Schedule Information System of Medical Profession Program

Zaidan Alvin Al Hanif
Electrical Engineering Department
Sebelas Maret University
Surakarta, Indonesia
zaidanalvin@student.uns.ac.id

Faisal Rahutomo*
Electrical Engineering Department
Sebelas Maret University
Surakarta, Indonesia
faisal_r@staff.uns.ac.id

Meiyanto Eko Sulistyo
Electrical Engineering Department
Sebelas Maret University
Surakarta, Indonesia
mekosulistyo@staff.uns.ac.id

*Corresponding author: faisal_r@staff.uns.ac.id

Received: 2025-05-29, Year; **Accepted:** 2026-01-21

***Abstract*—This article details the development of an automated, web-based scheduling system for medical professional education at UNS Surakarta, Indonesia. The goal of this research was to develop an information system that assists UNS medical faculty administrators by automating the scheduling of medical professional education and enabling fast, precise, and efficient access to schedules. This study employed a prototyping method, sequentially defining the objective object (system requirements), designing, implementing, and evaluating the system. The prototyping method focuses on determining the general purpose of the system, as this prototype serves as an initial description of the system for further development. Several diagrams, such as ERD and use case diagrams, describe the system architecture design. The team performed system testing using the black box method to ensure the system functions as designed. The results show that the system can produce clinical clerkship scheduling more quickly and effectively. After testing the black box system, a user acceptance test was conducted through a satisfaction survey, yielding an assessment result of 76.9%. This system's user satisfaction was relatively high.**

Keywords—information system, laravel, scheduling, web-based

I. Introduction

Health services are required to continually improve the quality of service to the community. Improvements can be made by enhancing the infrastructure and training health workers. The doctor is one of the health workers who determines the quality of a health service. A professional doctor's training involves four stages. The first stage involves academic education, carried out by the medical faculty, until the completion of a bachelor's degree. The second stage is medical professional education (clinical clerkship) in related hospitals and health centers. This stage lasts approximately 1.5 to 2 years and encompasses 14 to 15 stages. After completing their professional education, prospective doctors take a competency test for doctoral professional program students (UKMPPD) and take the Hippocratic Oath. In the third stage, prospective doctors intern at a hospital or health center for a period of one year. The fourth stage involves following a specialist school in the chosen field.

The clinical clerkship program is designed to provide medical interns with hands-on experience in treating patients through real cases, as their services are continuous and ongoing. Scheduling for the clinical clerkship involves placing medical interns at various stations for specified periods. The scheduling administrator collects information, including the number of interns, the stations that require coverage, and the scheduled times for each station. Previously, the UNS Faculty of Medicine manually scheduled the clinical clerkship program using Microsoft Excel. The scheduling administrator must plan each week to adjust for any vacant station openings. [1].

Scheduling in the medical profession program is unique. Students study in a block system, progressing from one section to another. Each incoming group of students is distributed across all existing blocks and then transitioned according to a predetermined pattern. The duration of study in each section varies, leading to an overlap of students within those sections. As a result, each section includes both senior and junior students.

Preparing the schedule for stations in medical professional education using manual methods requires significant accuracy and patience, as it can be a time-consuming process. There is a large amount of data to manage related to young doctors and the stations that need to be scheduled. Additionally, various factors can impact the scheduling process, such as lecturers' availability and willingness to teach on specific days, often resulting in scheduling conflicts. To address these challenges, we are researching the development of an automated information system for scheduling in professional medical education. In this study, the authors employed the prototyping method to create the information system. [2], [3]. The presentation in this paper aims to address real-world problems, as in previous work. [4], [5].

II. Literature Review

A. *Information System*

An information system is a structured system designed to collect, store, process, analyze, and disseminate information for a specific purpose. Within an organization, it integrates various needs to support operations and transaction processing, while also serving managerial functions and aligning with the organization's strategy. Additionally, it generates multiple reports and shares information with external parties. Essentially, an information system comprises interconnected components that work together to collect, process, store, and distribute information, thereby supporting the organization's activities. [6].

