation interface or inference procedure.

2.3 Rule-Based Inference and Forward Chaining

For rule R_i, the antecedent consists of a clinical symptom G_j, a feeding pattern P_k, and an additional condition K_l. When all antecedent facts are satisfied, the rule produces health status S_m:

$$R_i : (G_j \wedge P_k \wedge K_l) \rightarrow S_m \tag{1}$$

Forward chaining begins with the facts submitted through the consultation form. The inference engine scans the rule base, selects rules whose antecedents are contained in working memory, and adds the resulting health status and recommendation to the current fact set:

$$WM_{t+1} = WM_t \cup \{S_m \mid \text{antecedent}(R_i) \subseteq WM_t\} \tag{2}$$

The process stops when a matching conclusion has been obtained or no additional rule can be executed. Because the rule conditions are explicit, the system can display the selected facts, the resulting status, and the recommendation that corresponds to the matched rule.

2.4 Web-Based Architecture and Database

The application was implemented with PHP, HTML, CSS, JavaScript, and MySQL. A Model–View–Controller structure separated data access, application logic, and interface presentation. The relational database stored administrator accounts, clinical symptoms, feeding patterns, additional conditions, health statuses, diagnostic rules, and diagnosis history. Public users could access information and consultation functions, whereas authenticated administrators could manage the knowledge base.

2.5 Functional Testing

Functional verification used black-box testing. The scenarios covered the home, consultation, method, and information pages; selection of all consultation variables; execution of forward chaining; display of diagnosis results; administrator authentication; knowledge-base operations; and diagnosis history. Each scenario was assessed by comparing the observed response with the expected output.

2.6 System Modeling and Interface Design

a Algorithm, Actors, and Data Structure

The system design translated the inference procedure into an operational flow, defined the interaction of public users and administrators, and organized the database relationships required by the rule base.

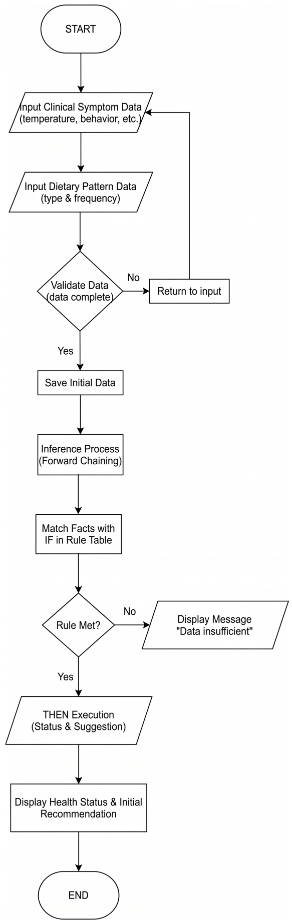

Figure 1. Forward-chaining algorithm flowchart

Figure 1 shows the complete diagnostic flow. Users enter clinical symptoms, feeding patterns, and additional conditions; the system validates the input, stores the facts, executes forward chaining, and returns a health status with an initial recommendation. An incomplete input returns the user to the consultation form.

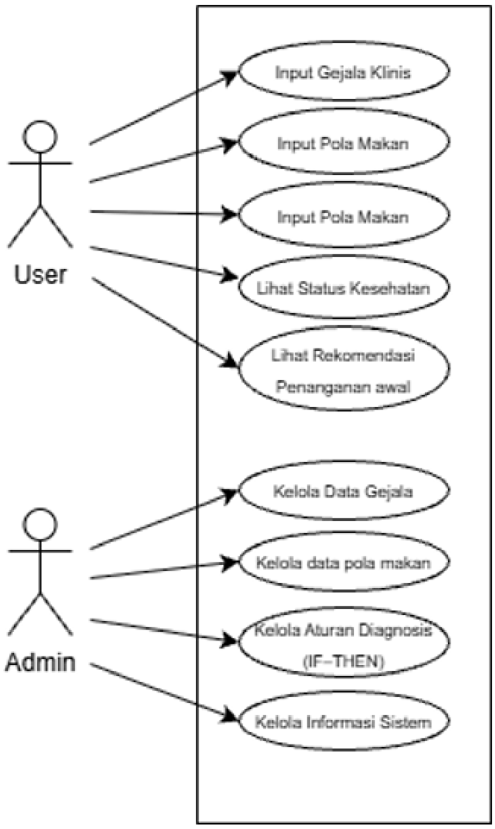

Figure 2. Use-case diagram of the goat-health expert system

Figure 2 identifies two actors. Public users access the consultation and view the resulting status and recommendation. Administrators authenticate before managing symptoms, feeding patterns, additional conditions, health-status categories, diagnostic rules, and system information.

