Jurnal Media Teknik Elektro dan Komputer
Vol. 02, No.01 (January - June, 2025)
E-ISSN: 3063-2927

Expert System for Diagnosing Respiratory Diseases Using the Forward Chaining Method

Baginda Fikri Haikal[1], Reza Abdillah Siregar[2]

[1]Universitas Islam Negeri Sumatera Utara; bf_haikal07@gmail.com
[2]Univeristas Muhammadiyah Sumatera Utara; rezasiregar98@gmail.com

ABSTRACT

Respiratory diseases are one of the health disorders whose prevalence continues to increase and require rapid and accurate diagnosis to support effective medical treatment. This study aims to develop an expert system for diagnosing respiratory diseases using the forward chaining method. This method was chosen for its ability to perform data-driven reasoning, starting from the facts of the symptoms experienced by the patient, which then trigger certain rules to produce a diagnostic conclusion. The system is designed with a knowledge base containing validated data on symptoms and types of respiratory diseases, as well as an inference engine to process diagnostic rules. The system is implemented using a web-based programming language with database integration that stores information on symptoms, diseases, and reasoning rules. Test results show that the system is capable of providing quick and accurate initial diagnoses based on the symptom data entered, and can serve as a tool for medical professionals and the public in detecting respiratory diseases early. This research is expected to contribute to the development of artificial intelligence-based health technology that supports more effective and efficient medical services.

***Keywords**:* Expert System, Respiratory Diseases, Forward Chaining, Diagnosis, Artificial Intelligence.

Corresponding Author:
Baginda Fikri Haikal
Universitas Islam Negeri Sumatera Utara; bf_haikal07@gmail.com

1. INTRODUCTION

The rapid development of information and communication technology (ICT) has had a significant impact on various areas of life, including the health sector. One of the results of implementing technology in the world of health is the development of expert systems, which are computer-based systems designed to mimic the decision-making abilities of experts. This system enables the diagnosis and problem-solving process to be carried out more quickly, efficiently, and accurately, while reducing dependence on the physical presence of experts (Anggara et al., 2025).

Respiratory diseases are one of the most common health problems encountered in society and can affect anyone regardless of age. These diseases encompass a wide range of conditions, from mild colds to serious infections such as pneumonia and tuberculosis, which can be fatal if not treated properly (Wenda et al., 2023). Although the management of respiratory diseases requires medical expertise, not all communities have direct access to specialist doctors, especially in remote areas or those with limited healthcare facilities. This highlights the need for a widely accessible medical decision support system that can assist in the initial diagnosis process.

Expert systems can be used as a potential solution to address problems. An expert system is software that models the knowledge and problem-solving abilities of an expert, using a knowledge base and inference engine to generate rule-based decisions. One of the inference techniques commonly used in expert systems is the forward chaining method, which is a data-driven approach that processes known facts to generate conclusions through the application of logical rules in a step-by-step manner (Syamsudin & Sudarsono, 2022).

Based on research conducted by (Mauliza et al., 2022) discussing the development of an expert system for early diagnosis of infectious diseases in children using the forward chaining method. This study aims to detect diseases based on symptoms such as high fever, cough, diarrhea, and abdominal pain, utilizing 17 rules (graphs) in the inference process. This research implements a trace tree model to trace the facts and symptoms entered by the user, enabling the system to identify diseases such as Hepatitis A and provide diagnostic results, solution recommendations, and prevention measures. The results indicate that the system is capable of displaying accurate diagnostic information based on the symptoms entered.

In addition, research conducted by (Mardian et al., 2023) This study aims to develop an expert system for diagnosing lung diseases and providing relevant treatment information. Through a comparative analysis of several inference methods, the combination of forward chaining and certain factor proved to provide the highest level of accuracy. The results of testing through software, user questionnaires, and expert validation show that the system has a diagnosis accuracy rate of 93% and is considered suitable for use by the public.

Research conducted by (Elkana & Kuswanto, 2023) his study developed an information system to support the diagnosis and treatment recommendation process in outpatient services at hospitals. The system was designed to address the limitations of health records, which have been considered suboptimal. Using the forward chaining method, the system assists medical personnel in identifying diseases and providing medication recommendations quickly and accurately. The implementation results show that the application is capable of supporting the service process from registration to medication dispensing, and has been successfully tested using black-box testing with positive results from respondents.

However, based on preliminary studies and literature reviews, the application of expert systems for respiratory disease diagnosis is still limited, especially in the form of Java-based desktop applications integrated with modern database systems such as MySQL. Most previous studies have focused only on web-based systems or have only modeled inference logic without implementing functional applications that are ready for use. Therefore, this study was conducted to address this gap by designing and implementing an expert system capable of automatically performing initial diagnosis of respiratory diseases using the forward chaining method, built using the Java programming language, utilizing a MySQL database, and running on a desktop environment based on JDK 17.

This study aims to develop an expert system that can help users recognize symptoms and identify types of respiratory diseases independently with the support of structured inference logic. In addition to providing practical solutions for the community, this research is also expected to contribute to the development of a replicable, scalable, and integrable desktop-based expert system that can be further integrated into modern health information systems. Thus, this research not only offers technical benefits but also provides initial diagnostic solutions that support a more inclusive and adaptive healthcare system in line with technological advancements.

2. LITERATURE REVIEW

2.1. Expert System

Expert systems are a branch of artificial intelligence (AI) that aim to adopt the knowledge and experience of an expert into a computer system. These systems are used to solve problems that can generally only be solved by human experts, such as disease diagnosis, risk analysis, or technical recommendations. The main components of an expert system include: (1) a knowledge base, which is a collection of facts and rules that form the foundation of the system; (2) an inference engine, which is a reasoning engine that draws conclusions based on the given facts; and (3) a user interface that enables interaction between the user and the system. (Nengsih & Putra, 2020). A good expert system not only stores knowledge, but can also adjust its inference process to new facts, justify its decisions, and explain the reasons behind its recommendations (Putra et al., 2024). In the context of healthcare, expert systems provide added value in the form of efficient diagnosis and access to medical information that was previously only available through direct consultation with medical professionals.

