Web-Based WhatsApp Blast Application for Automatic Information Dissemination at PT PLN UP3 Singkawang

Alanda Randa[1)*], Muhammad Usman, S.T., M.Kom[2)], Erifa Syahnaz S.Kom. M.M[3)], Heldi Hastriyandi S.T., M.Kom[4)]
Manajemen Informatika, Politeknik Negeri Sambas
* Corresponding author : alandaranda763@gmail.com[1], usman.mtaib2@gmail.com[2], syahnazerifa@gmail.com[3], heldi.poltesa@gmail.com[4]

Abstract

PT PLN (Persero) UP3 Singkawang previously relied on a third-party WhatsApp service with limited features and lacked centralized customer data storage, resulting in high operational costs and inefficient manual broadcasting. This research aims to design and build an independent web-based WhatsApp Blast application to automate customer service information dissemination and operational report management. The research methodology follows a sequential Waterfall model, comprising five systematic stages: requirements analysis of previous broadcasting constraints, system design using UML architecture and MySQL database, core implementation via the Python Flask framework, Wablas API integration for messaging automation alongside APScheduler, and comprehensive black-box testing. Quantitative evaluation results demonstrate that the application successfully processed 6,203 customer records, achieved a 97.4% message delivery success rate for verified numbers, and reduced report compilation time from 120 minutes to under 5 seconds—representing a 95.8% time-efficiency increase while eliminating third-party dependency.

Keywords- WhatsApp Blast, Flask, Wablas API, Waterfall Method, PLN UP3 Singkawang

1. INTRODUCTION

State Electricity Company (PLN) UP3 Singkawang has a big responsibility in providing electricity services and distributing various information to the public in six Customer Service Unit (ULP) areas. The information distributed includes promotions of superior programs, notification of planned network maintenance, information on electricity safety (K2), and education on the use of the PLN Mobile application. Currently, PLN UP3 Singkawang is utilizing the WhatsApp Blast service from a third party to disseminate this information en masse.

However, in its implementation, the company still relies on third party services which have limited features and are not fully relevant to the unit's specific needs. Sending messages is carried out by preparing recipient number data manually using Microsoft Excel because it is not yet stored in a centralized database, so officers have to re-enter it every time a message is sent. Another operational obstacle is that the WhatsApp blasting service previously used required quite a large fee with a cost calculation scheme based on the quantity of messages sent to customers. Considering the large number of customers, this condition causes operational costs to be high. As a result, the process of disseminating information becomes less efficient and is very dependent on the availability of officers in the field.

Apart from obstacles in information distribution, the process of processing operational target realization reports such as Independent Meter Reading (BMM), Swacam, and Meter Reading Route Codes (RBM) is also still done manually using the Microsoft Excel application which uses various calculation formulas. To compile the report, officers must first access the Centralized Meter Recording Application (ACMT) website to download the data via a menu tailored to the type of report required. ACMT is an application installed on the smartphones of PT PLN (Persero) meter readers; it enables the accurate, digital reading and recording of electricity meters, capturing photographic evidence and GPS tags that are automatically integrated with PT PLN (Persero)'s core system, AP2T (Centralized Customer Service Application) [1]. The data that has been downloaded is then transferred and reprocessed in Microsoft Excel through a series of stages starting from the data input process, preparing the report structure, to calculating using

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

certain formulas. This process, which is still manual, requires quite a long time and is prone to causing various obstacles, such as the high risk of input errors (human error), duplication of data, and lack of time efficiency in preparing reports. After the report has been processed, the results still have to be sent manually to the PLN UP3 Singkawang internal WhatsApp group to then be checked directly by management or superiors as material for monitoring and evaluating performance. This condition causes delays in submitting reports which has a direct impact on the smoothness of the monitoring process and strategic decision making.

Based on these problems, this research aims to design and build an independently integrated web-based WhatsApp Blast application to automate the dissemination of information and management of operational reports in the PLN UP3 Singkawang environment. This digital communication system was developed using the Python programming language with the Flask framework, and utilizes the Application Programming Interface (API) from the Wablas service as the main engine for sending messages. This application is equipped with a number of supporting features, including management of contact databases and message templates, automatic delivery scheduling, real-time monitoring of message status, automation of sending operational reports (BMM, Swacam, and RBM), as well as simple interactive chatbot features for automated services.

