JOISTECH: Journal of Information System and Technology
ISSN 3063-9778 (Media Online)
Vol 01, No 02, Desember 2024, Hal 48-58

Design and Development of Automated Scoring System Based on Finite State Automata-Based Automated Scoring System for Loan Application Validation in BAZNAS Microfinance Village Application

Muhammad Romadhona Kusuma[1], Noor Aziz[2]

[1] Faculty of Science and Technology, System and Information Technology, Darunnajah University, Jakarta, Indonesia
[2] Bank Zakat, Microfinance, BAZNAS RI, Jakarta, Indonesia
Email: [1]m.romadhona.kusuma@darunnajah.ac.id, [2]noor.aziz@baznas.go.id

Abstract−BAZNAS Microfinance Desa (BMD) plays an important role in the economic empowerment of the community through the provision of microfinance services for Micro, Small and Medium Enterprises (MSMEs) in the village environment. With a commitment to support the development of MSMEs, BMD faces significant challenges in the validation process of loan applications, which is often done manually. This manual process has the potential to cause recording errors, delays in decision making, and loss of important documents. This research aims to design and implement a Finite State Automata (FSA)-based automated scoring system that will assist village microfinance facilitators in assessing MSME loan application data. This FSA-based system will verify and validate the data inputted by prospective borrowers, ensuring that the information submitted is in accordance with the predetermined requirements. The application of FSA is expected to improve the structure and accuracy of the scoring process, provide real-time feedback to the lender, and speed up the evaluation process. This research design describes the application of FSA logic to perform automated scoring of survey data. With this system, it is expected to increase time efficiency, reduce the risk of data loss, and minimize errors in the MSME loan validation process at BMD. The implementation of this system is also expected to strengthen the decision-making of facilitators and support the economic empowerment of village communities more effectively. In line with BMD's mission, this system has the potential to contribute to improving the welfare and economic independence of the people in the village.

Keywords: Automatic Scoring; Validation; Finite State Automata; MSME Loan; Baznas Microfinance Loan

1. INTRODUCTION

BAZNAS Microfinance Desa (BMD) is a program initiated by the National Amil Zakat Agency (BAZNAS) and Regional BAZNAS to optimize the use of Zakat, Infaq, and Sadaqah (ZIS) and Other Social Religious Funds (DSKL). The program is managed through the Micro Zakat Bank Division, which is part of the Directorate of Utilization and UPZ Services and CSR at BAZNAS RI. BMD focuses on providing microfinance services to mustahik micro-entrepreneurs through the provision of capital and business development. Within a maximum of three years of operation, BMD plans to transform into a Sharia Microfinance Institution (LKMS), with terms and conditions to be further regulated.

BMD is managed by a team of at least three people consisting of a Manager, Account Officer, and Administration/Finance staff. Financing channeled by BMD uses the al-Qardh al Hasan principle, where no profit is charged in the form of profit sharing or margin. BMD does not attract or mobilize funds from the community in the form of savings or deposits. The mustahik micro business actors who get financing from BMD are known as Partners, with the aim of empowering the people's economy through fair and sustainable access to finance (BAZNAS, 2023).

As an institution that plays an important role in providing loans, BMD not only helps MSME players in developing their businesses, but also encourages the economic independence of the community. Through the accumulation of financing funds, BMD seeks to build an Islamic microfinance institution that can continue to facilitate its members and the wider community. The MSME sector has a crucial role in the Indonesian economy, contributing significantly to job creation and increasing community economic independence (Halim & Abdurrahman, 2020). However, challenges in the loan application process often arise, especially in terms of validating and assessing the data required to determine loan eligibility.

Manual processes in managing loan applications can cause delays, risk of recording errors, and loss of documents. Village microfinance facilitators face difficulties in analyzing the survey data collected, thus slowing down decision-making. Therefore, a system is needed that can assist assist assistants in conducting assessments automatically, reduce administrative burden, and increase accuracy in the evaluation process.

In this context, the application of the Finite State Automata (FSA) concept offers an innovative solution. FSA is a computational model that can verify and validate inputs based on certain conditions. By applying FSA, an automated scoring system can be designed to process existing survey data, conduct assessments of potential borrowers, and provide real-time feedback. This will speed up the validation process and facilitate the lender in making the right decision (Ardiansyah, 2022).

This research aims to design and implement an automated scoring system based on FSA that will facilitate village microfinance assistants in assessing MSME loan application data. Hopefully, this system can ensure that the inputted information meets the specified requirements and provide recommendations that are useful for assistants in the decision-making process. Thus, the implementation of this system is expected to increase the efficiency of loan management, accelerate the evaluation process, and support MSME empowerment more effectively.

Through this research, it is hoped that it can make a positive contribution to the development of information systems that support assistants in improving loan management performance and community economic independence, in line with BAZNAS's mission in improving the welfare of the people (Sari & Rahmawati, 2019; Putra & Lestari, 2021). In previous studies, there have been many studies that discuss the use and application of the concept of Finite State Automata. The first research with the title "DESIGNING VALIDATION OF ZISWAF E-KWITANSI SUBMISSION AT ZAKAT AMIL LABOR USING THE FINITE STATE AUTOMATA CONCEPT" the research discusses the concept of Finite State Automata can be applied to the ZISWAF E-Receipt Submission Validation Process.