2.2. Metode Forward Chaining dalam Sistem Pakar

Forward chaining is an inference method that starts from initial facts and uses if-then logic rules to deduce new information until a conclusion is reached. This technique is known as a data-driven approach because reasoning starts from known conditions (facts/symptoms) to find the ultimate goal (diagnosis/solution). This differs from backward chaining, which starts from a hypothesis and traces back to the facts for validation. A forward chaining flowchart can be illustrated in Figure 1.

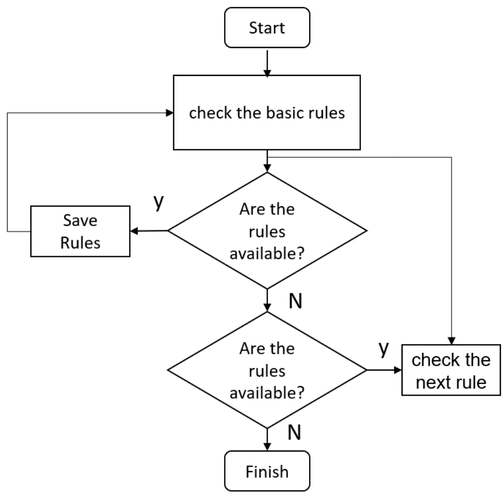

Figure 1. Forward Chaining Flowchart

The Forward Chaining method describes a reasoning technique in expert systems that works with a data-driven approach, where the process begins with known facts and gradually uses them to trigger rules stored in the knowledge base to produce conclusions. In the initial stage, initial facts are entered into working memory as the foundation for the inference process. The knowledge base contains a set of rules representing the relationships between facts and conclusions, while the inference engine checks whether the conditions of the rules are satisfied by the facts in working memory. If the conditions of a rule are satisfied, the action specified by that rule is executed, resulting in new facts that are then added to working memory. This process occurs iteratively, where each new fact has the potential to trigger other rules, until no further rules can be applied or until the final conclusion is reached. Facts represent the initial data that serves as the starting point for reasoning, rules act as the logical bridge to conclusions, and conclusions are the final results obtained after all reasoning steps are completed.

2.3. Diagnosis of Respiratory Diseases with the Aid of Technology

Respiratory diseases such as asthma, bronchitis, pneumonia, and tuberculosis are common diseases with symptoms that often overlap. This makes it difficult for the general public to distinguish between different types of diseases based on the symptoms they experience. Therefore, information technology can be used to develop an early diagnosis tool that can guide patients in recognizing their health conditions.

Previous research has examined the implementation of web-based expert systems for diagnosing respiratory diseases, but these systems do not provide an interface that can be used offline or via a desktop platform. Another study by (Wenda et al., 2023) highlights the importance of integrating symptom data with inference logic capable of processing various symptom combinations adaptively. This underscores the need for an expert system that is not only accurate but also flexible and easily accessible to the general public, even without an internet connection.

2.4. Research Gap

Based on the literature reviewed, most expert systems for diagnosing respiratory diseases are still web-based, with limitations in accessibility in conditions without an internet connection. In addition, many studies have not explored in depth the use of forward chaining methods in complex and dynamic rule structures. The lack of desktop applications for diagnosing diseases that use Java and MySQL-based inference approaches also indicates that there is room for new contributions.

This study aims to address these gaps by developing a desktop-based expert system for respiratory disease diagnosis, utilizing the forward chaining method, built with Java, and integrated with MySQL. This system is expected to be used independently by the general public and serve as an initial model for the development of broader artificial intelligence-based healthcare systems.

3. METHODS

3.1. Research Methods

This study uses a research and development (R&D) approach with a focus on the design and implementation of a rule-based expert system for diagnosing respiratory diseases. The development model used is based on the Waterfall approach, which consists of five main stages: (1) requirements analysis, (2) system design, (3) knowledge base development, (4) system implementation, and (5) testing and evaluation (Rifki et al., 2023). The main objective of this research is to produce an expert system that can assist users in performing initial diagnosis of respiratory diseases based on symptoms inputted using the forward chaining inference method.

3.2. Data Collection Techniques

Data collection was carried out using two main methods (Rifki & Syamia, 2024):

1. Expert Interviews: Information was collected from pulmonologists to obtain data on symptoms, types of diseases, and treatment recommendations.
2. Literature Study: Additional data was obtained from medical journals, clinical diagnosis books, and previous research related to health expert systems.

All of this data was compiled into a knowledge base consisting of a list of symptoms (facts), a list of diseases (hypotheses), and if–then rules that describe the process of drawing conclusions from symptoms to diagnoses.

3.3. Knowledge Base Development

Based on the acquisition of knowledge obtained from the literature process and also validation. So there are 8 diseases with 30 symptoms. The list of diseases and symptoms can be seen in table 2 and table 3. From these diseases and symptoms, the relationship between diseases and symptoms is generated which can be seen in Table 3. The design of this respiratory disease diagnosis application uses a Flowchart design.