B. *Model View Controller (MVC)*

The model-view-controller (MVC) architecture is commonly used in web application frameworks. The model module comprises classes that represent tables in the database, enabling the manipulation of the database. The model serves as a connection between the controller and the

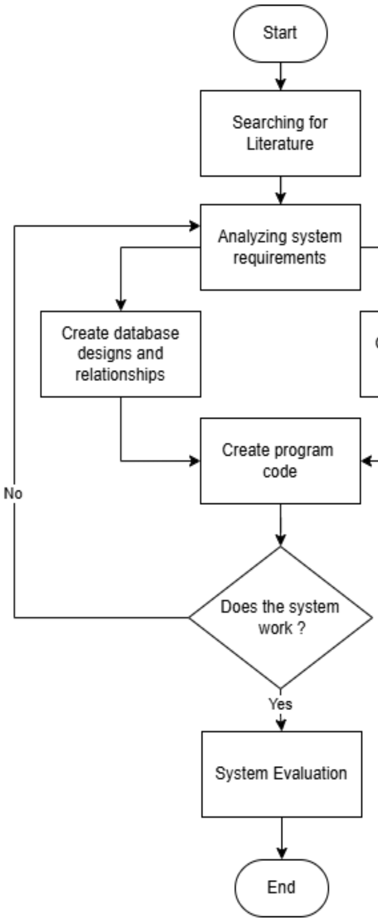

Fig. 1. Research flowchart

database when retrieving data. The controller is a class written by programmers that handles the system's logic. It acts as the brain of the system, functioning as a bridge between the model and the view. The view is responsible for displaying the data that the controller sends. One of the key advantages of the MVC architecture is that it separates models and views, allowing the same model to be utilized by multiple views. This separation makes it easier to implement, test, and maintain models in applications since all interactions with the model pass through the MVC components. Overall, the MVC concept enhances flexibility and reduces complexity in software development. The following illustrates the MVC concept [7].

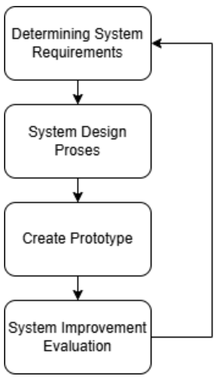

Fig. 2. Prototyping flowchart

C. *Entity Relationship Diagram (ERD)*

An Entity-Relationship Diagram (ERD) is a key data modeling tool that organizes data in a project into distinct entities and defines the relationships between them. An entity represents a real-world object or concept that can be distinguished from others, while a relationship illustrates how these entities interact with one another. [8]. Utilizing Entity-Relationship Diagrams simplifies the analysis of a system's requirements, making it easier and faster to design and implement. Several types of relationships can exist between entities. These relationships help to clarify how data is structured and how different entities interact within a system:

- One-to-One Relationship: This occurs when a single record in one entity is related to a single record in another entity.
- One-to-Many Relationship: This relationship exists when a single record in one entity can be associated with multiple records in another entity.
- Many-to-Many Relationship: In this case, multiple records in one entity can be linked to multiple records in another entity.

D. *Black Box Testing*

Black box testing is a software testing technique that focuses on evaluating the software's functionality without examining its internal code structure. [9]. This testing is based on the software's specifications and requirements. Black box testing is particularly effective for rapid testing, especially in fast-paced development environments such as those using the prototyping method. The primary purpose of black box testing is to identify issues such as user interface errors, performance problems, and incorrect system functions. One standard method used within black box testing is the equivalence partitioning technique. This technique involves testing input data by grouping each menu based on its function, which categorizes values as either valid or invalid.

E. *Prototyping Method*

Prototyping is a software development method that involves creating a working model of a system as an initial version. There are four main types of prototyping methods. Illustrative: This method produces research reports. Simulated: This approach simulates the system's workflow without using actual data. Functional: This method simulates the system’s workflow using real data. Evolutionary: This creates a model of the operational system. [10].

The prototyping method consists of several essential steps. Gathering Requirements: This involves collaboration between developers and users to collect the system's requirements. Fast Design Process: This step focuses on representing various aspects of the software from the user’s perspective. Building Prototypes: Once the design is complete, prototypes are developed based on the gathered requirements. Evaluation: After building the prototype, it is evaluated to assess its effectiveness and areas for improvement. Improvement: Final adjustments are made based on the assessment.

