



Development of Operational Application System at PT. XYZ with Flask Overriding

Aryudha Hattu

Faculty of Information Technology, Universitas Kristen Satya Wacana, Salatiga City, Central Java Province, Indonesia.

Email: 672020232@student.uksw.edu

Yeremia Susetyo *

Faculty of Information Technology, Universitas Kristen Satya Wacana, Salatiga City, Central Java Province, Indonesia.

Corresponding Email: yeremia.alfa@uksw.edu

Received: February 24, 2024; Accepted: April 1, 2024; Published: April 30, 2024.

Abstract: Advances in information technology have become a critical factor in increasing company operational efficiency. PT.XYZ, a business entity in the retail sector, faces an urgent need to design an operational application system that can adapt to current business dynamics. This research discusses and applies the development of operational application systems at PT.XYZ uses the Flask Overriding concept in the Flask framework. Flask Overriding is an idea in object-based programming that allows changing or replacing existing behavior in a system. This study explores the potential for implementing Flask Overriding to increase PT functionality, adaptability, and flexibility. PT XYZ operational application systems. Implementing this concept involves an in-depth analysis of a company's specific business needs, application architecture design, and practical implementation using Flask as the main framework. This research produces an operating system application head office application that is useful for assisting PT XYZ in storing and sending non-physical products needed by PT XYZ branches and head office so that the business processes and dissemination of information required for PT XYZ and its branches become efficient, fast, and structured.

Keywords: Python; Flask Framework; Flask Overriding.

1. Introduction

Technological developments are running very fast in various areas of life. The development of information systems supports business processes, especially for large-scale companies. An example is data management in a company, where this process will significantly influence the company's performance. Company data must be appropriately managed and correctly maintained to maintain business continuity. As time goes by, companies need proper and efficient management. Therefore, an information system is required to support these needs [1] as one of the largest retail companies in Indonesia, PT. XYZ plays an essential role in this country's economic ecosystem. For retail companies, data management dramatically impacts the entire business process. On the other hand, companies have millions of business processes that occur every day, so fast and efficient software is needed [2]. Inefficient data management can cause problems for companies, such as delays in managing customer orders, inventory errors that can result in shortages or excess stock, and the inability to analyze customer purchasing trends promptly. Therefore, it is essential for PT. XYZ to implement a reliable and efficient data management system to maintain the smooth and secure operation of the company's business [3]. Due to these problems, PT needs an appropriate data management system. XYZ, the information system required, is focused on managing non-physical product inventory data. The non-physical products in question are vouchers, tickets, etc. Management of non-physical products is compassionate because the products are digital. In a business environment that is increasingly connected digitally, PT is essential. XYZ to have a system that can face challenges related to data security and accuracy in managing non-physical product inventory.

The right information system will allow companies to monitor non-physical product stock in real-time, manage ordering and delivery processes efficiently, and integrate data with other systems such as payment and accounting. Python Flask Overriding technology can be used to solve this problem. Flask Overriding can be used to build enterprise data management systems by utilizing the advantages of the lightweight and flexible Flask framework [4] by using Flask Overriding, PT. XYZ can design and implement web applications specifically customized to meet their non-physical product inventory data processing needs with the advantages of Flask Overriding, such as the ability to access and manage databases easily, as well as the ability to integrate various additional features and extensions, PT. XYZ can quickly develop a system that suits the company's specific needs [5][6][7].

Previous research Implementation of a Framework in Creating Monitoring and Store Visit IT Support Applications at PT. In this study, Flask was used to monitor and plan IT visits, which helped resolve existing problems at PT.XYZ. Meanwhile, Flask is used in this research to make it easier for PT. Research conducted by Bartolomeus Bima Santoso and Pratyaksa Ocsa Nugraha Saian (2023) with the title "Implementation of the Flask Framework in the Development of Reporting Modules for Helpdesk Information System Applications at PT. Helpdesk information system application at PT XYZ [8]. PT XYZ is one of the leading minimarket retail companies in Indonesia. Various problems often arise in business operations that require solutions [9]. Currently, the problem is addressed with applications created using Oracle Forms. However, this application is considered less effective and efficient in terms of costs and reporting processes.

