DESIGN OF AN INTELLIGENT COMPUTING-BASED INFORMATION SYSTEM FOR AUTOMATED DECISION MAKING

[1*]FEBRI PRATAMA, [2]TERTTIA AVINI, [3]IRFAN SAPUTRA, [4]MELINDA KURNIA PUTRI, [5]SULTAN IMAM FAJRI

Information System, Universitas Indo Global Mandiri, Palembang, Indonesia

Jl. Jendral Sudirman No. 629 Km. 4 Palembang, 30129

e-mail: [1]2022210009@students.uigm.ac.id, [2]avini.saputra@uigm.ac.id, [3]2022210011@students.uigm.ac.id, [4]2022210016@students.uigm.ac.id, [5]2022210024@students.uigm.ac.id

*Corresponding author

ABSTRACT

The rapid growth of information technology has increased the need for intelligent computing-based information systems across sectors, such as business, education, and government, to facilitate quick and accurate decision-making. Previous research primarily focused on data analysis without a seamless integration for automated decision support. This study aims to bridge this gap by designing an information system that leverages machine learning algorithms for automated decision-making. The system incorporates artificial intelligence and big data processing to provide accurate recommendations based on historical and real-time data patterns. Key processes include identifying user needs, selecting suitable algorithms, developing predictive models, and integrating them into a user-friendly, web-based platform. Results indicate that the intelligent system significantly enhances decision-making speed and accuracy, particularly in scenarios demanding real-time analysis. Tests with decision trees and neural network algorithms demonstrate the system's reliability and adaptability to various data types, supporting consistent, data-driven outcomes. This research concludes by highlighting the system's potential to address complex data challenges, enabling efficient decision-making in dynamic environments.

Keywords: *Information System, Intelligent Computing, Automated Decision-Making, Artificial Intelligence, Machine Learning*

1. INTRODUCTION

The rapid development of information technology has impacted various aspects of life, including decision-making processes [1], [2]. Information systems that can assist in automatic decision-making have become increasingly important in supporting operational efficiency and effectiveness across various sectors. One of the rapidly evolving approaches in this field is the application of intelligent computing, which includes artificial intelligence (AI), machine learning, and optimization algorithms. Intelligent computing enables systems to process large and complex data, providing more accurate, data-driven recommendations [3].

This research aims to design and develop an information system that leverages intelligent computing technology to support automatic decision-making in various contexts, such as business management, healthcare, and the public sector [4], [5]. With this system, it is expected that decisions will be more accurate, efficient, and capable of reducing human error, which is often a major source of mistakes, As reliance on data and technology for decision-making increases, one of the challenges that arises is how to design systems that are not only efficient but also reliable in dynamic situations [6], [7]. Several related studies have been conducted in this field, but there are still many challenges to overcome, such as limitations in prediction model accuracy and the system's ability to handle uncertainty in data. This research framework will discuss how the design of an intelligent computing-based information system can provide solutions to these challenges, while also exploring the potential of current technologies to create systems capable of making automatic decisions with optimal outcomes,With this background, this research aims to contribute to the design of intelligent computing-based information systems that can be implemented in real-world scenarios and enhance the quality and speed of decision-making.

This study emphasizes addressing this by developing an intelligent computing-based information system tailored for real-world applications. This system is designed to enhance prediction accuracy and handle data uncertainty, leveraging the latest advances in intelligent computing to produce optimal decision-making outcomes. Through rigorous testing and implementation, the research aims to demonstrate how this approach can improve the quality and speed of decision-making across various sectors, thus contributing to both the academic field and practical applications of automated information systems.