At the end of this process, a flow diagram illustrating the prototyping method can be created to visualize the steps involved.

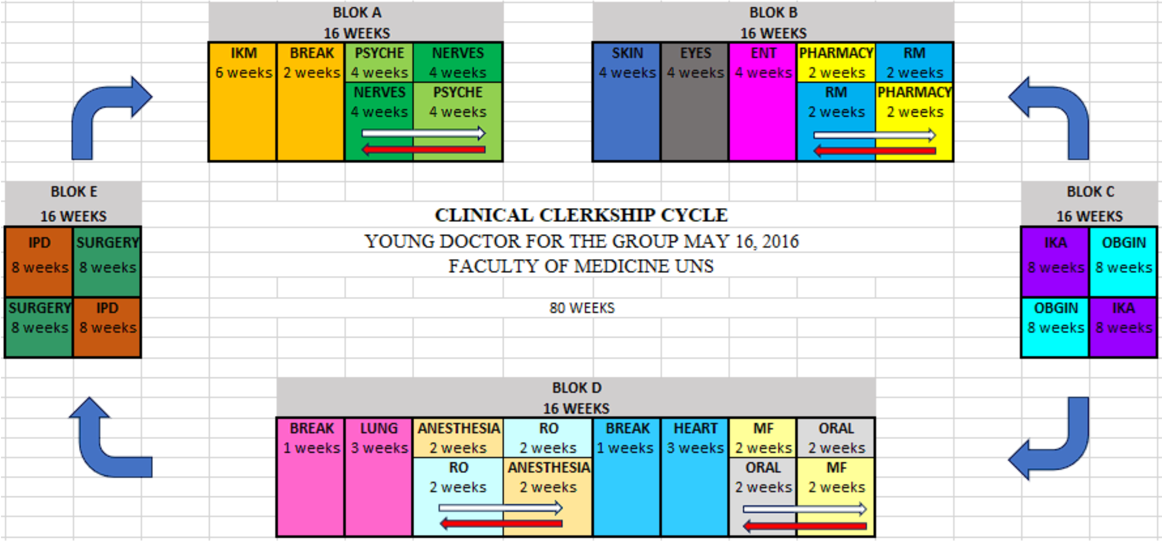

Fig. 3. First stage clinical clerkship cycle

III. Research Method

In this study, we have chosen the prototyping method for designing and developing information system software. The purpose of using this method is to create a representation and model of the information system that will be developed. The study involves several steps, which are illustrated in Fig.2, outlining the process of software creation.

A. *Analysis*

There are two types of software requirements specifications used to support this research, namely functional and non-functional [11], [12], [13]. The hardware used in this research is an Acer Aspire ES14 laptop. The software used in this research includes Windows 10, Microsoft Office, Visual Studio Code, Navicat, XAMPP, Laravel framework, Various Browsers, PHP, JavaScript, HTML, CSS, and MySQL.

Based on interviews conducted with administrators and discussions with supervisors, the following findings have been established regarding the scheduling of medical professional education at FK UNS. The study program administrator is responsible for this process, which begins with collecting data on students who will participate in the educational activities from the academic department. Once the data is obtained, scheduling is done manually using Excel. Weekly scheduling depends on the availability of stations, making it impossible to create a comprehensive schedule for all stations at once due to resource limitations. To improve the scheduling process, innovation is needed. An efficient scheduling system would streamline operations, minimize human error, and generate a complete schedule in one go.

Medical professional education consists of two implementation stages. In Stage 1, students must complete 18 different stations: IKM, psychiatry, neurology, dermatology, ophthalmology, ENT, pharmacy, community medicine (RM), nursing (IKA), obstetrics, oral dentistry, microbiology (MF), cardiology, radiology (RO), anesthesia, pulmonology, surgery, and internal medicine (IPD). Stage 1 spans 80 weeks, while Stage 2 requires an additional 24 weeks to complete five stages: the Emergency Room, UKMPPD, PKM, laboratory, ward, and PMI.