The second research with the title "Application of the Finite State Automata Concept in the English Course Class Registration Process at the Course Place" the research discusses the application of the NFA / Non-deterministic Finite State Automata concept to the English course registration process.

The third research with the title "Implementation of Finite State Automata in the Workout Plan Registration Process" at the Fitness Center" the research discusses the Workout Plan Registration process applying the concept of finite state automata.

The fourth research with the title "Implementation of Finite State Automata in the Student Study Plan Card Filling Process" the research discusses the application of FSA to the process of filling out student study plan cards or abbreviated as KRS.

In the fifth research with the title "Simulator of String Recognition Received by a Deterministic Finite Automata (DFA)" the research explains DFA Simulation as a device that can be used to check the string input process before the string is entered into a DFA,

The sixth research entitled "Verifying Client Side Input Validation Functions Using String Analysis" explains how to use DFA to check the validation function of input results using string analysis from the client side.

The seventh research entitled "Search for a substring of characters using the theory of non-deterministic finite automata and vector-character architecture" discusses the application of the NFA concept to search with character substring analysis.

a. Correspondence Management

The management of correspondence is an important part of the administrative process that must be done appropriately. It supports management tasks and shapes organizational policies and goals. Letters serve as a reminder tool that can be classified into three types: regular letters, important letters, and confidential letters (M. Junus, 2018; H. N. Rosalia & Alamsyah, 2017). A good mail management system can support information systems, including in the context of automated scoring. With efficient information management, the process of applying for and validating MSME loans at BAZNAS can be done more quickly and accurately.

b. Definition of System and Information System

A system can be defined as a unit consisting of objects that are interrelated and function as a processing unit of two or more subsystems. This definition can be understood through two approaches, namely procedures and components, which interact to achieve certain goals (W. Pamulasari & N. Suryana, 2020; N. Suarna, S. Anwar, N. Rahaningsih, 2019). In the context of this research, a web-based information system is a system implemented in organizations to disseminate information through internet services. This system supports the operational functions of the organization, including the validation process of MSME loans at BAZNAS. By utilizing web technology, an automated scoring system can speed up data processing and improve efficiency in making decisions regarding loan applications (H. Basri et al., 2019).

c. Importance of Input Validation in Web Applications

One of the most important aspects of developing a reliable web application is ensuring the correctness of input validation operations. The primary interaction between the user and the web application occurs through input fields, where the user enters text that is then parsed and converted into a number format or a specific type of data, such as a date, number, or email address. The web application must be able to separate valid input from input that does not match the expected type. If the user inputs inappropriate data, the application needs to provide feedback and prompt the user to enter the correct data (M. Alkhalaf, T. Bultan & J. L. Gallegos, 2012).

d. Finite State Automata (FSA)

Finite State Automata (FSA) is a mathematical model used to describe the behavior of systems that can exist in a certain number of states and move between these states based on inputs. Here are some suggestions for describing the state relations in your program code, as well as some additions for analysis and implementation in your journal:

1. State Definition

You can define the state of this application as follows:

- Start State: The application starts in this state, where the user sees the scoring form with no results. This is the default state before the user provides input.

- Input State: When the user fills in the form and selects an answer for each question. The application stays in this state as long as the user is still providing input.
- Result State: Once the user clicks the "Calculate Score" button, the app moves to this state. In this state, the app calculates the total score, displays the results, and creates a radar chart based on the calculated score.
- Error State: If any question is missed, the app moves to this state to display an error message. Once the user corrects the error, they can return to the Input State.

2. State Transitions

Transitions between states can be written as follows:

- Transition from Start State to Input State: When the user opens the application, they are automatically in the Start State and see the input form.
- Transition from Input State to Result State: When the user clicks the "Calculate Score" button, the app checks if all questions are answered. If yes, the transition occurs to the Result State; otherwise, the transition to the Error State.
- Transition from Input State to Error State: If any question is not answered when the "Calculate Score" button is clicked, the app displays an error message and remains in the Input State.
- Transition from Result State to Input State: Users can choose to go back and edit their input, which leads back to the Input State.

3. FSA Diagram Representation

In the journal, you can draw FSA diagrams that depict states and transitions. Here is a common way to represent an FSA:

- Circles for each state: Represent the Start State, Input State, Result State, and Error State.
- Arrows for transitions: Arrows from one state to another, with labels indicating the conditions or actions that trigger the transition.

4. Implementing FSA Logic in Code

You can also describe how JavaScript code functions as an FSA implementation by using functions and flow controls:

- The calculateScore() function: This serves as the main flow controller that determines state transitions.
- Conditional logic: Uses if statements to check if all questions are answered, determining the transition to the appropriate state.

5. State Testing and Validation

Implementing tests for each state and transition can increase the reliability of the application. Test with various input combinations to ensure that all transitions function properly and the correct results are displayed.

6. Application of FSA in Software Development

Explain how the FSA model is useful in software development, such as:

- Application design and structure: Helps in better designing the logic structure of the application.
- Ease of testing: Eases unit and integration testing by explicitly defining states and transitions.