Meanwhile, this research explains how using flask overriding can help create easy and efficient web applications so that they can solve frequently encountered problems, namely, efficiency in developing web applications. Research conducted in 2022 by Alpha Adarrani Wibowo Putri and Yeremia Alfa Susetyo with the title "Implementation of Flask for Checking Distribution Center and Store Stock in the Stock Monitoring Application at PT XYZ" highlighted the use of the Flask Framework, a Python library, in creating a web application called Monitoring Stock at PT XYZ. This research aims to optimize stock control at PT XYZ to prevent losses due to errors in stock calculations [10].

On the other hand, this research aims to increase the company's operational efficiency and productivity by creating fast and responsive applications. The three previous studies described have a significant relationship in using the Flask Framework, even though they are applied in different contexts but with similar goals, namely to increase the company's operational efficiency. In the study Implementation of a Framework in Creating Monitoring Applications and Store Visit IT Support at PT. Meanwhile, in a study Implementation of the Flask Framework in the Development of Reporting Modules for Helpdesk Information System Applications at PT. On the other hand, the study Implementation of Flask for Checking Distribution Center and Store Stock in the Stock Monitoring Application at PT XYZ applies Flask in creating a stock monitoring application to optimize stock control at PT XYZ [10]. The fundamental similarity of these three studies lies in using the Flask Framework as a foundation for creating efficient and responsive solutions in dealing with various company operational challenges. This research aims to continue the direction of previous research by adopting Flask in

non-physical inventory data management from head office to branches, with the hope of increasing overall operational efficiency and productivity at PT.

2. Research Method

Figure 1 Operational Application System Development at PT. The initial stage of this research involves planning project requirements, including identifying the timeline and available budget. Development performs an in-depth analysis of the application's functional and non-functional requirements as a basis for determining the project scope and compiling a detailed work plan.

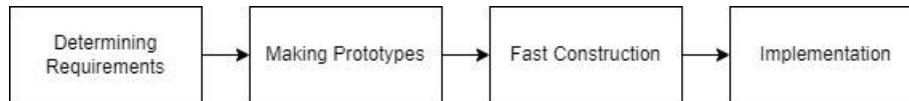


Figure 1. RAD Method

Once the project requirements are understood, the next step is system design, which involves creating a prototype using Flask Overriding. This prototype serves as an initial representation of the desired application, providing stakeholders at PT XYZ with a visual representation of the features and functionality that will be included in the operational application. The development process then begins with focusing on rapid construction and gathering feedback. The RAD method allows it to concentrate on fast iteration, build core application features efficiently, and implement changes based on user and stakeholder feedback. Ongoing communication ensures that the application meets their needs and expectations over time. The product implementation and finalization phase begins once all essential features are developed and approved. Perform final testing and debugging to ensure the application runs smoothly and meets the desired quality standards. Once the implementation is complete, the application is ready to be operated by PT XYZ, helping them increase efficiency and productivity in PT XYZ operations.

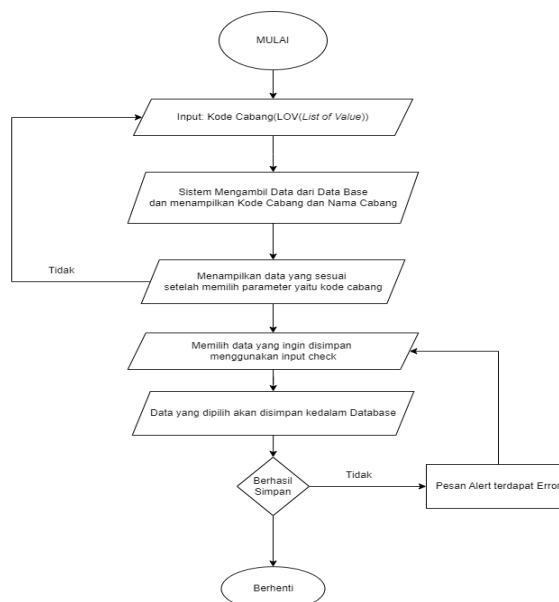


Figure 2. Flow Diagram

Figure 2 is a flow diagram used to explain a software process's flow, steps, or sequence from execution to completion. This process begins with the "Start" stage, where the user enters the branch code. After the branch code is entered, the system then acts to retrieve data from the database and displays the code and branch name that corresponds to the code that has been entered. The user can select other parameters if the displayed data does not match what is desired. These parameters can be other branch codes or specific criteria relevant to the data you want to search for. If the data displayed is as desired, the user will select the data they want to save. This data selection is done using input check, a method that allows users to select specific data from several data displayed. The selected data will then be saved into the database. If the saving process