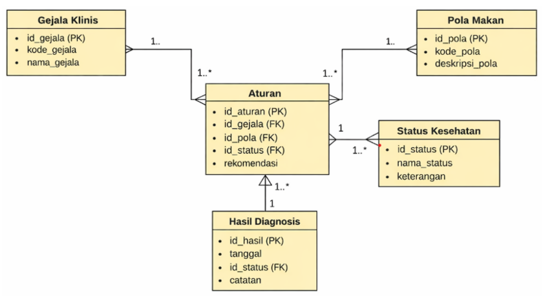

Figure 3. Entity-relationship diagram of the system database

Figure 3 presents the relational structure. Clinical symptoms, feeding patterns, and health statuses are referenced by the rule table, while diagnosis results store the selected conclusion and consultation record. This separation supports structured rule maintenance and historical reporting.

3 RESULTS AND DISCUSSION

3.1 Public Information and Diagnosis Output

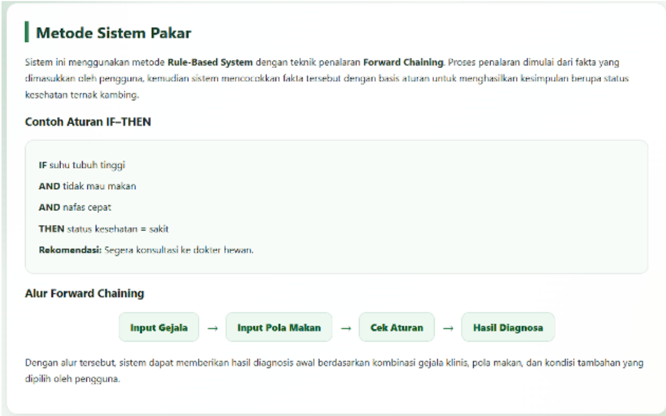

Figure 4. Implemented method-information page

Figure 4 illustrates the initial user-interface design of the web-based expert system, serving as the primary landing page for both public users and administrators. This home-page interface is strategically designed to provide intuitive and direct navigation to the systems core modules, including the health consultation form, method information, general system information, and the secure administrator login portal. Furthermore, the interface prominently features a summarized visual overview of the systems diagnostic workflow. This summary is crucial for enhancing the user experience, as it allows farmers and laypersons to clearly understand the sequential steps of the forward-chaining process before they begin inputting the animals clinical symptoms, feeding patterns, and additional conditions. By presenting a clean, structured, and user-friendly layout, Figure 4 demonstrates the systems commitment to accessibility and transparency, ensuring that users can confidently navigate the application and comprehend the diagnostic procedure without requiring prior technical or veterinary expertise.

Figure 5. Implemented diagnosis-result page

Figure 5 presents the implemented consultation form design, which serves as the primary data entry interface for the goat health expert system. This form is structured around three mandatory input fields that correspond to the key diagnostic variables identified in the knowledge acquisition phase: clinical symptoms (representing body temperature conditions), feeding patterns (indicating appetite levels such as normal, reduced, or refusal to eat), and additional conditions (capturing observable behaviors including active behavior, weakness, rapid breathing, passive behavior, or shivering). Each field is implemented as a dropdown selection menu to ensure data consistency and prevent invalid inputs that could compromise the inference process. The form incorporates client-side validation mechanisms that prevent submission until all three required fields are completed, thereby ensuring that the forward chaining engine receives a complete set of facts necessary for rule matching. This design choice reflects the system's commitment to data integrity and diagnostic accuracy, as incomplete information could lead to ambiguous or incorrect health status determinations. Upon successful completion and submission, the selected values are stored as initial facts in the working memory and passed to the inference engine, which then executes the forward chaining algorithm against the 45-rule knowledge base. The clean, intuitive layout minimizes user errors and accommodates farmers with varying levels of technical proficiency, while the clear labeling of each field ensures that users can accurately translate their observations of the goat's condition into structured diagnostic data. This consultation form represents the critical interface between human observation and automated reasoning, serving as the foundation upon which the entire expert system's diagnostic capability is built.