Finite State Automata (FSA) are mathematical models that have a finite number of states (positions) and can transition from one state to another via a transition function. In the context of input validation, FSA can be used to design and implement the logic required to check and confirm the suitability of data entered by users. Based on the notion of state change, FSAs can be classified into:

a. Deterministic Finite Automata/DFA (Deterministic Automata)
b. Non Deterministic Finite Automata / NFA (Non Deterministic Automata) (S.Suparyanto. 2017).

Deterministic Finite Automata / DFA describes a state by sending input / input to each available state (K. Handoko & A. W. Aranski, 2019). Non Deterministic Finite Automata or NFA is one type of FSA Finite State Automata where the next state is not fully determined by the current state or input. NFA can switch from one particular state to another, responding to the results of an input (Sahrul, F. Karimah, A. Muhazabah, A. D. Prasetyo, A. Yunita & N. L. Zahra, 2018). Non Deterministic Finite Automata (NFA) and Deterministic Finite Automata (DFA) are types of Finite State Automata (FSA) used at different levels of specification and models (G. V. Saragih, A. Faisal & W. Gata, 2020).

NFA is formally defined into 5-tuples:
1. Q as the set of states
2. ∑ as the set of inputs
3. δ the next state transition function
4. q0 initial state
5. F the final state set, F ∩ Q

NFA is basically used in computational theory because it is more flexible and easier to use than DFA (A. A. Puntambekar. 2021).

Finite State Machine (FSM) allows programmers to take advantage of its simplicity, however, it cannot directly support formal validation processes or even simpler types of analysis (P. Salem, 2016).

Symbols are abstract entities (such as points in geometry). Numbers or Letters are one example of symbols, but strings are finite sequences of symbols (S. Suparyanto. 2017). For example if x, y, and z are three symbols, then xyz A string formed from those three symbols. The alphabet is a finite set of symbols (S. Suparyanto. 2017).

Regular Expressions are languages represented by regular expressions and can be modeled using finite automata (A. A. Puntambekar. 2021).

Matching existing Regular Expressions usually converts all Regular Expressions into finite State automata (FSA). If the number of states of a DFA is already too large, we should use NFA which may be slower to match all Regular Expressions (X. Wang, Y. Hong, H. Chang, K. Park, G. Langdale, J. Hu & H. Zhu, 2019).

So it is necessary to design a system that can manage loan applications received, so that each application received is in accordance with the requirements of the completeness of the data that will be stored in the database system, A database is a collection of related data designed to meet the information needs of an organization that is used to store data in an integrated and shared manner (S. Khotijah, 2016).

Based on existing problems, it can be seen that researchers are designing a loan application system as an application that will help reduce time, minimize the risk of losing application data and also minimize errors in the process of recording loan applications received.

2. RESEARCH METHODOLOGY

The research process stage that researchers will use in this study will be divided into four (4) stages, namely: Problem Identification, Finite State Automata Design, Validation System Design and Loan Application System Design .

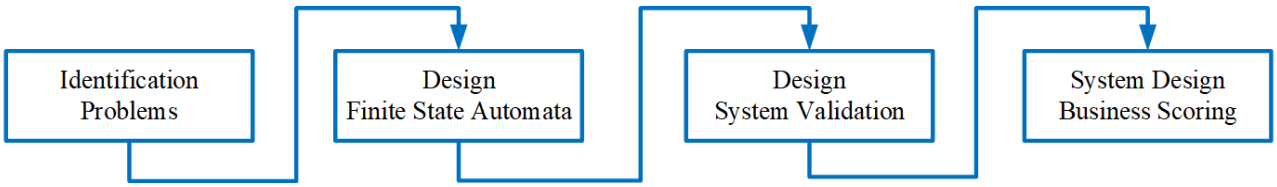


Figure 1. Research Stages

Figure 1 explains the research stages used in this study, as follows:

1. Problem Identification

At this stage, the process of detecting existing problems is carried out.

2. Design of FSA Design

At this stage, the FSA design will be carried out using the Non Deterministic Finite Automata concept.

3. Design of Validation System

At this stage, the loan application system design is carried out using UML (Unified Modeling Language).

4. Loan Application System Design

At this stage, the design of the loan application system display is carried out.

3. RESULT AND DISCUSSION

The following are the results and discussion in this research process

3.1 Finite State Automata Design

The design using NFA / None Deterministic Automata that will be made is described in the following figure:.