Table 1. List of Respiratory Diseases

No	Disease Code	Name of Disease
1.	P1	Influenza
2.	P2	Emphysema
3.	P3	Pneumonia

No	Disease Code	Name of Disease
4.	P4	Asthma
5.	P5	Bronchiolitis
6.	P6	Bronchitis
7.	P7	Sinusitas
8.	P8	*Common Cold*

Table 2. List of Symptoms of Respiratory Diseases

No	Symptom Code	Symptom Name	No	Symptom Code	Symptom Name
1	G1	Fever	16	G16	Trouble sleeping
2	G2	Cough	17	G17	Chest pain
3	G3	Nasal congestion	18	G18	Reduced sense of smell
4	G4	Headache	19	G19	Facial pain
5	G5	Sore throat	20	G20	Breath
6	G6	Hard to swallow	21	G21	Toothache
7	G7	Limp	22	G22	Disturbances of consciousness
8	G8	Shortness of breath	23	G23	Sweating and chilling
9	G9	Sneeze	24	G24	Cough with thick phlegm
10	G10	Rapid breathing	25	G25	Diarrhea
11	G11	Sound of rapid breathing or wheezing	26	G26	Nausea or vomiting
12	G12	Reduced appetite	27	G27	Lips and nails turn blue
13	G13	Hoarseness	28	G28	Runny nose
14	G14	Parents have a history of asthma	29	G29	Ear pain
15	G15	Trouble sleeping	30	G30	Watery eyes

Meanwhile, information about disease codes (Disease Code) and a list of symptom codes (Symptom Code) associated with each disease. The table structure consists of Disease Code, which is a unique code representing a disease (e.g., P1, P2, P3, up to P8). As well as Symptom Code – symptom codes (G1, G2, G3, and so on) related to that disease. Each disease may have one or more associated symptom codes. For example, if disease P1 has symptoms G1, G2, G3, G4, C5, C6, G7, G8, G9, G10, G11, G12, G13, G17, G22, and G29, while disease P2 only has four symptoms, namely G7, G8, G21, and G26. This relationship is important for the diagnostic process, particularly in expert systems or chaining methods, as it enables matching between the symptoms experienced by the patient and the possible diseases they may be suffering from.

Table 3. List of Disease and Symptoms Relationships

No	Disease Code	Symptom Code
1	P1	G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, G12, G13, G17, G22, G29
2	P2	G7, G8. G21, G26

No	Disease Code	Symptom Code
3	P3	G1, G2, G4, G8, G10, G12, G16, G23, G24, G25
4	P4	G7, G8, G11, G13, G14, G21
5	P5	G1, G2, G7, G8, G10, G11, G12, G15, G30
6	P6	G1, G2, G3, G7, G8, G11, G23, G27
7	P7	G1, G2, G3, G4, G8, G9, G10, G13, G15, G17, G18, G19, G20, G28
8	P8	G1, G2, G3, G4, G9, G13, G17, G27, G29

Table 4. List of Symptom Relationships and Conclusions

Symptom Code	Information	Conclusion
G1	P1 = 1 P3 = 1 P5 = 1 P6 = 1 P7 = 1 P8 = 1	Influenza or Pneumonia or Bronchiolitis or Bronchitis or Sinusity or Common Cold
G4	P1 = 2 P3 = 2 P7 = 2 P8 = 2	Influenza or Pneumonia or Sinusitas or *Common Cold*
G10	P1 = 3 P3 = 3 P7 = 3	Influenza or Pneumonia or Sinusitis
G11	P1 = 4	Influenza
G22	P1 = 5	Influenza
G29	P1 = 6	Influenza
G21	P1 = 6	Influenza
G24	P1 = 6	Influenza
G15	P1 = 6	Influenza
G28	P1 = 6	Influenza

3.4. Algorithm Used

Forward chaining is also known as bottom-up reasoning, because the reasoning process moves from facts at the bottom level to conclusions at the top level based on those facts. Bottom-up reasoning in expert systems can be considered equivalent to conventional bottom-up programming (Sukma & Petrus, 2020). Forward Chaining can be interpreted as a method used by the system to find new information based on known facts (Wajidi & Nur, 2021).

In storing knowledge data, this application uses a third-party application, namely the MySQL application. MySQL is one of the database management system (DBMS) applications that is widely used in data storage both for local (offline) and post-hosting (online) storage. The database is a collection of information on data that has been processed over a long period of time, often for many years . The use of third-party database applications is quite burdensome for ordinary users in running this application, not many desktop users understand well to install MySQL DBMS, therefore the developers really hope that this application can be further developed with a database that uses I/O

Stream which is directly provided by the programming language used, in this case the java programming language. The use of I/O Stream will greatly make it easier for users to operate the application, but on the other hand, of course, it will make it difficult for developers to process the database later, but then I/O Stream will still be a better choice than using a third-party DBMS application.

3.5. Research Framework

Here are some stages of work carried out before conducting the research stage.

1. Symptom data collection is the process by which the user determines what symptoms he or she feels (Ware et al., 2020). At this stage, the symptoms that have been acquired in the knowledge base are data that have been validated previously in the research process.
2. Following the flow of the decision tree (rules) is a stage where questions will be given to users based on the symptoms that have been compiled in the stage of preparing a knowledge base in the application development process.
3. Seeing the most symptom values is a stage where the determination of the diagnosis results (diseases) is taken from the symptom values that are most chosen by users, this method is very good if the user only experiences a few symptoms in a disease, so that the decision tree is not damaged. However, in this way, users can get results for more than 1 disease if the number of symptoms selected is the same.
4. After the process of calculating the most symptom values, the results of the diagnosis (disease) will be obtained.
5. The conclusion of dignosa is a description of the disease that the user may suffer.

After the results are obtained, the results of the diagnosis and conclusion will be displayed on the user's screen.