System development and problem solving in this research were carried out sequentially using the Waterfall method. This approach includes the stages of analyzing operational needs related to previous blasting service constraints, designing web architecture and MySQL databases, system development through coding, testing mass delivery functionality and automatic scheduling, to the system maintenance stage. Specifically, this application is designed to be implemented in the field of Electrical Energy Transactions (TEL) PT PLN (Persero) UP3 Singkawang. Therefore, the immediate development of this system is highly urgent to eliminate the existing bottlenecks in manual reporting and costly third-party dependencies that currently impede timely strategic decision-making. Ultimately, this research contributes to the field of applied informatics and digital transformation by providing a practical, scalable, and cost-effective framework for automating corporate communication and data management workflows in the utility sector.

2. LITERATURE REVIEW

2.1 WhatsApp Blast

WhatsApp Blast is an application for sending bulk WhatsApp messages without needing to save the numbers in the sender's contacts beforehand [2]. WhatsApp Blast can be used to optimize communication processes such as promotions, service notifications, delivery of operational information. In addition, WhatsApp Blast supports an automatic, scheduled and monitored message sending mechanism through API integration, so that it can increase work efficiency and minimize errors that occur in the manual sending process.

2.2 Scheduler

A scheduler is an important component in an application system to organize and execute tasks automatically at certain times or intervals without requiring user intervention. Scheduling is the process of planning activities for specific times to complete tasks effectively [3] .In this research, the author uses a scheduling module, namely APScheduler, as a component to manage scheduling in the application being built, so that the process of sending messages can be carried out automatically according to a predetermined time. APScheduler supports various types of scheduling, such as date-based scheduling, interval-based scheduling, and cron-based scheduling, making it flexible to use for various application needs.

2.3 Application Programing Interface (API)

A An API is a set of commands, functions, and protocols that programmers can use when building software for a specific operating system [4]. API functions as an intermediary that connects two systems so that the data exchange process can run safely, standardized and efficiently without needing to know

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

the internal implementation details of each system. Through APIs, an application can take advantage of functions that other platforms have without having to build these features from scratch. In the context of digital communication, APIs are also used to support automatic messaging services, including the WhatsApp Blast system. One API service that is widely used for these needs is (Wablas.com), which provides a programming interface for sending automatic messages, mass messages, managing contacts, and scheduling messages in a structured and real-time manner.

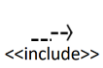
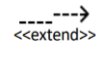
2.4 Flask

Flask is a Python-based micro-framework that does not have many tools and libraries [5]. Flask functions as an application framework and appearance of a website. By using Flask and the Python language, developers can create a structured website and can manage the way a website works more easily. The aim of using this framework is to speed up application development because flask provides libraries and a collection of program code that are ready to be used to build a webapp without having to create it from scratch. Microframework here means that Flask intends to make the core of this application as simple as possible but can still be easily added to. A microframework is a "small" version of a full-stack framework [6].

2.5 Use Case Diagram

A use case is a description of the function of a system from the perspective or point of view of system users. A use case diagram is a diagram that describes the benefits of a system from the perspective of someone outside the system—an actor [7]. With a use case diagram, the system's functional requirements can be described more clearly, structured, and easily understood by developers and users, thereby facilitating the system design and development process. Use case diagrams work by using scenarios which are descriptions of sequences or steps that explain what is done by the system and vice versa. A use case describes an interaction between one or more actors and the information system to be developed [8]. Use case diagrams have several symbols and functions, including those that can be seen in table 1 below:

Table 1. Use Case Diagram symbols

Image	Name	Information
	**Actor**	Entities that interact with the system or process the system itself
	**Use Case**	Use cases explain the actions/actions carried out by actors.
	**Include**	Include is a relationship that shows that a use case always includes or always calls other use cases as part of the process.
	**Extend**	Extend is a relationship that shows that a use case can add additional behavior to another use case, but only occurs under certain conditions.
	Association	The direct relationship between the actor and the use case shows that the actor interacts or uses the function.
	Generalization	Generalization is a relationship that shows inheritance between actors or between use cases