3.2 Knowledge-Base Administration

To ensure the system remains dynamic, scalable, and easily maintainable, the knowledge base is strictly separated from the core inference engine. This architectural choice allows administrators to update, modify, or expand the diagnostic variables without altering the underlying code or the public user interface. The management of these foundational elements is facilitated through a set of dedicated administrative forms designed for structured data entry. Figure 6 illustrates the master data entry interfaces created for this purpose, specifically focusing on the three input variables and the primary output categories that drive the forward chaining process.

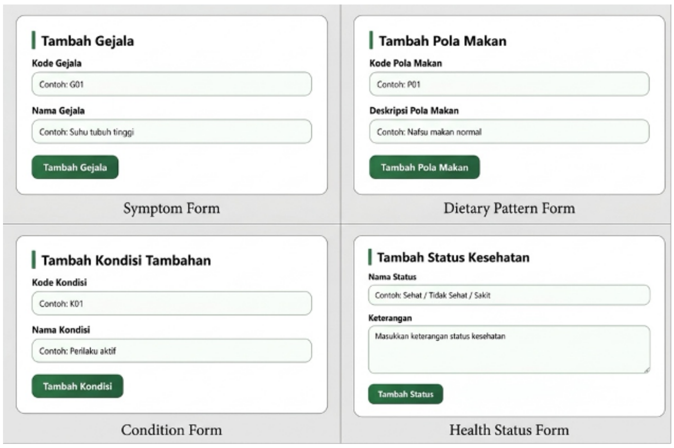

Figure 6. Administrator knowledge-base management forms

As depicted in Figure 6, the administrative dashboard is organized into four distinct modules, each corresponding to a critical component of the diagnostic logic. The 'Symptom Form' (top-left) and 'Feeding Pattern Form' (top-right) allow administrators to define clinical signs (e.g., body temperature conditions) and feeding behaviors (e.g., normal or reduced appetite) by assigning unique codes and descriptive names. The 'Condition Form' (bottom-left) captures supplementary observable states, such as active behavior or shivering, while the 'Health Status Form' (bottom-right) configures the system's output categories (e.g., Healthy, Sick) along with their explanatory notes and recommendations. The data entered into these structured forms directly populates the dropdown menus in the public consultation interface and serves as the foundational building blocks for the 45 IF-THEN diagnostic rules. This modular data entry approach guarantees data consistency, prevents invalid inputs during the public consultation phase, and significantly simplifies the ongoing maintenance of the expert system's knowledge base.

3.3 Rule Management and Diagnosis History

Following the establishment of master data for symptoms, feeding patterns, conditions, and health status categories, the next critical step in building the expert system's knowledge base is the creation of diagnostic rules that link these variables through logical IF-THEN relationships. The rule management interface enables administrators to construct the 45 diagnostic rules that form the core reasoning mechanism of the forward chaining inference engine. This interface provides a structured form where administrators can combine one clinical symptom, one feeding pattern, and one additional condition as antecedents, then assign a corresponding health status conclusion and initial handling recommendation as the consequent. Figure 7 presents the implemented rule entry form that facilitates this knowledge acquisition process, allowing administrators to systematically encode veterinary expertise into executable diagnostic rules without requiring direct database manipulation or programming knowledge.