4. RESULTS AND DISCUSSION

4.1. Result

Flowchart diagram of the application framework for Respiratory Disease Diagnosis with the Forward Chaining Method. This diagram illustrates the logical flow of the expert system from the initial process to the diagnosis determination process, through the stages of managing symptom data, disease data, and diagnosis rules. The following is an explanation of each element in the diagram:

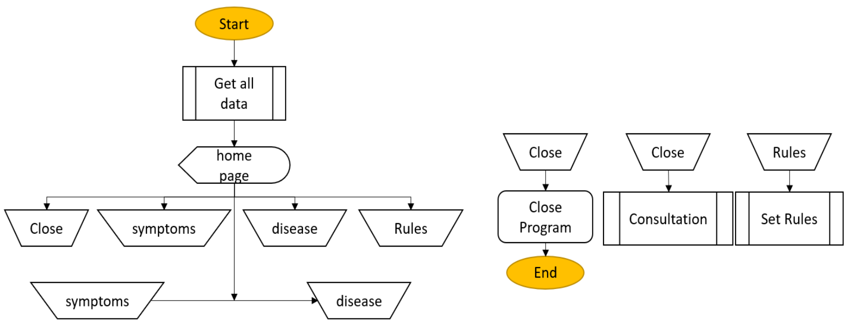

Figure 2. Flowchart Main Process

Figure 1 represents a flowchart of the workflow of an expert system based on symptoms, diseases, and rules used for consultation and data management processes. The diagram is divided into two main parts: the main process (left) and the derivative or follow-up process (right). On the left side, the process begins at the Start node, which leads to the Get all data step, which serves to retrieve all the data required by the system. After that, the flow goes to the home page, which is the hub for accessing various system modules. From this main page, there are four process branches: (1) Close, which directs to closing the application, (2) Symptoms, for managing or displaying symptom data, (3) Disease, for accessing disease data, and (4) Rules, for setting or viewing the rules used in reasoning. The relationship between the elements shows that the Symptoms module can lead directly to the Disease module, which represents a direct link between symptoms and diseases in this expert system.

On the right side of the diagram, a more detailed flow for the follow-up process is illustrated. The Close module is connected to the Close Program, which closes the entire application and ends at the End node. In addition, there is a flow from Close to the Consultation module, indicating that users can close the consultation session without closing the application. The Rules module here serves to set the rules that form the basis for inference in the expert system. Overall, this diagram visualizes the hierarchical and sequential relationships between system components, where the management of symptom, disease, and rule data forms the core of interconnected processes. This approach reflects a structured design to ensure each module has a clear function, intermodule connectivity is maintained, and the system flow can be logically traced from start to finish (Ashar & Iqbal, 2024).

Next, Figure 2 shows a flowchart that illustrates the process of managing symptom data in a system, which generally includes adding, editing, and deleting data. The flow begins at the Symptoms node, which then directs to the Symptoms form, an interface or module used to display and process symptom data. From this form, there are three main process branches: Add, Edit, and Delete, each representing a symptom data management function.

In the Add branch, the process continues to the Symptoms (Add) module, which adds new symptom data to the database. The Edit branch directs to Symptoms (Edit), which allows users to change existing symptom data. Meanwhile, the Delete branch has a more complex flow: first, the system executes the variable Var=tblSymptoms get row selection to retrieve the selected data row, then processes the Delete Row at SQL command to delete the data from the database. After the deletion is complete, the system runs Get all data to reload all stored symptom data, ensuring that the data display on the interface is synchronized with the latest database. In detail, the flowchart of symptom selection is shown in Figure 2.

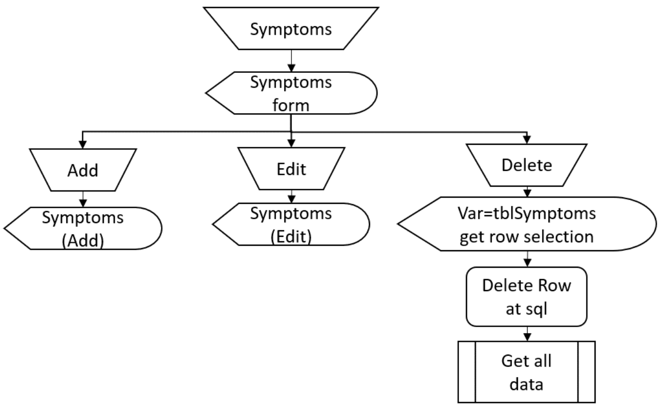

Figure 3. Symptom Options Flowchart

The diagram also reflects the CRUD (Create, Read, Update, Delete) operation pattern that forms the foundation of database management systems. This approach ensures that all data manipulations are performed in a structured, monitored, and directly integrated manner with the database. By refreshing the data through the "Get all data" operation after a deletion, the system guarantees the consistency and accuracy of the information displayed to users, which is a key principle in data-driven system development.

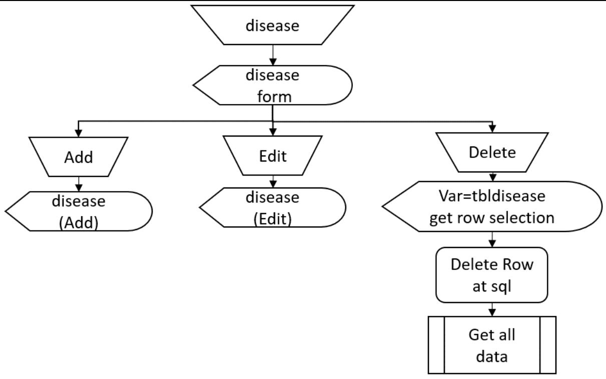

Figure 4. Disease Options Flowchart

Meanwhile, Figure 4 illustrates a flowchart that explains the database management process in a system, which includes three main functions: adding (Add), editing (Edit), and deleting (Delete) data. The flow begins at the base node, which then leads to the base form—an interface that serves as the database processing center. From this form, the flow branches into three main paths according to the operation selected by the user. The diagram also implements the CRUD (Create, Read, Update, Delete) principle, which is the core of data management in information systems. With this pattern, data integrity is maintained because every operation is carried out through clear and standardized procedures.