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

2.6 Activity Diagram

Activity diagrams describe the flow of functionalism in an information system. In full, the activity diagram defines where the workflow starts, where it stops, what activities occur during the workflow, and the sequence of activities that occur. Activity diagrams are a UML behavioral model that is suitable for system testing because they can depict the overall flow of a system [9]. Activity Diagram symbols can be seen in table 2 below:

Table 2. Activity Diagram symbols

Image	Name	Information
	**Start Point**	The starting point when an activity begins
	**Activity**	Activity describes the activities, processes, or actions carried out in the workflow. Usually rectangular with obtuse corners.
	**Decision**	Logic branching based on certain conditions.
	Join	A merger association where more than one activity is combined into one
	**End Point**	The endpoint of the activity process

3. METODOLOGI

The system development for the Web-Based WhatsApp Blast Application at PT PLN (Persero) UP3 Singkawang follows a structured and sequential Waterfall methodology. This approach ensures that each phase is thoroughly completed before proceeding to the next, maintaining a systematic progression from initial problem identification to final system deployment and maintenance. The waterfall model employs a systematic and sequential approach [10]. The implementation of each stage within this specific research context is elaborated as follows:

3.1 Requirements Analysis

In this initial stage, an in-depth investigation was conducted regarding the existing operational constraints at PT PLN (Persero) UP3 Singkawang. The evaluation revealed that the unit relied heavily on costly third-party messaging services with limited features and lacked centralized customer data storage, requiring officers to manually input numbers via Microsoft Excel. Furthermore, operational target reports—such as Independent Meter Reading (BMM), Swacam, and Meter Reading Route (RBM)—required tedious manual compilation before being shared to internal WhatsApp groups. Based on these findings, functional and non-functional requirements were established, including integrating the Wablas API, implementing the Python Flask framework, utilizing a MySQL database, and setting up automated scheduling via APScheduler.

3.2 Design

Following the requirements definition, the system architecture and user workflows were modeled to provide a clear blueprint for implementation. This stage involved designing the database schema in MySQL to centrally store active customer records, employee data, and message templates. Additionally, Unified Modeling Language (UML) diagrams were constructed, encompassing Use Case diagrams to define user interactions (Admin and Customers) and Activity Diagrams to map out logical procedures

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

such as the login process, customer data management, template administration, scheduled message blasts, and automated report dissemination.

3.3 Development (Implementation)

The core coding and system construction phase utilized the Python programming language combined with the Flask framework due to its lightweight and flexible microframework architecture. The backend logic was integrated with the Wablas API to handle HTTP POST requests for automated and mass messaging transmissions. Furthermore, an automated scheduler module (APScheduler) was coded into the system to handle time-sensitive broadcasts, such as daily report dispatches to internal management WhatsApp groups, while the frontend interface was developed using responsive web design principles.

3.4 Testing

To verify that all developed features met the functional expectations and resolved the initial operational problems, comprehensive testing was executed. Black-box testing was applied to examine the system's interface components and functional operations, ensuring that CRUD operations for customer databases, message template configurations, image attachments, and automated execution schedules performed reliably without errors. Quantitative performance evaluations were also recorded, measuring message delivery success rates and processing time improvements.

3.5.Maintenance

The final phase focuses on the ongoing support and operational sustainability of the deployed application within the Electrical Energy Transactions (TEL) field of PT PLN (Persero) UP3 Singkawang. This stage ensures the system remains stable, handles potential API updates from third-party services, and accommodates future feature expansions—such as advanced Artificial Intelligence chatbot integrations or two-way live chat features. The primary goal of maintenance is to provide cost-effective support for a software system throughout its entire lifecycle [11].

4. RESULTS AND DISCUSSION

4.1 System Design

This system design aims to model the application architecture and workflow based on the functionality that has been previously analyzed. This design model is used as a reference in implementing the database and system interface. The detailed design of the system being built is explained as follows.