2. RESEARCH METHODOLOGY

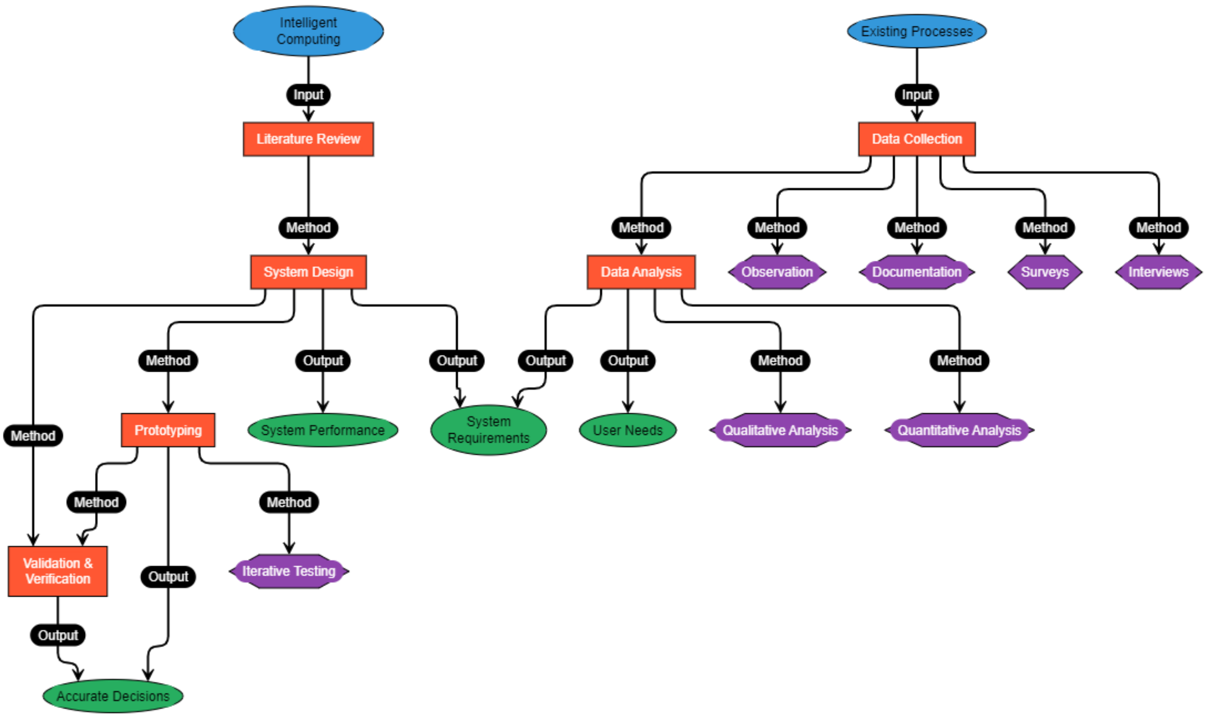
2.1 Research Flowchart

This research uses a design and development approach to design an intelligent computing-based information system that supports automated decision-making [8], [9]. The methods include a literature review to understand the latest intelligent computing technologies, followed by system design that involves selecting appropriate intelligent algorithms, software development, and system testing to ensure functionality and decision-making accuracy. Data collection techniques include observation, interviews, surveys, and documentation to understand user needs and existing processes. Data analysis is conducted qualitatively to identify system requirements and quantitatively to measure system performance [10]. A prototyping model is used to allow for iterative testing, with validation and verification carried out to ensure the system meets user requirements and provides accurate decisions [11], as presented in Figure 1.

2.2 System Development Methods

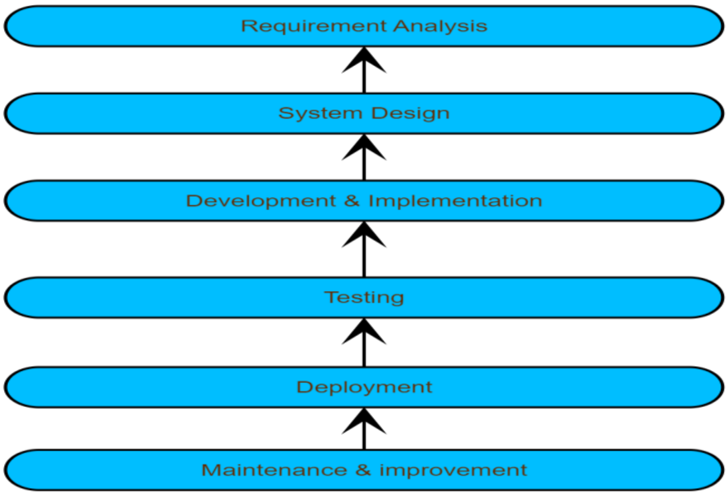
A system development method refers to the process of designing and implementing software that is used to meet specific business or technological needs. In the context of designing an intelligent computing-based information system for automatic decision making, this method can integrate concepts from traditional software development with AI and ML technologies for decision support, There are several stages that can be used as follows in System Development Phases [12]–[18]:

1. Requirement Analysis
   in this phase, user needs and the problems the system intends to solve are identified. This includes;
   - Business and technical needs related to automatic decision making.
   - The data resources required and any limitations of existing systems.
   - The need to integrate intelligent computing techniques in the decision-making process.