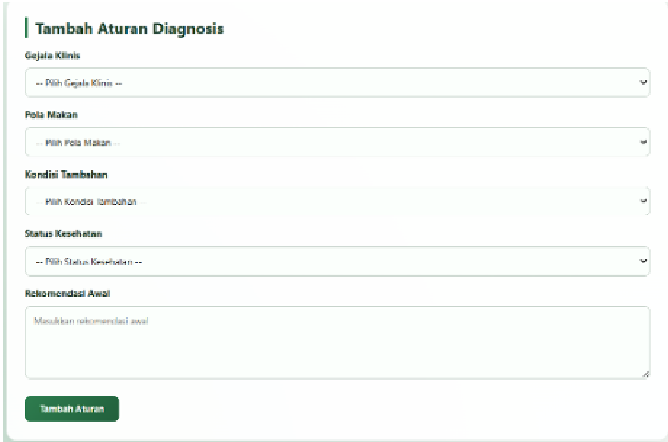

Figure 7. Diagnostic-rule management page

As illustrated in Figure 7, the diagnosis rule addition form employs a user-friendly interface consisting of five primary input components: dropdown selectors for clinical symptoms, feeding patterns, additional conditions, and health status, along with a text area for entering initial recommendations. The dropdown menus are dynamically populated from the master data tables previously configured through the forms shown in Figure 6, ensuring data consistency and preventing the creation of rules with invalid or non-existent references. When an administrator completes this form and submits the rule, the system stores the combination as a structured IF-THEN statement in the `diagnostic_rules` table of the MySQL database, where it becomes immediately available for the forward chaining inference process. This rule entry mechanism is fundamental to the system's functionality, as each of the 45 rules represents a specific diagnostic scenario that maps observable goat conditions to health status determinations. The separation of rule management from the inference engine allows domain experts or authorized administrators to refine, expand, or modify the knowledge base based on new veterinary insights or field observations without requiring system redeployment, thereby ensuring the expert system remains current and adaptable to evolving diagnostic requirements.

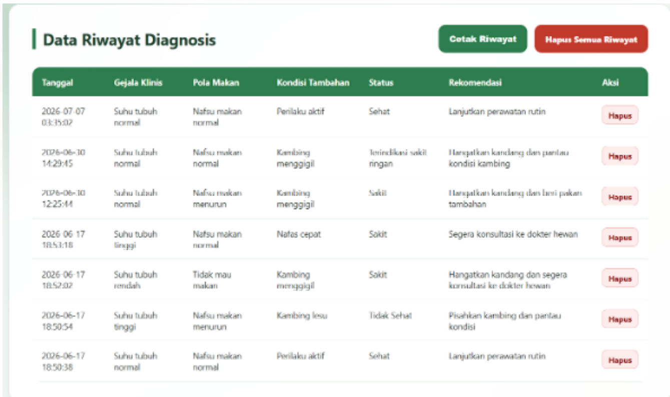

Figure 8. Diagnosis-history page

Figure 8 presents the implemented diagnosis history page, which serves as a comprehensive administrative log of all consultations performed within the expert system. The interface displays a structured table that records essential details for each diagnosis, including the consultation date and time, the selected clinical symptoms, feeding patterns, and additional conditions, as well as the resulting health status and initial handling recommendation generated by the forward chaining process. At the top of the page, administrative controls such as "Print History" and "Delete All History" allow for efficient data management and reporting. Additionally, individual records can be removed using the "Delete" button in the action column. This historical storage is crucial for system evaluation, as it provides administrators with a tangible record of system usage and enables the review of rule outcomes generated from various condition combinations. By maintaining this log, the system not only supports administrative oversight but also facilitates future validation of the 45 diagnostic rules against real-world field data, ensuring the knowledge base remains accurate and relevant.

3.4 Black-Box Testing

To ensure the reliability and functional correctness of the developed web-based expert system, a comprehensive black-box testing method was conducted. Black-box testing focuses on evaluating the system's functionality based on the specified requirements without examining the internal code structure or logic [16–20]. The testing process was designed to verify that all user interactions, data inputs, and system outputs operate as intended. The test scenarios were systematically divided into two main categories: public user functions and administrator functions. Public user testing covered the navigation, consultation form input, execution of the forward chaining inference engine, and the display of diagnosis results. Administrator testing verified authentication, knowledge-base management (adding, editing, and deleting master data and rules), and the diagnosis history log. For each scenario, the expected result was defined prior to testing and compared against the actual observed result to determine the system's validity. Table 1 summarizes the twelve representative test scenarios and their corresponding outcomes.