During the first phase of the clinical clerkship cycle at the Medical Faculty of UNS, which lasts 80 weeks, there are 18 stages divided into five main blocks: A, B, C, D, and E. Each block is assigned a 16-week duration with different stations. The block includes subjects such as IKM, psychiatry, and neurology, and it also features a two-week break. Block B covers dermatology, ophthalmology, ENT, pharmacy, and RM. Block C features nursing (IKA) and obstetrics. Block D involves oral dentistry, microbiology (MF), cardiology, radiology (RO), anesthesia, pulmonology, and vacation time. Block E consists of surgery and internal medicine (IPD). The rotation process for the first stage operates in a clockwise manner; as each station is completed, participants move to the next station on the right.

The distribution process for the first stage involves gathering information from all students registered for this phase from the academic section of the medical faculty. The students are then evenly divided into blocks to determine their starting points. Approximately 60 student candidates participate in each clinical clerkship session at UNS.

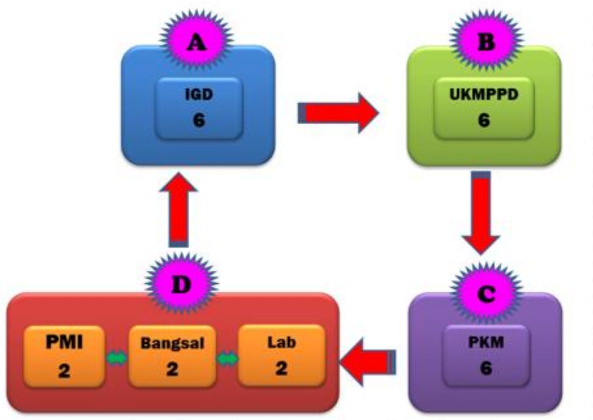

Fig. 4. Second stage clinical clerkship cycle

Journal of Electrical, Electronic, Information, and Communication Technology (JEEICT)
Vol. 08 No. 1, April-2026, Pages 1-5 DOI: https://dx.doi.org/10.20961/jeeict.8.1.103117

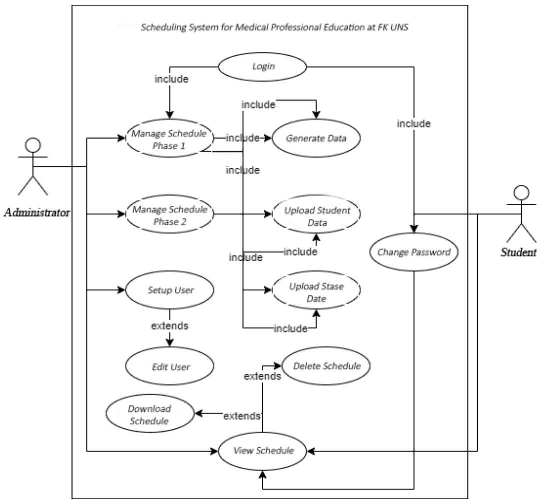

Fig. 5. Use case diagram

In the second stage of the clinical clerkship cycle, students must complete four blocks within approximately 24 weeks. Block A is assigned to the Emergency Room (IGD), Block B to UKMPPD, Block C to PKM, and Block D includes the laboratory, ward, and PMI stations. The rotation for this stage also proceeds in a clockwise direction.

A significant challenge in scheduling clinical clerkships arises from the increasing number of students. At the same time, the administrative staff responsible for scheduling remains limited to a single person, and the process continues to rely on manual Excel methods. Manual scheduling, as demonstrated, requires high accuracy and can be a time-consuming process. Interviews with the administrator have revealed that this manual process occurs weekly, which prevents the completion of a thorough schedule throughout the entire clinical clerkship period. Therefore, there is a pressing need for a system designed to expedite the scheduling process and reduce the potential for human errors.