is successful, the process will end, and the flowchart reaches the "Stop" stage. However, if the storage process fails, the system will display an error message explaining the reason for the failure [12].

3. Result and Discussion

3.1 Results

Figure 3 is a use case diagram that describes the data storage process using an overriding system in the Operation Application System Head Office PT application. XYZ. This process begins with detailed steps, where the user must first log in to access the application functionality. After successfully logging in, the user can select a menu from the various options available. One of the options relevant to the data storage process is "Select Branch Code (LOV)." Once the branch code is selected, the system will provide information about the "Store Code," which is an integral part of the data storage process. Next, users can select "Select Save" from the app's main menu. In this save option, there is an essential function, namely "Overriding," which gives users the ability to change or overwrite existing data [13] previously.

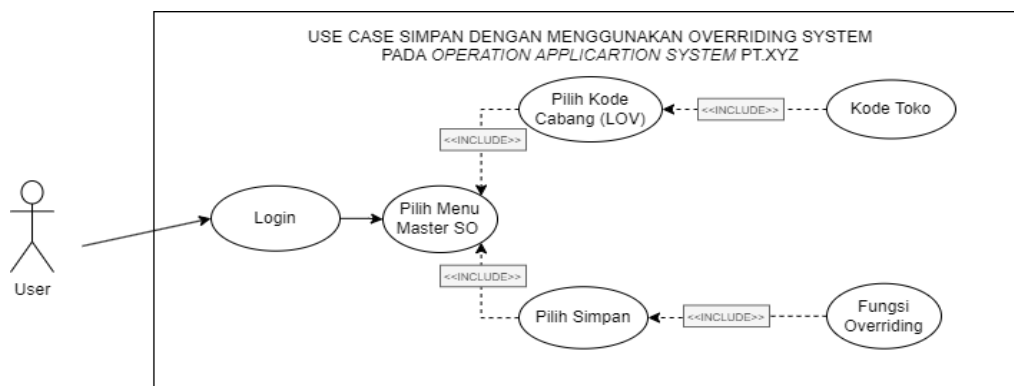


Figure 3. Use Case Diagrams

Figure 3 is a use case diagram that describes the data storage process using an overriding system in the Operation Application System Head Office PT application. XYZ. This process begins with detailed steps, where the user must first log in to access the application functionality. After successfully logging in, the user can select a menu from the various options available. One of the options relevant to the data storage process is "Select Branch Code (LOV)." Once the branch code is selected, the system will provide information about the "Store Code," which is an integral part of the data storage process. Next, users can choose "Select Save" from the app's main menu. In this save option, an essential function, "Overriding," allows users to change or overwrite existing data [13] previously.