Table 1. Black-box testing results

No.	Tested Feature	Test Scenario	Expected Result	Observed Result	Status
1	Home Page	Open the public home page	System information and navigation are displayed	Successful	Valid
2	Consultation Page	Open the consultation menu	The consultation form is displayed	Successful	Valid
3	Method Page	Open the method menu	Rule-based and forward-chaining info displayed	Successful	Valid
4	About Page	Open the about menu	System purpose and limitations are displayed	Successful	Valid
5	Clinical Symptom	Select a clinical-symptom value	The selected symptom is accepted	Successful	Valid
6	Feeding Pattern	Select a feeding-pattern value	The selected feeding pattern is accepted	Successful	Valid
7	Additional Condition	Select an additional condition	The selected condition is accepted	Successful	Valid
8	Forward Chaining	Execute diagnosis with complete input	A matching rule and conclusion are generated	Successful	Valid
9	Diagnosis Result	Open the diagnosis output	Status, selected facts, and recommendation shown	Successful	Valid
10	Admin Login	Enter valid administrator credentials	The administrator dashboard is displayed	Successful	Valid
11	Knowledge-Base Data	Add, edit, and delete master data and rules	The requested data changes are stored	Successful	Valid
12	Diagnosis History	Open the diagnosis-history page	Stored consultation results are displayed	Successful	Valid

As presented in Table 1, the black-box testing results demonstrate that all twelve tested features and scenarios produced the expected outputs, achieving a 100% success rate and classified as "Valid." The public interface successfully displayed all informational pages and correctly accepted user inputs for clinical symptoms, feeding patterns, and additional conditions. Crucially, the forward chaining inference engine accurately processed the complete set of input facts, matched them against the 45 stored IF–THEN rules, and generated the correct health status conclusions along with the corresponding initial handling recommendations. On the administrative side, the system successfully validated user credentials, securely restricted access to authorized personnel, and allowed seamless manipulation of the knowledge base and diagnosis history. These results confirm that the system's architecture, from the user interface to the database and inference logic, is fully functional and integrated. While the testing validates the system's operational correctness, it is important to note that the accuracy of the diagnostic conclusions ultimately depends on the completeness of the rule base and the accuracy of the observations entered by the user.

3.5 Discussion

The implementation and functional testing of the web-based rule-based expert system for determining goat health status have yielded significant insights regarding the application of artificial intelligence in preliminary livestock health assessment. The successful deployment of 45 IF–THEN rules processed through a forward-chaining inference engine demonstrates that observable indicators, namely clinical symptoms, feeding patterns, and additional physical conditions, can be effectively formalized into a structured, transparent diagnostic framework.

The systems architecture, which strictly separates the knowledge base from the inference engine and the user interface, aligns with established best practices in expert system design [10, 12]. This modular decoupling is particularly advantageous in agricultural contexts, where veterinary knowledge evolves and diagnostic criteria must adapt to regional disease patterns. By allowing administrators to modify symptoms, feeding patterns, conditions, and diagnostic rules via a graphical interface without altering the underlying source code, the system overcomes a common limitation identified in earlier livestock expert systems that required direct database manipulation or programming expertise for knowledge base updates [2, 8].

A primary novelty of this study is the integration of feeding patterns as a core diagnostic variable alongside clinical symptoms. Previous goat-disease expert systems have predominantly relied on clinical signs alone to identify specific diseases [1,15]. However, animal welfare and nutrition research indicates that changes in appetite and feeding behavior are often the earliest and most reliable indicators of declining health in goats [3, 11]. By incorporating this behavioral dimension, the system provides a more holistic health assessment that better reflects the multifaceted nature of on-farm animal monitoring, consistent with contemporary animal-based welfare indicators that emphasize combining multiple observable parameters for accurate status determination [6, 7].

The forward-chaining inference method proved highly suitable for this diagnostic task, as it naturally mirrors the data-driven cognitive process employed by veterinarians and experienced farmers who gather observable facts before reaching a conclusion. The black-box testing results, which demonstrated 100% validity across all twelve test scenarios, confirm that the inference engine correctly maps user-provided facts to the rule base. However, it must be acknowledged that the diagnostic accuracy is fundamentally constrained by the completeness of the 45 stored rules. While this rule set covers common health scenarios derived from the foundational literature, it may not encompass all possible disease presentations or rare conditions in diverse farming environments.

The four-tier health status classification (healthy, unhealthy, sick, and indicated mild illness) provides a practical gradient that is more nuanced than the binary healthy/sick classifications found in some earlier systems [4, 5]. This granularity allows the system to distinguish between animals requiring immediate veterinary intervention and those that may only need enhanced monitoring or minor dietary adjustments. Coupled with specific initial handling recommendations, this design enhances the systems practical utility by providing actionable guidance rather than mere diagnostic labels.