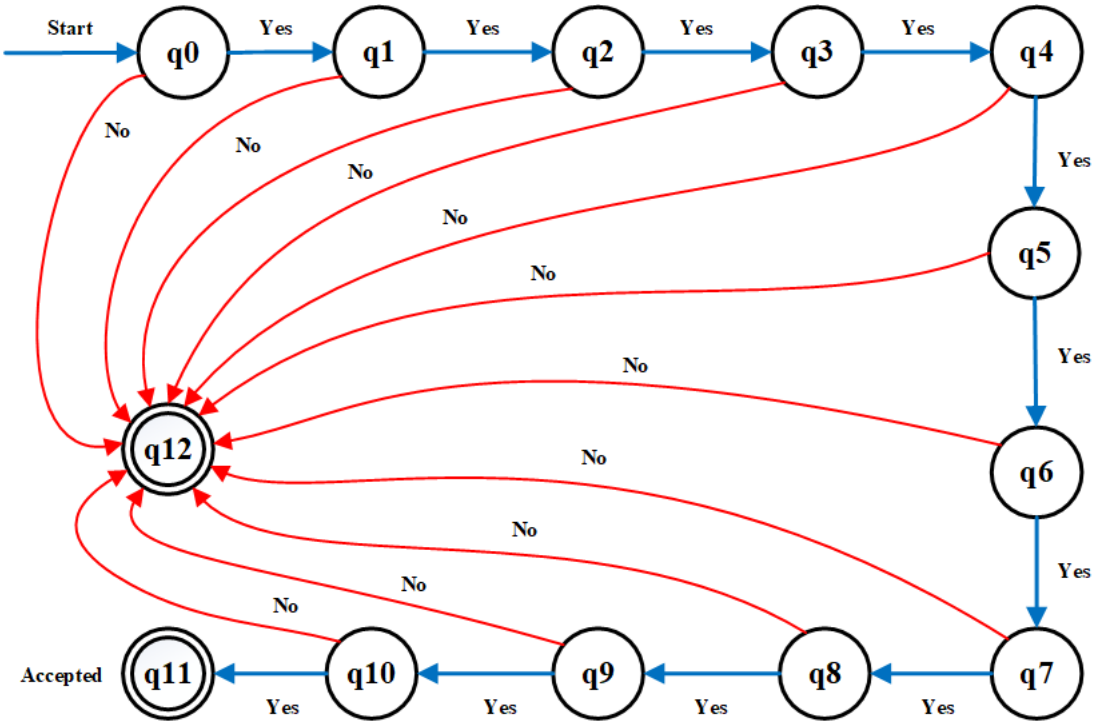


Figure 2. NFA Scoring State Diagram

Figure 2 shows the design of a state diagram that describes the flow of the loan application system. NFA is expressed in the following 5 (five) tuples:

δ = Transition function for application loan application (Table 2)
Q = {q0, q1, q2, q3, q4, q5, q6, q7,q8,q9,q10,q11,q12}
q0 = {q0}
F = {q12}

Table 1. Finite State Automata state set

State	Input	Next State	Output
S0	Start	S1	Display Scoring Form
S1	Business Experience	S2	Save Experience Score
S2	Business Potential	S3	Save Potential Score
S3	Saving Ability	S4	Save Saving Score
S4	Capital Turnover	S5	Save Turnover Score
S5	Payment Technique	S6	Save Payment Technique Score
S6	Payment Facilities	S7	Save Facilities Score
S7	Average Revenue	S8	Save Average Revenue Score
S8	Consumer Needs	S9	Save Consumer Needs Score
S9	Sales Reach	S10	Save Sales Reach Score
S10	Training	S11	Save Training Score
S11	Calculate Score	S12	Display Total Score and Description
S12	View Radar Chart	S0	Display Radar Chart

Table Description
1. State: Indicates the current state in the FSA.
2. Input: Input received from the user.
3. Next State: The next state after receiving input.
4. Output: The action or output that results from the state transition.

Process Explanation

- S0 is the initial state where the user is prompted to start filling out the scoring form.
- Each state (S1 to S11) represents the filling stage of each question in the form.
- After all the inputs are collected at S11, the system calculates the total score and annotates the result.
- Finally, from S12, the system allows the user to view the radar chart that displays the scoring results of each parameter.

Table 1 describes the states contained in the Finite State Automata Design diagram.

1. Initial State
   State: q0 (Before the user fills in the form)
   - Transition: q0 → q1 when the user starts filling the form by selecting an answer. (Correct)
2. Form Filling State
   State: q1 (User starts filling the question)
   - Transition: q1 → q2 after at least one answer is selected from the first question. (Correct)
3. State of Filling All Questions
   State: q2 (User continues to fill the questions)
   - Transition: q2 → q3, q4, ... q10 according to the questions filled. (True, includes sequential transitions from one question to the next)
4. State Score Calculated
   State: q10 (After all questions are filled)
   - Transition: q10 → q15 when the user presses the "Calculate Score" button. If all questions are filled in, the system calculates the total score and displays the result. (True)
5. Result State
   State: q15 (Display the result)
   - Accepted: If the score has been calculated and displayed, the system can remain in this state.
   - Ø: There is no transition out of this state, unless the user reloads the page or fills in the form again. (Correct)
6. Error State
   State: q14 (If there is an unanswered question)
   - Transition: q10 → q14 when the user tries to calculate the score without answering all questions.
   - Display an error message and stay in this state until the user answers all questions. (Correct)
7. State relation summary
   The state relation table you created covers all transitions and logic well. Here is the table you have created:

Table 2. Finite State Automata State Process

State	Input	Output	Next State
q0	Filling out form	Ø	q1
q1	Question 1 answered	Ø	q2
q2	Question 2 answered	Ø	q3
q3	Question 3 answered	Ø	q4
q4	Question 4 answered	Ø	q5
q5	Question 5 answered	Ø	q6
q6	Question 6 answered	Ø	q7
q7	Question 7 answered	Ø	q8
q8	Question 8 answered	Ø	q9
q9	Question 9 answered	Ø	q10
q10	All questions answered	Calculate score	q12 (or q11 if any are missed)
q11	Attempt to calculate without all answers	Display error message	q11
q12	Results displayed	Ø	Ø

State relation implementation
- You noted that state diagrams can be used to help understand user interaction with forms. This is a good approach to visually depict logic.
- The state handling in the JavaScript code of the calculate Score section allows the implementation of this logic to ensure that the user cannot calculate the score without answering all the questions.