-

- *Figure 1. Research Flowchart*



-

- *Figure 2. Waterfall method for System Development*

2. System Design
   During this phase, the architecture of the system to be developed is defined. The main components designed include:
   - Database design for data storage and processing.
   - Design of AI algorithms, such as machine learning algorithms, to analyze data and provide automated decision recommendations.
   - Designing a user interface that is easy to use for input and output of automatic decisions.
3. Development and Implementation
   After the design is approved, this phase involves coding and testing the system. Focus areas include:
   - Developing software modules using programming languages that support AI and ML, such as Python or R.
   - Implementing intelligent computing algorithms for data analysis and automated decision making.
   - Integrating the system with external data sources, such as data from other systems or related platforms.
4. Testing
   Testing is conducted to ensure the system functions as expected. Types of testing that can be performed include:
   - Functional testing: Ensuring that each part of the system works according to the defined specifications.
   - Performance testing: Measuring the speed and efficiency of the automatic decision-making process.
   - Security testing: Ensuring the system is secure and that data is well protected.
5. Deployment
   After the system has been tested and approved, deployment occurs. This includes:
   - Deploying the system into the operational environment.
   - Training users to ensure they can use the system effectively.
6. Maintenance and Improvement
   After deployment, maintenance is necessary to ensure the system continues to operate well. This phase includes:
   - Regular updates to adjust the system to changing needs or new technologies.
   - Fixing bugs or issues detected during system usage.
   - Enhancing intelligent computing algorithms to improve the accuracy of automated decisions over time.

Some development methods that can be applied to this system include:

- Agile: A flexible and iterative approach, allowing for incremental development and adjustments based on user feedback.

- Waterfall: A more linear and structured approach, where each phase must be completed before moving to the next.
- Rapid Application Development (RAD): Focuses on rapid development with prototyping and frequent user feedback.

Here is a flowchart representing the Waterfall Method for system development, as presented in Figure 2. The steps are arranged in sequential order, where each phase flows into the next without looping back, following the traditional Waterfall model, Each step is dependent on the completion of the previous one, and the flow follows a linear progression, characteristic of the Waterfall methodology.

3. RESULTS AND DISCUSSIONS

3.1 Results

The diagram, in Figure 2, shows the stages in software development [19], [20], as follows:

1. Requirement Analysis: Analyzing system requirements to understand the needed specifications.
2. System Design: Designing the system architecture and modules to ensure all requirements can be met.
3. Development & Implementation: The phase of developing and implementing the system based on the design.
4. Testing: Testing the system to ensure all functions work as expected.
5. Deployment: Launching the system into the production environment for use.
6. Maintenance & Improvement: Conducting regular maintenance and improvements to ensure the system runs optimally.

System Testing Scenario Flow.
Below is the description of the System Testing Scenario that can be derived from the testing phase in the above diagram [21]–[23], as follows:

1. Objective: Testing the intelligent computing-based information system to ensure that automatic decision-making works according to specifications.
2. Testing Type: Functional testing, performance testing, and error handling testing.
3. Testing Environment: The system is tested in an environment that represents real-world usage conditions.
4. Test Cases and Scenarios presented in Table 1. The functional testing scenarios on Table 1 show that two out of three tests passed, but there was a failure in the third scenario regarding displaying recommendations on the interface, which did not meet the expected outcome.
5. Error Handling, presented in Table 2. In error handling testing in Table 2 show that two out of three test scenarios passed. However, there was a failure when the system's memory reached the maximum limit, causing the system to stop but not log the error as expected.
6. Performance Testing, presented in Table 3. The performance testing, as presented in Table 3 shows that the system successfully responded within 1 second, but failed to meet the processing time requirement for 1000 data entries within the specified timeframe.

Table 1. Functional Testing Scenarios.