4.1.1 Use Case Diagrams

This use case diagram describes the interaction between actors - such as admin or staff - related to the system, which includes main functions such as customer contact management, message draft management, and the process of sending information in bulk (blast) which is more efficient than conventional methods. This aims to ensure that digital communication flows are more structured, fast and well documented. The following is a picture of the use case diagram design that I made:

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

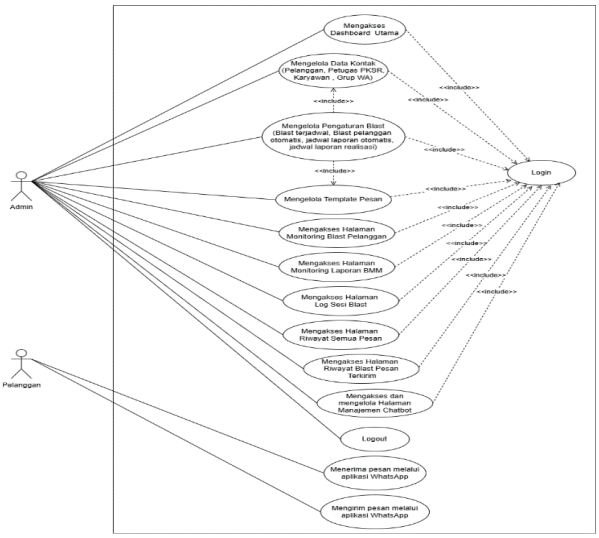


Figure 1. Use Case Diagram Design

4.1.2 Activity Diagrams

4.1.2.1 Activity Diagram Scheduled Message Blast

The Activity Diagram below illustrates the workflow of the scheduled message blast process. The activity diagram display of scheduled message blasts can be seen in Figure 7.

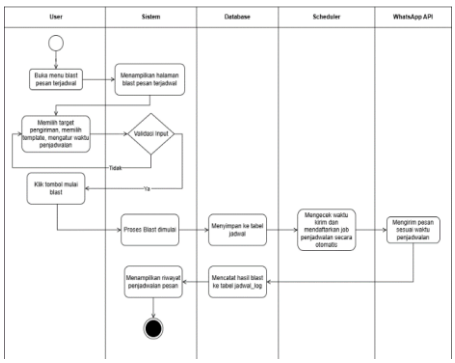


Figure 2. Activity Diagram Scheduled Message Blast

4.1.2.2 Activity Diagram Schedule Realization Report

The Activity Diagram below illustrates the work flow of the realization report schedule process (BMM, SwaCam, RBM). The activity diagram display of the realization report schedule can be seen in Figure 8.

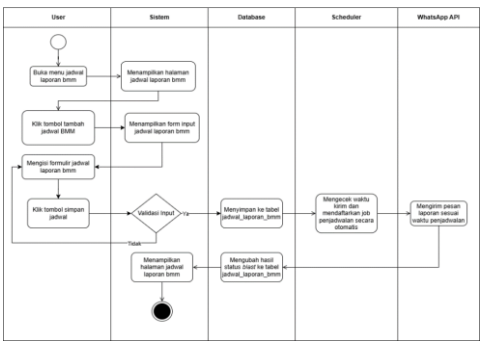


Figure 3. Activity Diagram Schedule Realization Report

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

4.2 System Implementation

4.2.1 Scheduled Message Blast Page

This page functions to manage the automatic message blast schedule, where in this section we set the delivery time, delivery target, message content, and set the delivery time scheme in the form of a delivery delay per message. The appearance of this page can be seen in Figure 13.