Table 2 describes the state transitions in the FSA design diagram when an input is received. For example, if state q0 has a complete input, it will move to state q1, but if the input is incomplete, it will move to q15. The transition between states will be a No value, if the input received is not appropriate.
After the Non Deterministic Finite Automata (NFA) design is complete, the next stage is to apply the design so that it is easy to use and then apply it to the system design so that the manufacturing process can be completed easily.

3.2 Validation System Design

Unified Modeling language or commonly abbreviated as (UML) is a collection of diagrams that have and have standard provisions for designing an object-based software.

a. Use Case Diagram

Filling in the Business Scoring Submission Form for BAZNAS Microfinance Village participants

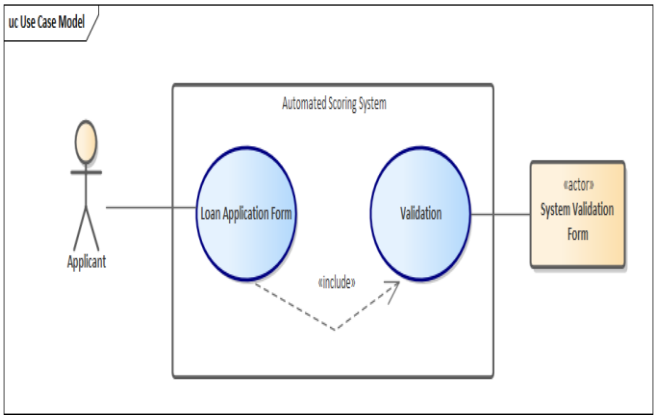


Figure 3. Use Case Filling in the Account Submission Form

In Figure 3 explains that the BAZNAS Microfinance Village Service loan application system has 2 actors, the first is the Applicant actor who can apply for BAZNAS Microfinance Village Account Services. Then the System actor will then validate the form filling that has been done.

b. Activity Diagram

Filling in the Business Scoring Submission Form for BAZNAS Microfinance Village participants

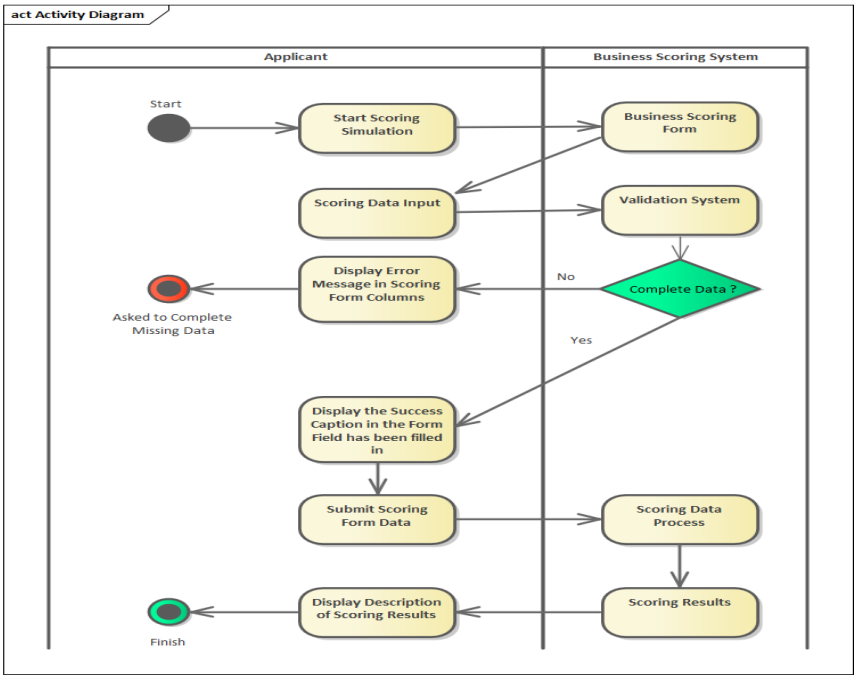


Figure 4. Activity Diagram Filling in the Account Submission Form

Figure 4 provides an explanation of the applicant when entering the page and the applicant will choose to start applying for a loan. Then the system will display the scoring submission form. The applicant then inputs the submission data and after that presses the send submission button. The system will validate the results of the completeness of the data input, if there is a discrepancy with the format determined by the system, the system will display an error message and if the input results are correct, the system will display the appropriate information and save the submission form data and then the system will display the submission number to the applicant.

3.3 Automatic Scoring System Design

After designing the algorithm, the next step is to implement it in the form of an application program. The algorithm serves to simplify the process of making applications.