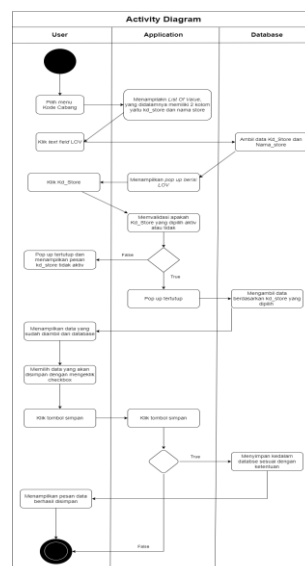


Figure 4. Activity Diagram

This activity diagram provides an in-depth illustration of the interaction between users, applications, and databases in the data storage process using the Operation Application System Head Office PT overriding system. XYZ. This process begins with detailed steps, where the user must first log in to access the application functionality. After successfully logging in, users can select a menu from the various options available. One such option is "Select Branch Code (LOV)," which is relevant to the data storage process. Once the branch code is selected, the system will provide information about the "Store Code," an essential part of the data storage process. Next, users can choose "Select Save" from the app's main menu. An important function, "Overriding," allows users to change or overwrite existing data previously in this option. Interactions between users, applications, and databases are well structured during this process. The user selects a menu and enters the required information. The system provides appropriate responses, such as displaying a list of reviews and pop-ups to confirm the user's actions. On the database side, activity diagrams reflect data storage and retrieval actions tailored to user interactions and processes within the application. Conditional statements check the active or inactive state of selected store codes, adding a layer of reliability to data management [14].

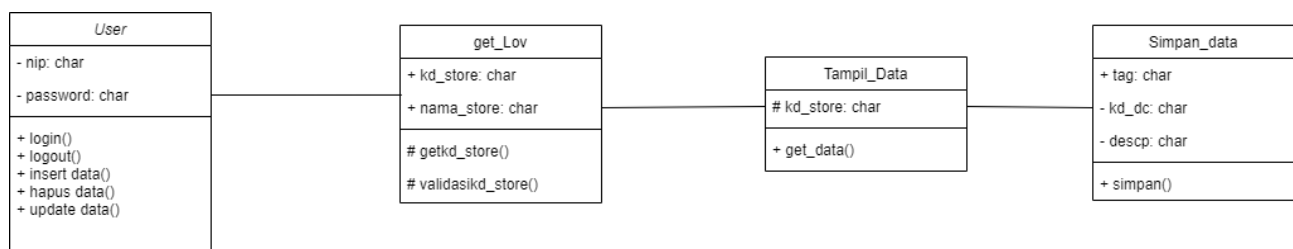


Figure 5. Class Diagram

In Figure 5, this Class Diagram depicts the interactions between users, applications, and databases in the Operation Application System Head Office PT application context. XYZ. The process begins with the user logging in to access the application functionality. After logging in, the user selects the "Store Code" menu, which triggers a method in the application object to display a list of ID values. The user then selects the ID value and clicks "OK," which triggers another technique in the application object to display detailed information. If the data is correct, the user saves it into the database by pressing the save button, triggering a method in the database object. However, if there are errors or incomplete data, the system will provide feedback as an error message, asking the user to correct and repeat the process. This diagram illustrates how methods in these objects are triggered by user actions, showing structured interactions between the user, application, and database [15]. Program Code 1 Import Framework Flask:

```
from flask import Flask, request
```

Program Code 1 is the program code used to import the flask framework. ``from flask import Flask`` This inserts the ``Flask`` class from the ``flask`` module. Flask is a lightweight and flexible web framework for Python. It provides the tools to build web applications quickly and efficiently [16]. The ``Flask`` class is the most critical part of Flask, which is used to create Flask application objects. ``from flask import request`` functions to import ``request`` objects from the ``flask`` module. The ``request`` object is one of the global objects provided by Flask that allows you to access data sent by users during HTTP requests. In the context of web applications, data submitted by users via forms or AJAX requests often needs to be accessed in building web applications using Flask, and this can be done easily using the ``request`` object. By importing both required modules, web applications can be built with Flask, and user data can be accessed via HTTP requests. For example, JSON data sent in a POST request can be obtained using the ``request.get_json()`` method. The ``Flask`` class is used to create a Flask application object and add routes to it, which determine how the application responds to received requests, including accessing and using data sent by the user. Program Code 2 Implementation of Flask Overriding:

```
def __init__(self):
    super().__init__()
    self.connect_branch = FUNGSI_CONNECT_DB_HO
    self.db = DatabaseServer

    def post(self):
        # Base implementation
```

```

class SimpanDataCon(MyBaseResource):
    def post(self):
        result = {}
        conn = FUNGSI_CONNECT_DB
        try:
            (...)
        except Exception as e:
            result['msg'] = str(e)
            result['status'] = False
        finally:
            conn.close()
        return result