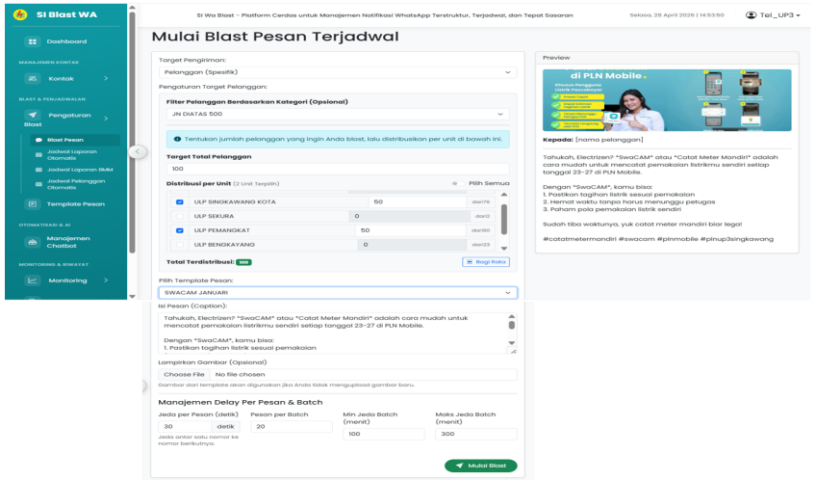


Figure 4.Scheduled Message Blast Page

4.2.2 Scheduled Message Blast Page

This page functions to manage the schedule for automatically sending monitoring reports to WhatsApp groups. This feature is designed to distribute four main types of reports, which include swacam monitoring, meter reading monitoring, monitoring that has been swacam (100% RBM), as well as recapping RBM 90-99% or those that have not done swacam. The appearance of this page can be seen in Figure 14.

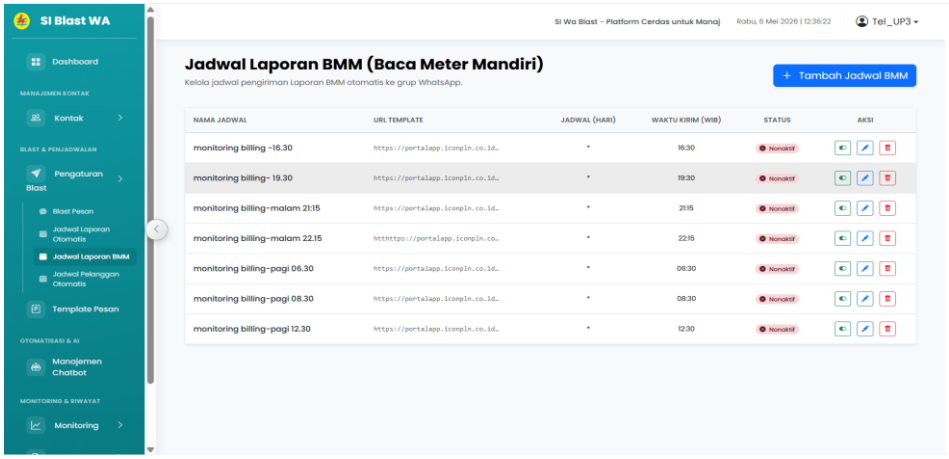


Figure 5.Scheduled Message Blast Page

4.3 System Testing and Performance Analysis

To evaluate the effectiveness of the proposed Web-Based WhatsApp Blast application in addressing previous operational constraints at PT PLN (Persero) UP3 Singkawang, system testing was conducted using black-box testing and performance metric analysis.

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

4.3.1 Functional Testing

Functional testing employs the black-box method to ensure that all modules and key features function correctly according to system design specifications, free from errors. The results of this functional testing are summarized in Table 3 below:

Table 3. Black-Box Testing

Feature	Test Scenario	Expected Result	Status
Authentication (Login and Logout)	Testing credential validation (empty, incorrect, correct), the Show/Hide Password feature, "Remember Me," and session termination (logout).	The system rejects invalid access, saves the authorization session if selected, and securely terminates the session upon logout.	Succes
Data Management	Add, modify, search, perform bulk deletion, and import data files (.xlsx) for Customer, PKSR, Employee, and WhatsApp Group entities.	Data is accurately stored/updated in the database, search filters are responsive, and the system rejects incorrect import file formats.	Succes
Template & Chatbot Management	Drafting messages, adding image parameters, testing hover previews, and configuring chatbot auto-reply rules.	The system displays image previews, rejects empty columns, and responds automatically based on keyword rules.	Succes
Blast Scheduling & BMM Reports	Configure transmission delays, save daily BMM/RBM report schedules, and execute the deletion of Blast Session Logs.	APScheduler accurately reads the schedule and processes automated blast deliveries at the designated times.	Succes
Monitoring & History	Filter the delivery period, download the PDF summary export, and display the sent message history preview modal.	The KPI dashboard accurately calculates total metrics, PDF reports are successfully generated, and delivery log details are displayed in full.	Succes

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

Based on the results of the design, implementation and testing that have been carried out, the Web-based WhatsApp Blast application using the Flask (Python) framework and the Wablas API has been successfully built and is running as designed. This system is able to increase work efficiency by automating the dissemination of customer service information as well as sending operational realization report schedules (BMM, Swacam, and RBM) to internal groups automatically. Centralizing contact data using MySQL has also succeeded in replacing the manual number input process via Microsoft Excel, so officers no longer need to enter data repeatedly. Apart from that, functional testing using the Black Box Testing method proves that all the main features—such as contact management, message templates, automatic scheduling, monitoring, and simple chatbots—function well, so that the implementation of this independent system is effective in reducing dependence on third parties in disseminating communication operational information, especially in the field of Electrical Energy Transactions at PT PLN (Persero) UP3 Singkawang.