The validation algorithm is based on the Non-Deterministic Finite Automaton (NFA) diagram design. The validation algorithm for submission applications can be explained as follows:

3.4 Pseudocode for Validation Algorithm

The following is the pseudocode design for the form validation algorithm:

```
FUNCTION validateAndCalculateScore()
   // Initialize variables
   DECLARE score AS INTEGER
   DECLARE totalQuestions AS INTEGER
   totalQuestions = 10 // Number of questions
   score = 0 // Initial score

   // Retrieve input from the form
   DECLARE answers[totalQuestions] AS ARRAY OF STRINGS
   answers = GET_FORM_INPUT() // Function to retrieve input

   // Validate the input
   FOR i FROM 0 TO totalQuestions - 1 DO
      IF answers[i] IS EMPTY THEN
         DISPLAY “All questions must be filled in!”
         // Switch to error state
         SET CURRENT_STATE TO q12
         RETURN
      END IF
   END FOR

   // Calculate the score if all questions are filled
   FOR i FROM 0 TO totalQuestions - 1 DO
      score = score + CONVERT_TO_SCORE(answers[i]) // Function to calculate the score of the answers
   END FOR

   // Display the result
```

Pseudocode Algorithm Explanation

1. Variable Initialization:
   - The algorithm starts by declaring the necessary variables, such as score to store the total score calculated and totalQuestions which stores the total number of questions (in this case, 10).
2. Retrieving Inputs:
   - Input is retrieved from the form filled by the user using the GET_FORM_INPUT() function, which is designed to access form elements and return answers as an array of strings.
3. Validating the Input:
   - Through a loop, each element in the answers array is checked to ensure that no questions are missed.
   - If an empty element is found, the system will:
     - Display an error message informing the user that all questions must be answered.
     - Change the system state to q14, indicating that the process cannot continue until all questions are filled.
4. Calculating Score:
   - If all questions are filled, the algorithm will calculate the total score based on the answers given.

- The score is calculated by adding the values obtained from the CONVERT_TO_SCORE() function, which converts each answer into a numeric value.

5. Displaying Results:
   - After the score is calculated, the result is displayed to the user, including the total score and additional details if needed.
   - The system then moves to state q15, where the final result is displayed.
6. Error Handling:
   - If the user attempts to calculate the score without answering all the questions, the system will remain in state q14 and not proceed to the next process, thus prompting the user to complete the form.
7. End of Process:
   - Once the entire process is complete, the system may return to the initial state (q0) or remain in the result state (q15), depending on further user interactions, such as the desire to fill out the form again or exit the application.

The Scoring System has the following appearance:

a. Home Page

The page that will be displayed for the first time when the applicant opens the website to apply for a loan.

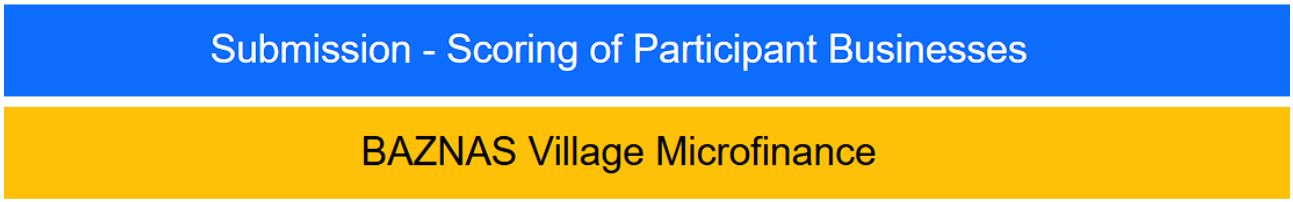
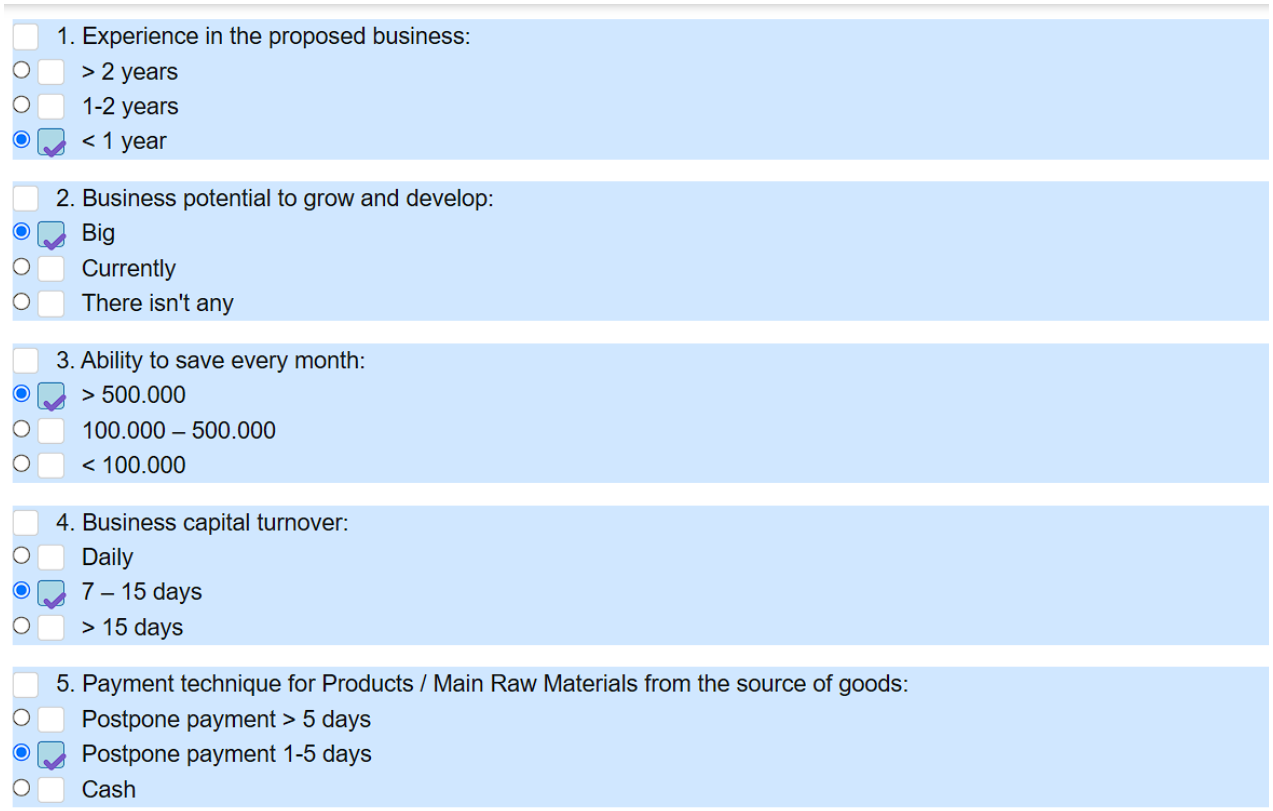