```

Program Code 2 program codes are used to develop operational application systems at PT.XYZ using Flask and the overriding concept. There are two main classes: 'MyBaseResource' and 'SaveDataCon.' 'MyBaseResource' is a base class that functions as a base resource. In the '__init__' constructor method, initialization is performed by calling the parent class's constructor using 'super().__init__()', allowing the child class to access and utilize the attributes defined in the parent class. This class also has 'connect_branch' and 'db' attributes, intended to initialize connections to branches in the database and represent objects of the 'DatabaseServer' class. 'SaveDataCon' is a derived class that extends 'MyBaseResource' and redefines the 'post()' method. In the 'post()' method, logic attempts to save data to the database. The 'conn' attribute may refer to the connection created to interact with the database. Error handling uses the 'try-except-finally' block to handle situations when an error occurs in a data save operation. Error messages are captured and included in the results to be returned. When error handling is complete, the connection to the database is closed in a 'finally' block to ensure that used resources are correctly cleaned up and prevent resource leaks. Overall, this code snippet reflects software development practices that focus on sound resource management and error handling and using overriding concepts in object-oriented programming to modify the behavior of methods derived from parent classes.

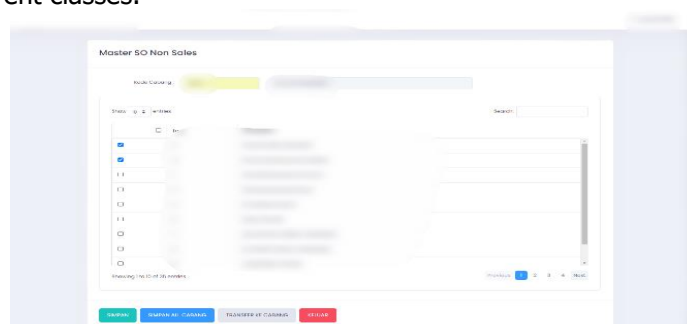


Figure 6. Home Page Display

Figure 6 shows the main page of the Controller So Non-Sales Operational Module application. Users can change tables containing data by selecting or deselecting the data list. If the user wants to save the selected data and only wants to save it in a particular branch code, the user can press the 'Save' button. If the user wants to save data to all branch codes, the user can select 'Save All Branches.' The last button is 'Branch Transfer,' which transfers data stored in the head office database into the branch database.

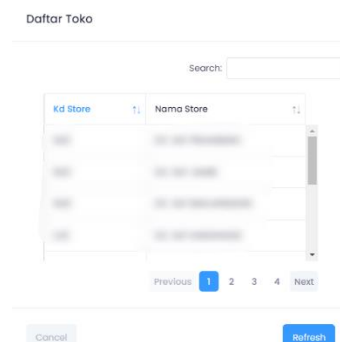


Figure 7. KD_STORE modal display

Figure 7 is a modal display for users to select a branch code. The table contains two columns containing the shop code and shop name that the user will select later for the data storage process in the database. Testing is carried out using the black box method to ensure the software runs well.

Table 1. Black Box Testing

Test Scenario	Expected Outcome	Status
User opens Master SO Non Sales menu	When opening the form, access rights check is performed	Able to open menu
User selects LOV KD_DC	Able to display LOV KD_DC	OK
User can check (T) and uncheck (F) data	Able to select data	OK
"Save" button, if there is duplicate data, will update the data	Able to insert and/or update data in the table	OK
"Save All Branches" button, if there is duplicate data, will update the data	Able to save to all branches	OK
"Transfer to Branch" button, if there is duplicate data, will update the data	Able to transfer data to branches	OK

Table 1 shows the results of the last stage of black box testing carried out on the software. Tests were carried out using the save function in the head office and branch databases. Based on the results of the tests carried out, all experiments have achieved a success rate of 100%, so it can be stated that the software is running well and in accordance with the initial requirements.