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)

5.2 Recommendations

For the development and improvement of this application system so that it can provide wider benefits in the future, the author suggests adding the following:

1. Artificial Intelligence (AI) integration: Chatbot features that are currently still based on static keywords should be integrated with OpenAI API technology so that they are able to respond to customer questions more intelligently, interactively, and understand the context of the conversation naturally.
2. Improved Data Security: The security aspect of the system needs to be further improved through implementing database encryption to protect sensitive information as well as implementing a two-step verification method (Two-Factor Authentication) when logging in to the admin, to ensure the system is not easily accessed by irresponsible parties.
3. Expansion of Monitoring Features: Development of the monitoring feature which is currently only available for the monthly customer category, in the future it will need to be expanded to monitor the performance of sending messages to employees, PKSR officers and WhatsApp groups to help monitor the overall effectiveness of internal information distribution.
4. Full Two-Way Sending Feature: The addition of the Live Chat feature is highly recommended to support full two-way communication, so that administrators can view and reply to incoming messages from customers directly via the application's web interface.

BIBLIOGRAPHY

[1] A. T. Salsabilla, M. Hendayun, and C. Hendriyani, "The Use of ACMT and AP2T Applications In The Process of Making Customer Accounts In PT . PLN (Persero) UP3 Bandung Penggunaan Aplikasi ACMT dan AP2T Dalam Proses Pembuatan Rekening Pelanggan di PT . PLN (Persero) UP3 Bandung," vol. VI, no. 1, pp. 1–8, 2022.

[2] R. Permadi, A. Ahmad, and A. Munir, "WhatsApp blast as digital multiliteracy 4 . 0 product to improve students ' English learning achievement," vol. 6, no. 2, pp. 127–138, 2024.

[3] M. Pelajaran, "Penerapan Algoritma Genetika Dalam Penjadwalan," vol. 10, no. 2, 2023.

[4] K. Rizki, A. Adil, I. Pendahuluan, M. Barang, S. Motor, and M. Penumpang, "Implementasi Google Maps Api Berbasis Android Untuk Lokasi Fasilitas Umum Di Kabupaten Sumbawa," vol. 17, no. 2, 2018.

[5] R. K. Ngantung and M. A. I. Pakereng, "Model Pengembangan Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python," vol. 5, 2021, doi: 10.30865/mib.v5i3.3054.

[6] I. Yatini, F. W. Nurwiyati, K. Anam, J. Raya, and J. No, "Performa Microframework Php Pada Rest Api," vol. 19, no. 2, pp. 12–20, 2021.

[7] R. Rohmanto and T. Setiawan, "Perbandingan Efektivitas Sistem Pembelajaran Luring dan Daring Menggunakan Metode Use case dan Sequence Diagram," vol. 5, no. 1, pp. 53–62, 2022.

[8] H. Malius *et al.*, "Sistem Informasi Sekolah Berbasis Web Pada Sekolah Dasar Negeri (SDN) 109 SERITI," vol. 1, no. 3, pp. 156–168, 2021.

[9] A. Gutama, A. Arwan, and L. Fanani, "Pengembangan Kakas Bantu Pembangkitan Kasus Uji pada Model-Based Testing Berdasarkan Activity Diagram," vol. 3, no. 9, pp. 8325–8334, 2019.

[10] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," pp. 1–5, 2020.

[11] E. R. Suteja and L. C. Munggaran, "Penerapan Pemeliharaan Systems Maintenance Life Cycle Bank Swasta Nasional Berdasarkan Ieee Dan Iso Jisicom (Journal of Information System , Informatics and Computing) p-ISSN : 2579-5201 (Print)," vol. 4, no. 1, pp. 46–54, 2020.

DOI : https://doi.org/10.21063/jtif.2026.V14.1.1-3
Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)