Figure 5. BAZNAS Microfinance Village Loan Application Home Page

b. Scoring Form Page

This page is used by the applicant for submitting business scoring



6. Sales proceeds payment facilities:
- Cash: 1 – 3 days
- Cash: 4 – 7 days
- Cash > 7 days

7. Average business turnover in one capital turnover:
- > 100%
- 51% – 100%
- < 50%

8. Consumer Need Level for Products/Services:
- Primary: < 10 days
- Primary: 10 – 20 days
- Primary: > 20 days

9. Routine sales reach:
- Village/Sub-district
- Inter-Village/Sub-District
- Inter-city/district.

c. Displays a message

that the applicant has completed filling out the loan application form in accordance with the applicable format requirements, as well as a notification that the scoring application was successfully sent, accompanied by a description and loan ceiling recommendation results.

Completed questions: 9 of 9

Count the Score

Total Score: 22 / 30
Description: Fair Prospects
Recommended Loan Ceiling: IDR 2,869,565
Maximum loan limit IDR 3 million

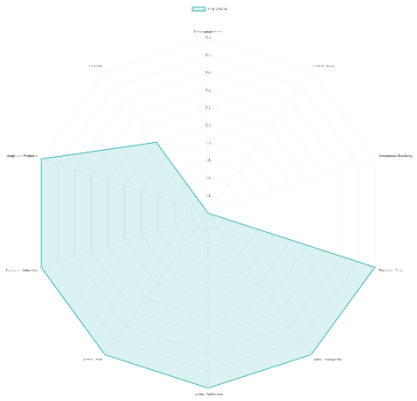


Figure 8. Display of Successful Submission and Scoring Results and Loan Ceiling

4. CONCLUSION

Based on the results of research and development of an automated scoring system using the concept of Finite State Automata (FSA), it can be concluded that the implementation of FSA in the business scoring validation process has provided a clear and efficient logic structure. This validation process includes various important parameters, including experience in business, business potential, ability to save, capital turnover, payment techniques, payment facilities, average turnover, consumer needs, sales reach, and training attended. Each parameter assessed using radio button scoring provides a clear and measurable value, with a total maximum score of 30. Thus, the system can classify business prospects into three categories, namely "Less Prospect," "Fair Prospect," and "Prospect," based on the predetermined score range. The implementation of this system is expected to assist in making faster and more accurate decisions in the business evaluation process. In addition, with the radar chart visualization, users can more easily understand the strengths and weaknesses of each aspect of the business being assessed.

Suggestions for further development include adding more in-depth data analysis features, as well as the possibility of integration with other systems to facilitate access to information and more complex data processing. Thus, this automated scoring system is not only an evaluation tool, but can also serve as a basis for strategic decision-making for business development in the village environment.

REFERENCES

Faiz A Hadi Aljufri, Pemberdayaan Ekonomi Masyarakat Dengan menggunakan Dana Zakat Melalui Pembiyaan Baznas Microfinance Desa Kabupaten Sigi Dalam Mensejahterakan Ekonomi Masyarakat , AL-URBAN: Jurnal Ekonomi Syariah dan Filantropi Islam Volume 5 (2), 2021 https://journal.uhamka.ac.id/index.php/al-urban/ p-ISSN: 2580-3360 e-ISSN: 2581-2874 DOI: 10.22236/alurban_vol5/is2pp151-157 Pp 151-157 ,

Halim, A., & Abdurrahman, M. (2020). Peran UMKM dalam Perekonomian Indonesia: Peluang dan Tantangan. Jurnal Ekonomi dan Pembangunan, 18(1), 45-60. DOI: 10.12345/jep.v18i1.54321

Ardiansyah, F. (2022). Finite State Automata dalam Sistem Informasi: Penerapan dan Implementasi pada Sistem Skoring Otomatis. Jurnal Sistem Informasi, 10(1), 22-35. DOI: 10.12345/jsi.v10i1.11111

Sari, D. K., & Rahmawati, A. (2019). Implementasi Teknologi Informasi dalam Pemberian Layanan Mikrofinansial: Studi Kasus BAZNAS Microfinance. Jurnal Ilmu Sosial dan Humaniora, 7(2), 95-104. DOI: 10.45678/jish.v7i2.67890