B. Use Case

The figure illustrates two types of roles that will be implemented on the website: study program administrators and users. [14]. User Role: Users have access to several features, including the ability to log in and out, view their results for Stage 1 scheduling, and see their results for Stage 2 scheduling. Admin Role: Administrators have a broader range of features and capabilities. These include:

- Logging in and out.
- Uploading user data, allowing users to log in and out.
- Uploading student data for Stage 1 to facilitate scheduling.
- Uploading dates for Stage 1 stations to meet scheduling needs.
- Generating schedules for Stage 1 and viewing the results of Stage 1 scheduling.
- Deleting Stage 1 scheduling results.

In Stage 2, administrators can perform several additional functions:

TABLE I. CHARACTERISTIC OF EACH USERS

Role	Characteristics
Administrator	Users who do data processing such as uploading student data, uploading stase dates, and generating data to produce a doctor's professional education schedule.
Student	Users who view the results of doctor's professional education scheduling

- Uploading student data for Stage 2 scheduling.
- Uploading dates for Stage 2.
- Generating a schedule for Stage 2.
- Viewing the results of Stage 2 scheduling.
- Deleting Stage 2 scheduling results.

C. Database

In this study, the scheduling process is divided into two stages. The first stage utilizes database tables, as illustrated in the accompanying picture. Several key relationships exist among these tables. The roles table and the user table are connected by a column named role_id. The coass_olah table and the coass_date table are linked through a column named id_week. The coass_olah table is also related to the coass_mhs table, which relates via the id_mhs column. Additionally, the coass_olah table is associated with the coass_stase table through the id_stase column. [8].

In the second stage of scheduling, multiple relationships exist among the tables as well. The coass_date2 table is linked to the coass_olah2 table through a column named id_week. The coass_mhs2 table is related to the coass_olah2

TABLE II. DESCRIPTION OF EACH ROLE

Role	Menu	Description
Administrator	Manage student data phase 1	This menu has various functions, namely uploading student data, uploading stase dates and generating data to perform phase 1 scheduling
	Manage student data phase 2	This menu has various functions, namely uploading student data, uploading stase dates and generating data to perform phase 2 scheduling
	Setup User	This menu adds and edits user data for login purposes
	View schedule phase 1	This menu displays the overall phase 1 scheduling results
	View schedule phase 2	This menu displays the overall phase 2 scheduling results
	Guide	This menu contains guide to use the scheduling application
Student	View student schedule phase 1	This menu displays the results of pahse 1 scheduling for the student concerned
	View student schedule phase 2	This menu displays the results of pahse 2 scheduling for the student concerned
	Change password	This menu changes the default password given

Journal of Electrical, Electronic, Information, and Communication Technology (JEEICT)
Vol. 08 No. 1, April-2026, Pages 1-5 DOI: https://dx.doi.org/10.20961/jeeict.8.1.103117

table via the id_mhs column. Lastly, the coass_stase2 table is connected to the coass_olah2 table through the id_stase column. This structured approach highlights the interconnectedness of the various tables within the scheduling framework.

D. System Workflow Flowchart

The workflow of this system begins with the user logging into the website. Once logged in, the system verifies whether the data entered by the user is correct. If the data is accurate, the user is directed to the dashboard page, and their login information is saved, so they do not need to log in again.

For users with an admin role, the process includes logging in, uploading student data, and specifying the start date for the clinical clerkship. After completing these uploads, the admin generates a schedule to obtain the scheduling results.

On the dashboard page, users can access features to view the results of clinical clerkship scheduling, available for both admin and student roles. Admin users have additional features that allow them to add, delete, and generate scheduling results data.

Finally, when the user logs out of the system, it checks for any errors that may have occurred during their session. If no errors are found, the system deletes the user data from the active session and completes the logout process. If an error occurs, the system will display an error message.

IV. RESULTS AND DISCUSSION

This chapter describes the testing results of the system. Several tests were conducted: a functional test, a speed performance test, and an acceptance test.

A. Functional Test

Based on the user requirements, this system implements all features to address all specified needs. It allows for the addition of student data to facilitate the processing of clinical clerkship schedules. The system can input student data information, which will later be processed into scheduling results. Additionally, the system can generate output based on the data entered, producing the final scheduling results. Users have the option to delete any previously generated scheduling results and can also display past results.