Furthermore, the web-based implementation offers distinct advantages over standalone or mobile-only applications in the context of smallholder farming. Web accessibility eliminates the need for local software installation, reduces device-specific maintenance overhead, and ensures all users interact with the most current version of the knowledge base. The systems transparency, achieved through the explicit display of the matched IF–THEN rules and the forward-chaining sequence, directly addresses common criticisms of expert systems as opaque "black boxes" [13]. By allowing users to trace which specific symptoms and conditions led to a particular conclusion, the system builds user trust and serves an educational function, potentially improving farmers' observational skills over time.

Despite these strengths, several limitations must be explicitly acknowledged. First, the system is designed strictly as a decision-support tool and does not replace professional veterinary examination, as emphasized by the systems disclaimer. The accuracy of the output is highly dependent on the farmers ability to correctly observe and report conditions; misinterpretation of signs (e.g., estimating body temperature without a thermometer) could lead to inaccurate diagnoses. Second, the current knowledge base has not yet been validated through extensive, large-scale field trials involving practicing veterinarians. Third, the system relies on qualitative categorical inputs rather than quantitative measurements (e.g., exact body temperature values or weight loss percentages), which could further enhance diagnostic precision. Finally, the current administrator authentication mechanism is basic and would require enhancement through password hashing and role-based access control for large-scale, multi-user deployment.

In summary, this study demonstrates that integrating clinical symptoms, feeding patterns, and additional observable conditions within a transparent, rule-based workflow can effectively support structured preliminary health assessments for goats. The system successfully meets its objectives of formalizing domain knowledge, executing reliable forward-chaining inference, and providing actionable recommendations. Future work should focus on validating the rule base with veterinary experts, expanding the knowledge base to cover a broader spectrum of diseases, integrating quantitative sensor data, and exploring offline-capable mobile implementations to address connectivity limitations in remote rural areas.

4 CONCLUSION

This study successfully implemented a web-based expert system for determining goat health status by integrating clinical symptoms, feeding patterns, and additional observable conditions. Utilizing a rule-based system with a forward-chaining inference engine, the system processes 45 explicit IF–THEN rules to classify animal health into four distinct categories: healthy, unhealthy, sick, and indicated mild illness, each accompanied by specific initial handling recommendations. The architectural separation of the knowledge base from the inference engine and user interface ensures that the system is highly maintainable and adaptable to future veterinary knowledge updates.

Black-box testing confirmed the functional reliability of the system, with all twelve tested scenarios, spanning public navigation, consultation input, forward-chaining execution, result presentation, and administrative knowledge-base management, producing the expected and valid outputs. The integration of feeding behavior alongside clinical signs provides a more holistic and practical assessment tool for farmers, addressing the limitations of symptom-only diagnostic systems. Ultimately, this system serves as an effective, transparent, and accessible decision-support tool that facilitates structured preliminary health assessments, bridging the gap between farmer observation and professional veterinary intervention. It is emphasized that the system is designed to support, not replace, direct examination by a veterinarian. Future work should focus on validating the rule base with practicing veterinarians, expanding the knowledge base to cover a broader spectrum of regional diseases, enhancing system security, and developing offline-capable mobile implementations to accommodate areas with limited internet connectivity.

DECLARATIONS

Author Contributions:

Azka Naufal Elrumi: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Writing – Original Draft.

Mulkan Azhari: Supervision, Validation, Resources, Data Curation, Writing – Review & Editing.

Funding:

The authors received no financial support for the research, authorship, and/or publication of this article.

Conflicts of Interest:

The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability:

The data and source code that support the findings of this study are available from the corresponding author upon reasonable request.

AI Use Statement:

During the preparation of this work, the authors used AI language models (such as ChatGPT) in order to improve the language, readability, and formatting of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the final content of the published article.

Additional Information:

No additional information is available for this paper.

REFERENCES

[1] P. Alicia, "An expert system using forward chaining to identify goat diseases," *J. Inf. dan Teknol.*, vol. 4, no. 4, pp. 191–197, 2022, doi: 10.37034/jidt.v4i4.216.

[2] M. Alfian, W. Aprianti, H. Rhomadhona, and J. Permadi, "A cattle-disease expert system using rule-based reasoning with forward chaining," *Teknomatika J. Inform. dan Komput.*, vol. 17, no. 2, pp. 22–29, 2024, doi: 10.30989/teknomatika.v17i2.1541.