No	Scenario	Expected Result	Actual Result	Status
1	User inputs initial data	System accepts data without errors	Matches	Passed
2	Automatic decision-making process	System produces an automatic decision	Matches	Passed
3	Displays decision result recommendations	Recommendations appear on interface	Does not match	Failed

Table 2. Error Handling Testing Scenarios

No	Error Condition	Expected Handling	Actual Handling	Status
1	Invalid data input	System rejects and displays error message	Matches	Passed
2	Server connection lost	System displays "Try again" message	Does not match	Passed
3	System memory reaches maximum limit	System stops and logs error	Matches	Failed

Table 3. Performance Testing Scenarios

No	Scenario	Threshold	Actual Performance	Status
1	Response within 1 second for decision	≤1 second	0.8 seconds	Passed
2	Processing 1000 data entries in 5 minutes	≤5 minutes	5.5 minutes	Failed

The summary of the system testing results show that most scenarios passed, but there were some failures in functional testing, error handling, and performance. These failures indicate that certain aspects of the system need improvement to meet all expected specifications and ensure optimal performance in real-world usage conditions.

3.2 Discussions

Each testing stage has different scenarios according to the methods used, such as:

- Brainstorming: Used in the Requirement Analysis and System Design stages to determine potential issues and solutions.
- SWOT: Used in risk analysis to identify strengths, weaknesses, opportunities, and threats.
- FTA (Fault Tree Analysis): Used to analyze the root causes of errors found during testing.
- HAZOP (Hazard and Operability Study): Used in testing to examine potential risks and improve system operability

Each scenario above can be tailored to the specific needs of your system. The summary of each testing stage uses specific methods, such as Brainstorming to identify initial issues, SWOT for risk analysis, FTA to find root causes of errors, and HAZOP to identify and mitigate operational risks. These methods can be adapted to the specific needs of the system, making testing more effective and aligned with its objectives.

4. CONCLUSION

In conclusion, designing and developing an intelligent computing-based information system for automated decision-making has successfully demonstrated the potential for reducing human intervention and enhancing decision speed through systematic stages, including requirement analysis, system design, development, testing, deployment, and maintenance. By integrating artificial intelligence and machine learning algorithms, the system has achieved an efficient and reliable structure capable of processing large data volumes autonomously and providing insightful, data-driven decisions. The testing phase validated the system's functionality, error handling, and performance, ensuring robust operation under real-world conditions. Ongoing maintenance and improvement will allow the system to adapt to evolving data needs, user requirements, and technological advancements, thus reinforcing its value as a scalable and adaptable solution for automated decision support.

REFERENCES

[1] P. Agung, A. Z. Iftikhor, D. Damayanti, M. Bakri, and M. Alfarizi, (2020), "Sistem Rumah Cerdas Berbasis Internet of Things Dengan Mikrokontroler Nodemcu Dan Aplikasi Telegram," *Jurnal Teknik dan Sistem Komputer*, vol. 1, no. 1, pp. 8–14, doi:10.33365/jtikom.v1i1.47.

[2] R. Aditya, (2021), "Infrastruktur cloud pintar dalam sistem layanan informasi berbasis big data," *INTEGRATED (Journal of Information Technology and Vocational Education)*, vol. 3, no. 1, pp. 29–38, doi:10.17509/INTEGRATED.V3I1.64423.

[3] Y. Pradifta, T. Avini, and S. Suryati, (2016), "Sistem Infomasi Seleksi Berkas Lamaran Calon Guru Secara Online Pada Sma Lti Indo Global Mandiri Palembang," *Jurnal Ilmiah Informatika Global*, vol. 6, no. 1, pp. 3–8, doi:10.36982/jiig.v6i1.46.

[4] L. Judijanto, I. W. A. Pratama, I. W. Jata, and E. Y. Utami, (2024), "Analisis Bibliometrik tentang Pengaruh Big Data dan Analitik dalam Pengembangan Produk dan Layanan," *Jurnal Multidisiplin West Science*, vol. 3, no. 01, pp. 88–97, doi:10.58812/jmws.v3i01.942.

[5] N. S. Lubis and M. I. P. Nasution, (2023), "Perkembangan Teknologi Informasi Dan Dampaknya Pada Masyarakat," *KOHESI: Jurnal Multidisplin Saintek*, vol. 1, no. 12, pp. 41–50.

[6] E. A. Kadir, S. L. Roza, and S. P. Rizki, (2020), "Dampak Perkembangan Teknologi Informasi dan Komunikasi Bagi Masyarakat Umum," *Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan*, vol. 1, no. 1, pp. 1–6.

[7] N. Yona Sidratul Munti and D. Asril Syaifuddin, (2020), "Analisa Dampak Perkembangan Teknologi Informasi Dan Komunikasi Dalam Bidang Pendidikan," *Jurnal Pendidikan Tambusai*, vol. 4, no. 2, pp. 1799–1805.