The addition of the Get all data step after deletion also reflects the application of data consistency in the system, so that synchronization between data in the interface and in the database is always maintained. This approach is crucial for ensuring the reliability and accuracy of the system, especially in multi-user or real-time applications.

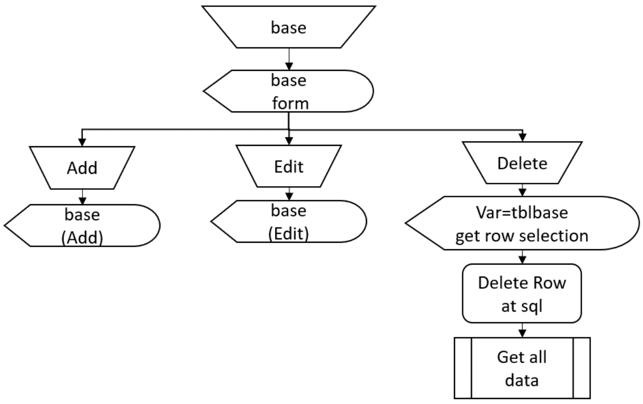

Figure 5. Base Options Flowchart

Figure 6 is a flowchart that illustrates the process of managing symptom data in an expert system or database application. The process begins at the Symptoms node (option), which is a menu option for managing symptom data. Next, there is a logical branch to check whether the selected option is “add” (data addition). If the condition is met (Yes), the system will execute the step “Var = Get total row tblSymptoms” to calculate the number of rows in the symptom table. This value is used as the Symptom code (new symptom code). After that, the user is prompted to enter the disease (inputting data related to the symptom), then the data is saved, followed by the process of Insert value into tblSymptoms to insert the value into the symptom table in the database (SQL).

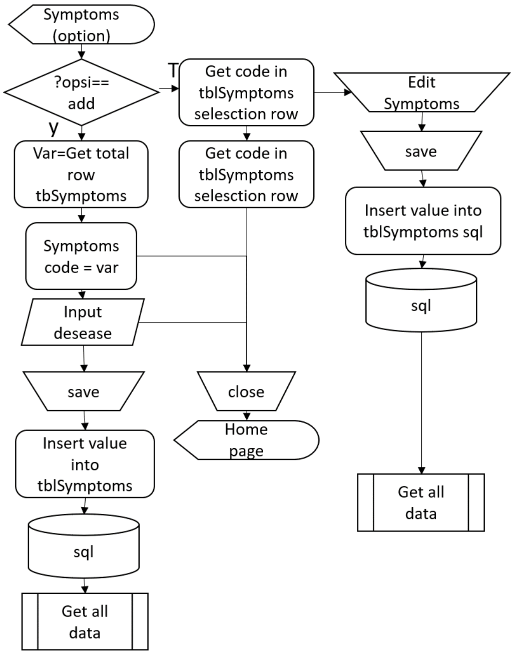

Figure 6. Flowchart Display Symptoms

If the initial condition is not met (No), the flow continues to the Get code in tblSymptoms selection row process, which is used to retrieve the symptom code from the selected data row. From here, there are two possible paths. First, proceed to Edit Symptoms to update existing symptom data, then save the data, insert it into the symptom table via an SQL command, and finally execute Get all data to update the data display. Second, the path leads to Close, which returns to the Home page, indicating cancellation or exit from the symptom management process.

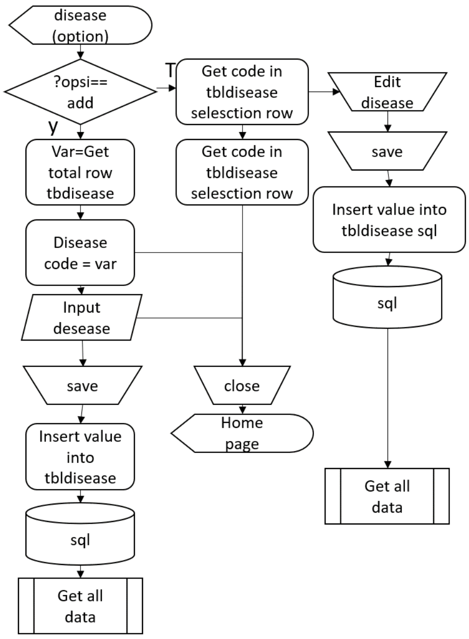

Figure 7. Disease Display Flowchart

Mechanism in Figure 7. If the initial condition is not met (No), the flow will continue to the Get code in tbldisease selection row process, which is used to retrieve the disease code from the data row selected by the user. From this process, there are two possible paths. First, proceed to Edit disease to update existing disease information, then save the data, re-enter it into the disease table via an SQL command, and continue with Get all data so that the data display on the interface matches the latest data in the database. Second, proceed to the close process, which returns the user to the Home page, indicating exit from the disease data management process without making any changes .

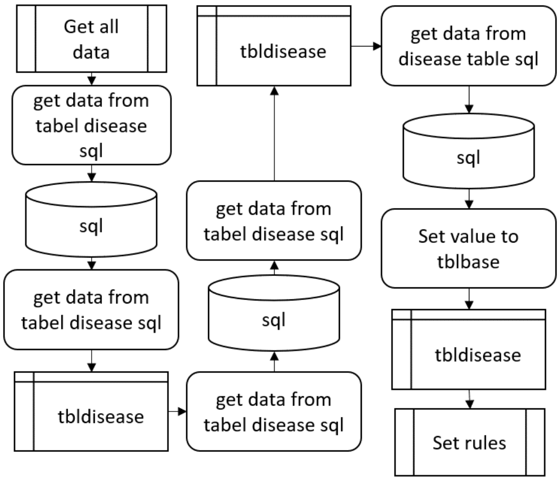

Figure 8. Flowchart of the getAllData function

Figure 8 is a flowchart of the process of integrating and loading data from the database (SQL) into the tables of the expert system interface, specifically for the diagnosis of respiratory diseases. This diagram shows how symptom, disease, and rule data are retrieved and presented systematically in the user interface (Sinaga & Aida, 2025).