[3] J. O. Daramola, M. O. Abioja, and O. M. Onagbesan, "Feeding behavior and health indicators in goats," *Trop. Anim. Health Prod.*, vol. 53, no. 4, pp. 1–9, 2021.

[4] M. D. Irawan, A. Widarma, Y. H. Siregar, and Rudi, "Application of forward-backward chaining in an expert system for prevention and treatment of cattle diseases," *J. Teknol. dan Inf.*, vol. 11, no. 1, pp. 14–25, 2021, doi: 10.34010/jati.v11i1.

[5] D. Y. Kasparova, T. Listyorini, and E. Supriyati, "Expert system for diagnosing diseases in goats using the forward chaining method: Case study of farms in Dersoyono Village, Dawe," *Tekmapro*, vol. 20, no. 1, pp. 196–208, 2025.

[6] V. Korelídou, I. Georgopoulou, and C. Papachristoforou, "Health status of Skopelos goats and its impact on milk yield: Animal-based indicators and welfare issues in different farming systems," *Animals*, vol. 15, no. 9, p. 1328, 2025, doi: 10.3390/ani15091328.

[7] A. Minnig et al., "Animal-based indicators for on-farm welfare assessment in goats: A review," *Animals*, vol. 11, no. 11, p. 3138, 2021, doi: 10.3390/ani11113138.

[8] R. Pratama, M. Siregar, and F. Lubis, "A livestock-health expert system based on a rule-based system," *J. Inform.*, vol. 10, no. 2, pp. 101–109, 2022.

[9] I. A. Pribadi, A. Adi Candra, and A. Azriansyah, "An Android-based goat-disease diagnosis expert system using forward chaining," *J. Pepadun*, vol. 2, no. 3, pp. 403–411, 2021, doi: 10.23960/pepadun.v2i3.83.

[10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.

[11] A. Sodiq and P. Yuwono, "Goat-feed management," *J. Ilmu Ternak*, vol. 15, no. 1, pp. 21–30, 2020.

[12] E. Turban, L. Volonino, G. R. Wood, and J. C. Sipior, *Decision Support Systems and Intelligent Systems*, 10th ed. Pearson, 2020.

[13] R. Sharma and A. Gera, "Rule-based expert systems: A review," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 3, pp. 456–463, 2021.

[14] D. Alicia, "An expert system for goat-disease diagnosis using forward chaining," *J. Inf. dan Teknol.*, vol. 4, no. 2, pp. 85–94, 2022.

[15] A. Pribadi, D. Rahmawati, and I. Setiawan, "A web-based goat-disease diagnosis expert system," *J. Pepadun*, vol. 2, no. 3, pp. 145–153, 2021.

[16] D. Triady, I. A. Musdar, and H. Surasa, "Pengujian Blackbox Pada Website WorkerS Menggunakan Metode Equivalence Partitioning," *KHARISMA Tech*, vol. 18, no. 1, pp. 84–98, 2023, doi: 10.55645/kharismatech.v18i1.292.

[17] G. Maulana and Tommy, "Enhancing Web Authentication Transparency: A Unified Framework Combining Bcrypt Hashing, RBAC Enforcement, and Educational Cryptographic Simulation," *Journal of Computer and Technology*, vol. 4, no. 1, pp. 1–4, 2026.

[18] R. S. Lubis and N. I. Syahputri, "Synergizing Historical Similarity and Probabilistic Attributes: A Dual-Engine Machine Learning Model for Subsidized Housing Credit Risk," *Journal of Computer and Technology*, vol. 4, no. 1, pp. 43–52, 2026.

[19] D. Widiastuti, I. Sholikah, C. Rahmawati, and A. O. Ramadhasari, "Enhancing Micro-Retail Inventory Management through Dynamic Reorder Point Algorithm: System Development, ISO/IEC 25010 Performance Evaluation, and Usability Assessment," *Journal of Computer and Technology*, vol. 4, no. 1, pp. 1–4, 2026.

[20] M. Raihan and Tommy, "Majority-Class Collapse in E-Commerce Emotion Mining: A Diagnostic Evaluation of LSTM Under Severe Class Imbalance and Noisy Text," *Journal of Computer and Technology*, vol. 4, no. 1, pp. 1–13, 2026.