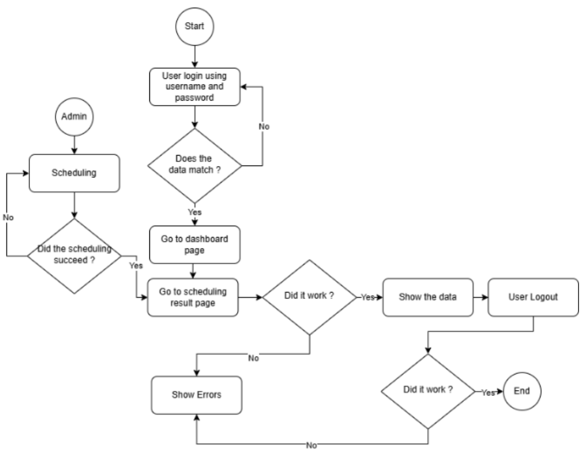

Fig. 6. System workflow flowchart

According to the use case diagram, there are two roles within the information system. Table 3 outlines the feature tests associated with each role.

- Login Page: Users must log in by entering the email and password registered by the admin before they can access any system features. The following illustrates the login page.
- Menu “Stage 1 scheduling”: This page showcases student data for those participating in Phase 1 of the clerkship. It contains various data processing features for scheduling Phase 1, such as uploading student data, uploading station dates, and generating schedules based on the uploaded information. The following displays the data processing menu for Stage 1.
- Menu “Stage 2 scheduling”: This page presents student data for Phase 2 of the clerkship, including features for processing data related to Phase 2 scheduling. Similar to the previous menu, it allows for uploading student data, uploading station dates, and generating schedules based on the uploaded information. The following shows the data processing menu for Stage 2.
- Menu “Scheduling result”: Figure X demonstrates a test conducted to verify whether the scheduling result data can be displayed.

The results indicate that the system successfully displays the scheduling results, confirming that this feature is functioning correctly.

B. Speed and Effectiveness of Scheduling Processes

To determine the effectiveness of this scheduling system,

TABLE III. RESULT OF FUNCTIONAL TEST

Role	**Feature**	**Sub-Feature**	**Test Result**
Administrator and Student	Login		Success
Administrator	Phase1 Scheduling	Upload student data phase 1	Success
		Upload stase date phase 1	Success
		Generate data phase 1	Success
		View schedule phase 1	Success
		Delete schedule phase 1	Success
	Phase 2 Scheduling	Upload student data phase 2	Success
		Upload stase date phase 2	Success
		Generate data phase 2	Success
		View schedule phase 2	Success
		Delete schedule phase 2	Success
	Setup User		Success
Student	View Scheduling Result	View student schedule phase 1	Success
		Change password	Success
		View student schedule phase 2	Success
Administrator and Student	Logout		Success

it is necessary to conduct field observations and discuss the

Journal of Electrical, Electronic, Information, and Communication Technology (JEEICT)
Vol. 08 No. 1, April-2026, Pages 1-5 DOI: https://dx.doi.org/10.20961/jeeict.8.1.103117

TABLE IV. ACCEPTANCE TEST

Question	5	4	3	2	1
Is this system easy to operate?	4	7	1	1	0
The user interface of this system is attractive	6	3	1	3	0
The features in this system are useful	4	7	1	1	0
Overall satisfaction using this system	3	7	1	2	0

process with end users, specifically the Faculty of Medicine study program administrators and students. Observations indicated that the scheduling process using this information system is significantly faster than the manual method previously employed with Excel. This system can generate a complete schedule for young doctors—from the initial station to the final station—at one time. In contrast, the manual process cannot produce a full schedule in a single attempt; it requires weekly scheduling and cross-checking to minimize human error. Based on the observations, this system proves to be effective for scheduling. The data from the scheduling results demonstrates that the system completes scheduling from the first station to the last station, along with the corresponding results produced by the system.

C. Acceptance Test

User acceptance testing involves asking users of the system four questions. Their responses are recorded on a scale from 1 to 5, where 5 indicates the highest level of satisfaction, and 1 indicates the lowest level of dissatisfaction. Table 4 shows the results. The majority of users reported being satisfied with the system, with only 1 or 2 respondents expressing dissatisfaction.