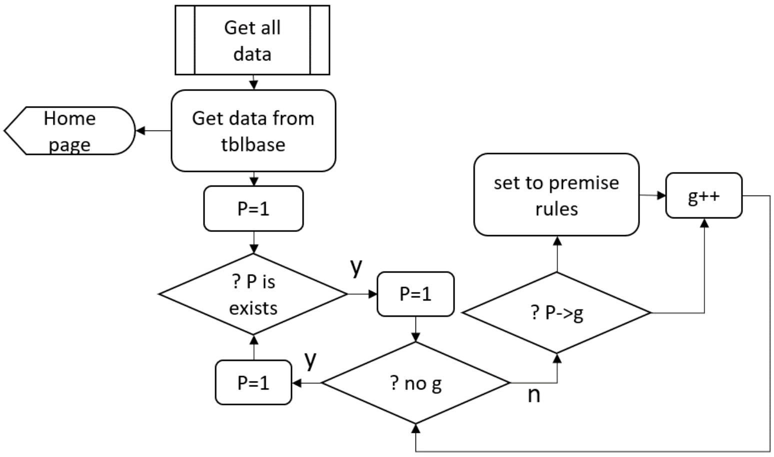

Figure 9. Flowchart Function Set Rules

Figure 9. is a flowchart of the rule traversal process in the Forward Chaining method in the respiratory disease diagnosis expert system. This flowchart illustrates how the system processes data from the rule table (tblbasis) to match symptoms with existing premises and generate inferences (disease diagnoses). Meanwhile, the consultation diagram can be seen in Figure 10.

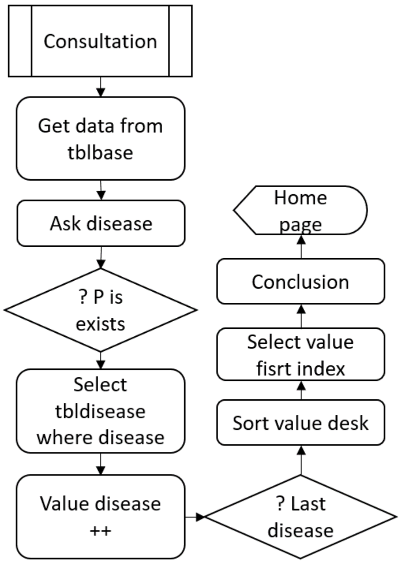

Figure 10. Flowchart Consulting function

3.3 *Screen Display*

The display of the application is divided into 2 parts, namely the main application and also the update application. The main application can be used in diagnosing, changing symptoms, changing diseases, and changing the knowledge base. The update application is used in making changes to knowledge data without going through the main application. The Home menu is used to display the amount of data in the database, including the number of symptoms, diseases, knowledge base, and rules. Documentation of the main page display and application updates can be seen in Figures 11 and 12.

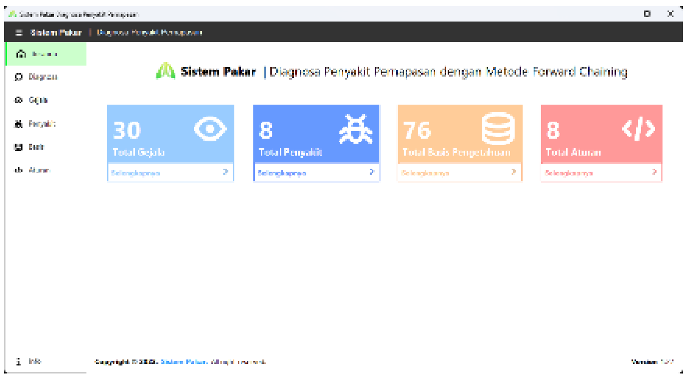

Figure 11. Main App Home Menu

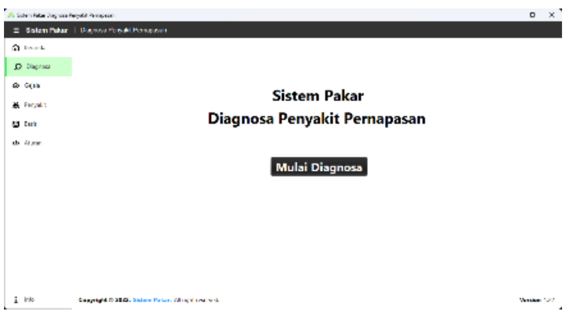

Figure 12. Main Application Diagnosis Menu

Jurnal Media Teknik Elektro dan Komputer
Vol. 02, No.01 (January - June, 2025)
E-ISSN: 3063-2927

The disease menu is used to add, change, or delete disease data in the database.

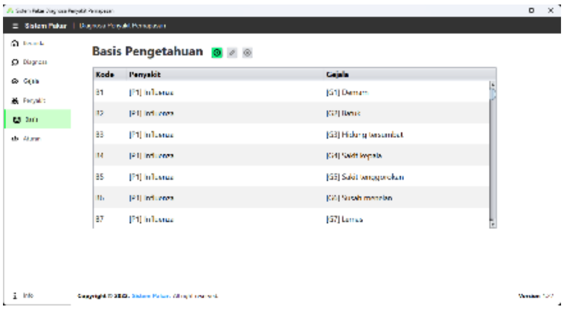

Figure 13. Main App Knowledge Base Menu

The knowledge base menu is used to add, change, or delete a knowledge base in a database.