4. Related Work

Development of operational information systems at PT. XYZ uses Flask technology as the primary foundation and has become a research subject that has attracted the interest of researchers in recent years. Several related studies have been conducted covering various aspects of using Flask in the context of business applications. One relevant research is the work of Surya and Susetyo (2023), which discusses the development of email blast automation in a document-sharing application using the Gmail API at PT XYZ. This research shows how Flask is used to develop applications that support automatic email sending, increasing the efficiency of internal company communications [1]. This implementation shows that Flask can be used in various application contexts, not just limited to developing operational information systems. Likhari and Purwanto (2021) analyzed and designed a web-based online ticketing helpdesk information system at PT XYZ. Although this research is not directly related to the use of Flask, it shows the importance of developing web-based information systems to support company operations [2]. This study provides insight into the need for integrated and responsive information systems to handle customer requests and problems effectively. Putra and Susetyo (2020) implemented the Master Store API using Flask, REST, and ORM at PT XYZ. This research highlights the use of Flask in building APIs for managing store data, providing insight into the potential use of Flask in service-based application development [3]. This implementation shows Flask can facilitate interaction between various system components, allowing for smoother and more efficient integration. Darmawan (2023) developed a web-based application using Python Flask for data classification using the C4.5 decision tree method. Although the main focus of this research is not on business applications, it provides further insight into the use of Flask in the development of complex web-based applications [4]. This study shows that Flask can be used in various contexts of application, including data analysis and artificial intelligence. Wijayanto and Susetyo (2022) reported the implementation of the Flask Framework in developing helpdesk information system (SIH) applications. This research shows that Flask can be used in the context of customer service management, which is relevant to PT XYZ's efforts to improve the quality of service to its customers [5]. Implementing a helpdesk information system using Flask allows PT XYZ to be more responsive to customer requests and ensures that customer problems can be handled efficiently and effectively. Jonathan (2023) implemented the Flask framework in developing IT support monitoring and store visit applications at PT XYZ. Although the focus of this research is different from operational system development, it provides an overview of the variety of applications that can be developed using Flask. This implementation shows Flask can be used in various contexts of application, including monitoring and technical support.

The research differentiates itself from others in developing operational information systems at PT XYZ by applying the Flask Overriding concept. The main contribution of this research lies in implementing Flask Overriding, which allows system behaviour to be modified according to specific business needs. In developing

information systems, especially in a dynamic business environment such as PT XYZ, flexibility and adaptability are the keys to achieving optimal operational efficiency. Flask Overriding allows PT XYZ to adapt its operational information system precisely to its business needs. With the ability to change or replace the behaviour of existing systems, companies can overcome the unique challenges faced in managing non-physical data, such as digital vouchers and tickets. This allows PT XYZ to speed up their business processes, increase data accuracy, and integrate information with other systems more effectively. In addition, this research proposes a development approach based on Rapid Application Development (RAD), which allows PT XYZ to obtain operational applications quickly and iterate them efficiently based on user feedback. This approach enables companies to prioritize the most critical business needs and respond to changing needs more quickly.

5. Conclusion

In developing operational application systems at PT.XYZ, the Flask web framework, made a significant contribution. A critical element in this development is implementing the overriding concept using Flask. An organized class structure is shown through the MyBaseResource class as a superclass with several essential attributes, such as connect_branch and db. The derived class, SaveDataCon, utilizes the concept of inheritance, taking attributes and methods from the superclass. Overriding is applied to the post method in the SimpanDataCon class, allowing developers to provide custom implementations that suit application needs at PT.XYZ. This example includes special handling of database operations using Flask and data processing from HTTP requests. Overall, the using overriding concept was used in implementing Flask in developing operational application systems at PT. This is important in ensuring that applications can be developed and managed efficiently as business needs evolve. Based on the research that has been carried out, many features can still be developed further. As a suggestion, the software still does not have a complete error-checking function. So that in the future, a function can be created to record errors or what is usually called a system log to make it easier for users and developers who carry out software maintenance and repairs.