V. CONCLUSION

The design of an information system for scheduling medical professional programs is developed with careful consideration of user requirements established at the outset. These requirements ensure that the system meets all necessary functionality. The system includes a guide to the scheduling software, making it easier for users to navigate and utilize it effectively. It features tools that assist in scheduling Phases 1 and 2, eliminating the need for administrators to schedule using Microsoft Excel manually. This enhancement not only speeds up the scheduling process but also reduces the risk of inaccuracies associated with manual methods, allowing for a quicker and more efficient scheduling experience. Testing the system using the black box method reveals that each scheduling feature for Phases 1 and 2 is functioning correctly. This includes capabilities for uploading student data, inputting station dates, generating schedules, displaying scheduling results, and deleting completed scheduling records.

ACKNOWLEDGMENT

This research is supported by funding from the research and community service institution of the applied research scheme of Sebelas Maret University, with contract number 228/UN27.22/PT.01.03/2023 "Development of academic and student data analytics applications".

REFERENCES

[1] Indonesian Medical Council, *Standar Pendidikan Profesi Dokter Indonesia*. Jakarta: Konsil Kedokteran Indonesia, 2021.

[2] I. Sommerville, *Software Engineering*, 9th ed. Harlow, England: Addison-Wesley, 2010.

[3] R. Pressman, *Software Engineering: A Practitioner's Approach*, 7th ed. New York, NY, USA: McGraw-Hill, Inc., 2010.

[4] F. Rahutomo, A. S. Mulyo, and P. Y. Saputra, "Automatic Grammar Checking System For Indonesian," in *International Conference on Applied Science and Technology (iCAST)*, Manado, 2018.

[5] Moch. F. Shadiqin Thirafi and F. Rahutomo, "Implementation of Naïve Bayes Classifier Algorithm to Categorize Indonesian Song Lyrics Based on Age," in *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2018, pp. 106–109. doi: 10.1109/SIET.2018.8693201.

[6] K. C. Laudon and J. P. Laudon, *Management Information Systems: Managing the Digital Firm*, 13th ed. Boston: Pearson, 2013.

[7] S. Necula, "Exploring The Model-View-Controller (MVC) Architecture: A Broad Analysis of Market and Technological Applications," *Preprints (Basel)*, Apr. 2024, doi: 10.20944/preprints202404.1860.v1.

[8] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 2nd ed. USA: McGraw-Hill, Inc., 2000.

[9] A. Kusuma and N. Hadinata, "The Implementation of the Black Box Method for Testing Smart Hajj Application Ministry of Religion," *Journal of Information Systems and Informatics*, vol. 4, no. 3, pp. 673–686, Sep. 2022, doi: 10.51519/journalisi.v4i3.306.

[10] E. Bjarnason, F. Lang, and A. Mjöberg, "An empirically based model of software prototyping: a mapping study and a multi-case study," *Empir Softw Eng*, vol. 28, no. 5, p. 115, 2023, doi: 10.1007/s10664-023-10331-w.

[11] S. Saroja and S. Haseena, "Functional and Non-Functional Requirements in Agile Software Development," in *Agile Software Development*, John Wiley & Sons, Ltd, 2023, ch. 5, pp. 71–86. doi:/doi.org/10.1002/9781119896838.ch5.

[12] S. Sindhgatta Renuka and Thonse, "Functional and Non-functional Requirements Specification for Enterprise Applications," in *Product Focused Software Process Improvement*, S. Bomarius Frank and Komi-Sirviö, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 189–201.

[13] D. Dave and V. Anu, "Identifying Functional and Non-functional Software Requirements From User App Reviews," in *2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2022, pp. 1–6. doi: 10.1109/IEMTRONICS55184.2022.9795770.

[14] J. L. Barros-Justo, F. B. V Benitti, and S. Tiwari, "The impact of Use Cases in real-world software development projects: A systematic mapping study," *Comput Stand Interfaces*, vol. 66, p. 103362, 2019, doi: doi.org/10.1016/j.csi.2019.103362.