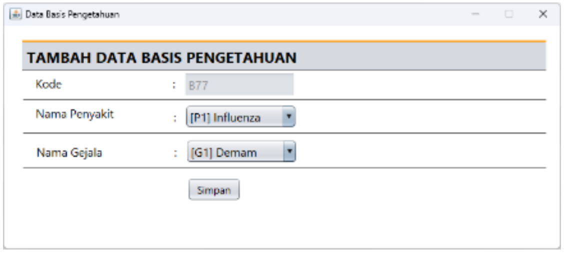

Figure 14. Main App Knowledge Base Add Menu

Menu add or change knowledge base.

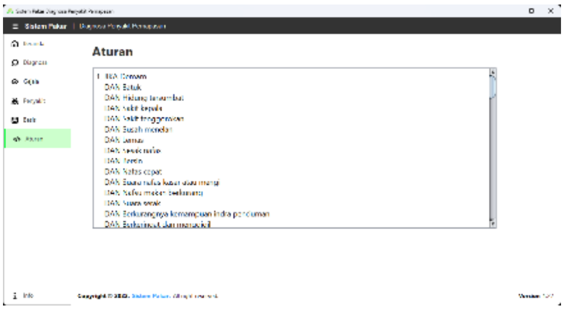

Figure 15. Main App Rules Menu

The rules menu provides an overview of the if-then rule tree in the form of rows. The number of rules is equal to the number of diseases used in the knowledge base.

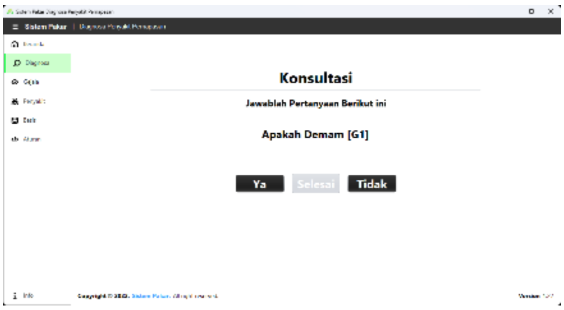

Figure 16. Main Application Consultation Menu

The consultation menu will run after the Start Diagnosis button is pressed in the Diagnosis Menu, every symptom that has been compiled in the knowledge base will be displayed during the consultation process. Select "Yes", if the symptoms displayed on the screen are experienced by the user, No if the symptoms displayed on the screen are not experienced by the user, and Done if the user wants to complete the consultation process, the completed option will only be selected when the user selects at least 1 Yes option.

Figure 17. Main App Conclusion Menu

The black box testing was conducted to evaluate the functional performance of the Expert System for Diagnosing Respiratory Diseases using the Forward Chaining Method. The test covered all critical features of the system, starting from the patient login process, input of patient data, symptom selection, and the final diagnosis results. Each test case was executed to verify whether the system responded according to the predefined requirements and provided accurate outputs (Nugroho & Rahmadani, 2024).

During the patient login testing, the system successfully authenticated users with correct credentials and displayed appropriate error messages when incorrect data was entered. The patient data input feature worked as expected, allowing users to save information without errors. The symptom selection module accurately recorded the chosen symptoms and proceeded to the diagnosis process without delays or data loss. Finally, the diagnosis result display provided precise disease predictions based on the Forward Chaining reasoning mechanism, with the output matching the expected results from the knowledge base. All tested functionalities met the intended criteria, and the results for every test case were marked as Valid, indicating that the system is ready for deployment and can be reliably used for respiratory disease diagnosis.

Table 5. Application Testing Results using the Black Box Method

No	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Remark
1	Patient login with correct credentials	Open patient login page, enter valid username and password, click "Login"	Username: patient01 Password: 123456	System verifies credentials and displays patient dashboard	Patient dashboard displayed	Valid
2	Patient login with incorrect password	Open patient login page, enter correct username and wrong password	Username: patient01 Password: abcdef	System displays "Invalid username or password" message	Error message displayed	Valid
3	Patient login with empty fields	Open patient login page, leave username	Username: *(empty)*	System displays "Please enter username and	Error message displayed	Valid

No	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Remark
		and password empty, click "Login"	Password: *(empty)*	password" message		
4	Patient login with incorrect username	Open patient login page, enter wrong username and correct password	Username: patientX Password: 123456	System displays "Invalid username or password" message	Error message displayed	Valid
5	Display list of diseases and symptoms	User selects "List of Disease and Symptoms Relationships" menu	-	System displays a list showing diseases along with their related symptoms	List displayed successfully	Valid
6	View detailed symptoms for a disease	From the disease list, click "View Details" on a specific disease	Disease: Diabetes Mellitus	System displays all symptoms associated with the selected disease	Details displayed correctly	Valid
7	Search disease by symptom	Enter symptom name in search bar	Symptom: Fever	System displays diseases related to the entered symptom	Related diseases displayed	Valid
8	Search symptom by disease	Enter disease name in search bar	Disease: Influenza	System displays symptoms related to the entered disease	Related symptoms displayed	Valid
9	No search results	Enter symptom or disease not in the database	Symptom: XYZ	System displays message "No data found"	Message displayed	Valid

5. CONCLUSION

Based on the results of the analysis, design, implementation and testing carried out on the respiratory disease diagnosis expert system with the desktop-based forward chaining method using the Java programming language, it can be concluded that this application can be used without the need for mobile data (offline), updates to the knowledge base can be done manually by users through the main application, and can also use the update application that will be provided separately in the program main. Based on the results of the analysis, design, implementation and testing carried out on the respiratory disease diagnosis expert system with the desktop-based forward chaining method using the Java programming language, it can be concluded that this application can be used without the need for mobile data (offline), updates to the knowledge base can be done manually by users through the main application, and can also use the update application that will be provided separately in the program main. that the application is only used for the dissemination process, it is not recommended to be used for the actual consultation process.