References

- [1] Surya, A., & Susetyo, Y. A. (2023). Pembangunan Automasi Email Blast pada Aplikasi Document Sharing Menggunakan Gmail API di PT XYZ. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 8(3), 1061–1073. <https://doi.org/10.29100/jupi.v8i3.4031>
- [2] Likhar, W., & Purwanto, H. (2021). Analisa Dan Perancangan Sistem Informasi Ticketing Helpdesk Online Berbasis Web: Studi Kasus Pt Xyz. *JSI (Jurnal sistem Informasi) Universitas Suryadarma*, 8(2), 103-116.
- [3] Putra, B. P., & Susetyo, Y. A. (2020). Implementasi Api Master Store Menggunakan Flask, Rest Dan Orm Di Pt Xyz. *SISTEMASI*, 9(3), 543-556. <https://doi.org/10.32520/stmsi.v9i3.899>.
- [4] Darmawan, A. C. (2023). Pengembangan Aplikasi Berbasis Web dengan Python Flask untuk Klasifikasi Data Menggunakan Metode Decision Tree C4. 5. <https://dspace.uui.ac.id/handle/123456789/42602>.
- [5] Wijayanto, C., & Susetyo, Y. A. (2022). Implementasi Flask Framework Pada Pembangunan Aplikasi Sistem Informasi Helpdesk (SIH). *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7(3), 858-868. <https://doi.org/10.29100/jupi.v7i3.3161>.
- [6] Jonathan, R. (2023). *Implementasi Framework Flask pada pembangunan Aplikasi Monitoring dan Store Visit IT Support pada PT. XYZ* (Doctoral dissertation). Thesis. <https://repository.uksw.edu/handle/123456789/30365>.
- [7] Suraya, S., & Sholeh, M. (2022). Designing and implementing a database for thesis data management by using the python Flask Framework. *International Journal of Engineering, Science and Information Technology*, 2(1), 9-14. <https://doi.org/10.52088/ijesty.v1i1.197>

-
- [8] Santoso, B. B., & Saian, P. O. N. (2023). Implementasi Flask Framework pada Development Modul Reporting Aplikasi Sistem Informasi Helpdesk di PT. XYZ. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 7(2), 217-226. <https://doi.org/10.35870/jti>
 - [9] Soliha, E. (2008). Analisis Industri Ritel di Indonesia. *Jurnal Bisnis dan Ekonomi (JBE)*, 15(2), 128–142.
 - [10] Wibowo Putri, A. A., & Susetyo, Y. A. (2022). Implementation of Flask for Stock Checking in Distribution Center & Store on Monitoring Stock Application in PT. XYZ. *Jurnal Teknik Informatika (Jutif)*, 3(5), 1265–1274. <https://doi.org/10.20884/1.jutif.2022.3.5.334>
 - [11] BSI, G. B. A. L. A. (2018). Metode Rapid Application Development (RAD) Pada Perancangan Website Inventory PT. Sarana Abadi Makmur Bersama (SAMB) Jakarta. *Evolusi: Jurnal Sains Dan Manajemen*, 6(2). <https://doi.org/10.31294/evolusi.v6i2.4414>.
 - [12] Walingkas, H. L., & Saian, P. O. N. (2023). Penerapan Framework Flask pada Pembangunan Sistem Informasi Pemasok Barang. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 7(2), 227-234. <https://doi.org/10.35870/jtik.v7i2.729>.
 - [13] Adha, B., Praditya, E., Nataliani, Y., & Tanaem, F. (2021). Perancangan Sistem Informasi Penyimpanan Data Transaksi Simpan Pinjam pada Koperasi Berbasis Web. *AITI: Jurnal Teknologi Informasi*, 18(Februari), 34–53.
 - [14] Najmi, A., & Nadjib, M. (2020). Analisis Dan Perancangan Sistem Informasi Simpan Pinjam Berbasis Website Pada Koperasi Karya Abadi. *Jurnal Satya Informatika*, 5(01), 9-17. <https://doi.org/10.59134/jsk.v5i01.40>.
 - [15] Putra, T. D. E., Khusnuliawati, H., & Hernanjaya, A. N. (2018). Penerapan Framework Codeigniter pada Pembangunan Sistem Informasi Akademik di Universitas Sahid Surakarta. *Jurnal Gaung Informatika*, 11(2).
 - [16] Evan, D. J., & Saian, P. O. N. (2023). Implementasi Python Framework Flask pada Modul Transfer Out Toko di PT XYZ. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 8(4), 1121-1131. <https://doi.org/10.29100/jipi.v8i4.4020>.