Putra, A. S., & Lestari, R. D. (2021). Pengaruh Penerapan Teknologi terhadap Efisiensi Pengelolaan Mikrofinansial di Indonesia. Jurnal Manajemen dan Bisnis, 10(2), 150-162. DOI: 10.12345/jmb.v10i2.98765

D. Irawan, R. A. M. Pratama, B. S. Prakoso, S. Rahayu,W. Gata. (2021). PERANCANGAN VALIDASI PENGAJUAN E-KWITANSI ZISWAF PADA LEMBAGA AMIL ZAKAT MENGGUNAKAN KONSEP FINITE STATE AUTOMATA . Jurnal Ilmiah Ilmu Komputer, Fakultas Ilmu Komputer Universitas AL Asyariah Mandar Vol. 7, No. 2, September 2021

A. Aziz, N. Said, A. Sudrajat. (2020). Penerapan Konsep Finite State Automata Dalam Proses Pendaftaran Kelas Kursus Bahasa Inggris Pada Tempat Kursus . Jurnal Ilmu Komputer dan Teknologi Informasi, Vol.12, No.2, September 2020.

T. Rivanie, T. Adilah M, Y. Alkhalifi (2020). Implementasi Finite State Automata dalam Proses Registrasi Workout Plan Pada Pusat Kebugaran. Jurnal Ilmu Komputer dan Teknologi Informasi Volume 12, No. 1 (2020), pp 94-98 DOI : 10.18860/mat.v12i1.8573

R. A. Ma'arif, Fauziah. (2018). Implementasi Finite State Automata dalam Proses Pengisian Kartu Rencana Studi Mahasiswa . Journal of Information Technology and Computer Science Vol. 3, No. 3, September 2018

A. A. Puntambekar (2021). Theory of Computation, First edition: Januari 2021, hlm. 66.

X. Wang, Y. Hong, H. Chang, K. Park, G. Langdale, J. Hu, H. Zhu. 2019. Hyperscan: A Fast Multi-pattern Regex Matcher for Modern CPUs. Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2019 (2019) 631-648.

S. Khotijah. (2016). PERANCANGAN DATABASE E-LEARNING MANAJEMEN SYSTEM UNTUK PEMBELAJARAN PADA SEKOLAH MENENGAH PERTAMA. Jurnal String (Satuan Tulisan Riset dan Inovasi Teknologi) Vol. 1 No. 1 Tahun 2016

Suparyanto, Selo (2017). Simulator Pengenal StringYang Diterima Sebuah Deterministic Finite Automata (DFA). 9th National Conference on Information Technology and Electrical Engineering (CITEE) 2017. Yogyakarta, 27 Juli 2017.Komputika: Jurnal Sistem Komputer Volume 9, Nomor 1, April 2020, hlm. 75 – 83.

M. Alkhalaf, T. Bultan, J. L. Gallegos. (2012). Verifying Client-Side Input Validation Functions Using String Analysis. In 34th International Conference on Software Engineering (ICSE). Zurich, Switzerland, 2012. IEEE.DOI:10.1109/ICSE.2012.6227124 Stellenbosch, South Africa, August 24–26, 2015 Proceedings. Springer. DOI : 10.1007/978-3-319-23404-5.

T. Bultan. (2015). String Analysis for Vulnerability Detection and Repair. In 22 International Symposium, Spin 2015.

D. V. Paschchenko, D. A. Trokoz, A. I. Martyshkin, M. P. Sinev, B. L. Svistunov. 2020. Search for a substring of characters using the theory of non-deterministic finite automata and vector-character architecture. Bulletin of Electrical Engineering and Informatics. Vol. 9, No. 3, 1238-1250, June 2020, ISSN: 2302-9285, DOI : 10.11591/eei.v9i3.1720

H. N. Rosalia, Alamsyah. (2017) ANALISIS PENERAPAN SISTEM PENGELOLAAN PERSURATAN DALAM KEGIATAN TEMU KEMBALI ARSIP SURAT DI PT PELINDO III (PERSERO) CABANG TANJUNG EMAS SEMARANG. Jurnal Ilmu Perpustakaan Vol 6, No 1 (2017): Januari 2017

W. Pamulasari, N. Suryana. (2020). RANCANG BANGUN SISTEM INFORMASI MANAJEMEN SURAT BERBASIS WEB PADA KANTOR BPJS KETENAGAKERJAAN CABANG SUKABUMI. ENSAINS: Vol. 3 Nomor. 1 Januari 2020

M. Junus. (2018). SISTEM INFORMASI PENGELOLAAN SURAT MASUK & SURAT KELUAR JURUSAN TEKNIK ELEKTRO POLITEKNIK NEGERI MALANG BERBASIS WEB MELALUI JARINGAN INTRANET POLINEMA. Jurnal ELTEK, Vol 16 Nomor 02, Oktober 2018.

N. Suarna, S. Anwar, N. Rahaningsih. (2019). SISTEM INFORMASI MANAJEMEN PENGARSIPAN BERBASIS FRAMEWORK CODE IGNITER UNTUK MENTERTIBKAN PELAYANAN SURAT MENYURAT (STUDI KASUS : DINAS KEARSIPAN DAN PERPUSTAKAAN). Information System Journal (INTERNAL) Vol 2, No 1 (2019)