The methods used are still far from good, it is necessary to make improvements or even replacements to the methods used, such as the classification method brought by machine learning. The author also hopes that this application can be used for similar purposes, because this application can be used for different operations without the need to go through the coding process again. And also the database used still uses third-party applications (DBMS) so that it is difficult for ordinary users to make diagnoses, it is highly recommended for desktop application developers to use input/output streams rather than third-party database applications for the convenience of users.

REFERENCES

Anggara, Y. A. R., Wibowo, S. A., & Pranoto, Y. A. (2025). Sistem Pakar Diagnosa Kerusakan Laptop Berbasis Forward Chaining Dan Certainty Factor. *IJAI (Indonesian Journal of Applied Informatics)*, *9*(2), 296–309. https://doi.org/https://doi.org/10.20961/ijai.v9i2.95556

Ashar, S. F., & Iqbal, F. (2024). Web-Based Information System for Borrowing Practical Laboratory Equipment for The Faculty of Public Health. *Jurnal Metrokom: Media Teknik Elektro Dan Komputer*, *1*(2), 121–133. https://doi.org/https://doi.org/10.1307/metrokom.v1i2.77

Elkana, S. R., & Kuswanto, V. (2023). Analysis and Design of Disease Diagnosis Systems and Patient Medicine Recommendations with Forward Chaining Method. *Bit-Tech*, *6*(2), 134–143. https://doi.org/https://doi.org/10.32877/bt.v6i2.937

Mardian, D. R., Risnanto, S., Garnia, E., Adhinugraha, K. M., Rahayu, H., & Rachman, R. S. (2023). Forward Chaining and Certainly Factor Method Optimization for Lung Disease Expert System. *2023 17th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, 1–4. https://doi.org/10.1109/TSSA59948.2023.10366958

Mauliza, M., Ula, M., Saputra, I., Afdelina, R., & Ikhsan, M. (2022). Application of expert system with forward chaining method in detecting infectious diseases in children. *Science Midwifery*, *10*(4), 2777–2785. https://doi.org/https://doi.org/10.35335/midwifery.v10i4.714

Nengsih, Y., & Putra, N. (2020). SISTEM PAKAR MENGGUNAKAN FORWARD CHAINING DAN CERTAINTY FACTOR UNTUK DIAGNOSA KERUSAKAN SMARTPHONE. *JURSIMA*, *8*(2), 61–69. https://doi.org/10.47024/js.v8i2.205

Nugroho, A., & Rahmadani, N. F. (2024). Web-based Visit List Information System at the Ministry of Religious Affairs of Deli Serdang Regency. *Jurnal Metrokom: Media Teknik Elektro Dan Komputer*, *1*(2), 58–74. https://doi.org/https://doi.org/10.1307/metrokom.v1i2.79

Putra, H., Harahap, I. S., Islam, U., & Sumatera, N. (2024). Expert System for Detecting Potential Obesity in the Environment of the State Islamic University of North Sumatra Using Naïve Bayes Classifier. *Jurnal Media Teknik Elektro Dan Komputer*, *1*(1), 27–36.

Rifki, M. I., Raditya, M. E., & Hasugian, A. H. (2023). Text Data Security Application Using a Mobile-Based Base64 Algorithm. *Instal: Jurnal Komputer*, *15*(02), 224–235. https://doi.org/https://doi.org/10.54209/jurnalkomputer.v15i02.146

Rifki, M. I., & Syamia, N. (2024). Message Security Application Using Mobile-Based AES Algorithm. *Journal of Computer Science, Information Technology and Telecommunication Engineering*, *5*(2), 595–606. https://doi.org/https://doi.org/10.30596/jcositte.v5i2.20834

Sinaga, Y. A., & Aida, M. (2025). WEB-BASED EMPLOYEE LEAVE APPLICATION SYSTEM AT THE DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU KOTA MEDAN. *Jurnal Metrokom: Media Teknik Elektro Dan Komputer*, *2*(1), 1–17. https://doi.org/https://doi.org/10.1307/metrokom.v2i1.100

Sukma, I., & Petrus, M. (2020). Sistem Pakar Penyakit Kucing Menggunakan Metode Forward Chaining Berbasis Web. *Simtek: Jurnal Sistem Informasi Dan Teknik Komputer*, *5*(1), 52–58. https://doi.org/https://doi.org/10.51876/simtek.v5i1.73

Syamsudin, I., & Sudarsono, E. (2022). Penerapan Metode Certainty Factor Pada Sistem Pakar Mendiagnosa Penyakit Tuberculosis (Tb) Paru Berbasis Web Mobile. *JUSIM (Jurnal Sistem Informasi Musirawas)*, *7*(2), 127–136. https://doi.org/doi.org/ 10.35970/ jinita.v5i2.2096

Wajidi, F., & Nur, N. (2021). Sistem Pakar Diagnosis Penyakit Stunting pada Balita menggunakan Metode Forward Chaining. *Jurnal Informatika Universitas Pamulang*, *6*(2), 401–407. https://doi.org/DOI: 10.32493/informatika.v6i2.11938

Ware, S., Yue, C., Morillo, R., Lu, J., Shang, C., Bi, J., Kamath, J., Russell, A., Bamis, A., & Wang, B.

(2020). Predicting depressive symptoms using smartphone data. *Smart Health*, *15*, 100093. https://doi.org/https://doi.org/10.1016/j.smhl.2019.100093

Wenda, A., Kraugusteeliana, K., Suryanto, A. A., Suhada;, & Karya;, S. N. A. (2023). Sistem Pakar Untuk Mendiagnosa Penyakit Paru-Paru dengan Menggunakan Metode Teorema Bayes. *Jurnal Media Informatika Budidarma*, *7*(1). https://doi.org/https://doi.org/10.30865/mib.v7i1.5394