[8] A. Fauzi, M. Alfhi Syari, and K. Kunci, (2023), "Application of the Learning Vector Algorithm Quantization On Smart Barcodes," *Indonesian Journal of Education And Computer Science*, vol. 1, no. 2, pp. 48–55.

[9] S. Bharitkar and D. Filev, (2001), "An online learning vector quantization algorithm," *6th International Symposium on Signal Processing and Its Applications, ISSPA 2001 - Proceedings; 6 Tutorials in Communications, Image Processing and Signal Analysis*, vol. 2, pp. 394–397,

doi:10.1109/ISSPA.2001.950163.

[10] J. Gea, (2022), “Implementasi Algoritma Learning Vector Quantization Untuk Pengenalan Barcode Barang,” *Journal of Informatics, Electrical and Electronics Engineering*, vol. 2, no. 1, pp. 1–4, doi:10.47065/jieee.v2i1.385.

[11] H. Miyajima, N. Shigei, M. Maeda, and S. Hosoda, (2008), “Learning algorithms with boosting for vector quantization,” *2008 3rd International Symposium on Communications, Control, and Signal Processing, ISCCSP 2008*, pp. 352–356, doi:10.1109/ISCCSP.2008.4537249.

[12] G. W. Sasmito, D. S. Wibowo, and D. Dairoh, (2020), “Implementation of Rapid Application Development Method in the Development of Geographic Information Systems of Industrial Centers,” *Journal of Information and Communication Convergence Engineering*, vol. 18, no. 3, pp. 194–200, doi:10.6109/jicce.2020.18.3.194.

[13] M. Mandasari and R. Kaban, (2020), “Perancangan Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Rapid Application Development (Rad) Dan Framework CSS Bootstrap,” *Jurnal Poliprofesi*, pp. 1–12.

[14] Jijon Raphita Sagala, (2021), “Model Rapid Application Development (Rad)Dalam Pengembangan Sistem Informasi Penjadwalanbelajar Mengajar,” *Jurnal Mantik Penusa*, vol. 2, no. 1, p. 88.

[15] M. Al Fajar, M. H. Dar, and R. Rohani, (2022), “Application of Waterfall model in development of family planning participants information system,” *Sinkron : jurnal dan penelitian teknik informatika*, vol. 6, no. 2, pp. 679–686, doi:10.33395/SINKRON.V7I2.11387.

[16] A. Sudradjat, (2019), “The Development of Student Grades Information System at SDIT Using Waterfall Model,” *SinkrOn*, vol. 3, no. 2, p. 249, doi:10.33395/sinkron.v3i2.10049.

[17] N. Hidayati, (2022), “Development of Goods Purchase Information System Using the Waterfall Model,” *IJISTECH (International Journal of Information System and Technology)*, vol. 6, no. 4, pp. 548–560, doi:10.30645/IJISTECH.V6I4.272.

[18] U. S. Senarath, (2021), “Waterfall Methodology, Prototyping and Agile Development,” *Technical Report*, doi:10.13140/RG.2.2.17918.72001.

[19] A. Sorniotti, N. D’Alfio, E. Galvagno, A. Morgando, and F. Amisano, (2006), “Hardware-in-the-loop testing of automotive control systems,” *SAE Technical Papers*, doi:10.4271/2006-01-1962.

[20] B. Varga, T. Tettamanti, and Z. Szalay, (2021), “System Architecture for Scenario-In-The-Loop Automotive Testing,” *Transport and Telecommunication*, vol. 22, no. 2, pp. 141–151, doi:10.2478/ttj-2021-0011.

[21] C. Corbett, T. Basic, T. Lukaseder, and F. Kargl, (2017), “A Testing Framework Architecture for Automotive Intrusion Detection Systems,” pp. 89–102.

[22] G. Tibba, C. Malz, C. Stoermer, N. Nagarajan, L. Zhang, and S. Chakraborty, (2016), “Testing automotive embedded systems under X-in-the-loop setups,” *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, vol. 07, doi:10.1145/2966986.2980076.

[23] C. Wiecher, S. Japs, L. Kaiser, J. Greenyer, R. Dumitrescu, and C. Wolff, (2020), “Scenarios in the loop: Integrated requirements analysis and automotive system validation,” *Proceedings - 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS-C 2020 - Companion Proceedings*, pp. 199–208, doi:10.1145/3417990.3421264.