

---

## Dynamic Approval Matrix Application Design for Technical Department Document Change Request Application

Riska Zulfiqoh<sup>1</sup>, Muhamad Bahrul Ulum<sup>2</sup>

<sup>1,2</sup>Informatics Engineering Department, Faculty of Computer Science  
<sup>1,2</sup>Esa Unggul University

<sup>1,2</sup>Jl. Arjuna Utara No.9, Duri Kepa, Kebon Jeruk, West Jakarta, DKI Jakarta 11510  
E-mail: <sup>1</sup>zulfiqoh.riska@student.esaunggul.ac.id, <sup>2</sup>m.bahrul\_ulum@esaunggul.ac.id

---

### ABSTRACT

---

#### Article:

Accepted: August 16, 2022

Revised: July 13, 2022

Issued: November 15, 2022

© 2022 The Author(s).



This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

---

#### \*Correspondence Address:

zulfiqoh.riska@student.esaunggul.ac.id

This study discusses problems on the website of the NPPS company which will also be used by its branch company, namely NPPSLK. While the website currently only supports one company, it is necessary to develop the website to become multicompany. If the website is made multicompany, then the applications in it are automatically adjusted. One of the applications on the website is an application with systematic approval, namely the Dynamic Approval Matrix (DAM). DAM was developed to be able to support multicompany, support flow management features, status, and approval actions for each company.

So, a multi-company configuration is carried out on one system, and using the waterfall method with steps. Advance Dynamic Approval Matrix (ADAM) was created as multi-enterprise support DAM development. So, it can be used by multi-company. ADAM is made using waterfall method using Odoo with Python, XML, and PostgreSQL. The app contains forms to manage related apps, approval flow settings, multi-company, status settings, and Call-to-Action buttons to facilitate document approval.

The ADAM application has been tested on the TDCR application using black box indicator with the conclusion that website results are more effective and efficient because it can be used by multicompany, and administrators can freely adjust according to company needs.

**Keywords:** *Website, Multicompany, Approval System, Waterfall, Python, Odoo*

---

## I. INTRODUCTION

NPPS is a company that uses a website-based digital system. NPPS has an Engineering Department which is in charge of technical. The Engineering Department uses the Engineering Department Document Replacement Request (TDCR) application available on the NPPS website with systematic approval. The approval flow is governed by the Dynamic Approval Matrix (DAM) application.

Currently, NPPS has a branch company, namely NPPSLK which requires a website on one system so that it is easy to connect. However, the system's website only supports single companies. Furthermore, these two companies have their own employee structures. Thus, it requires different approval flows, states and actions. However, the running system does not support this feature.

Approval Management Application that runs automatically, can reduce errors that often occur related to the approval of a document [1]. This dynamic website is more effective, easier, and faster because it is on the same system [2]. Odoo makes it easy to solve problems in processing business transaction data and makes it easier for users, especially admins to manage business transaction processes [3].

To overcome the problems that exist in the current system, with reference to other previous research journals, a dynamic website system will be more effectively used. In this study, the website system not only runs automatically with dynamic approval management, but also multi-company configuration on one system, and a Dynamic Approval Matrix (ADAM) is created as a DAM development that supports multi-company. So, it can be used by multi-company.

### 1.1 Theory Review

#### 1.1.1 Website Application

Web-based applications are applications that can be accessed via a web browser when connected to the internet or intranet [4]. Based on the above definition, the author concludes that the website application is a software program commonly referred to as software that functions to execute commands as required by the user. Users interact with this website application interface by using a page display that displays various kinds of information or data in the form of text, or images that can be dynamic or static. Then, this website application

can be accessed by the user connecting to the internet network using a browser.

#### 1.1.2 Approval Matrix

The Approval Matrix known as the "schedule of authority" is a visual representation of this rule. It clearly designates who is responsible for approving what and when [5]. Based on the statement above, the Dynamic Approval Matrix is a rule which will appoint who will be responsible for agreeing on a matter within a certain time frame. This Approval Matrix is efficient and can prevent errors in the dynamic approval flow.

#### 1.1.3 SDLC Model Waterfall

SDLC (Systems Development Life Cycle), in systems engineering and software engineering, is the process of creating and modifying systems and the models and methodologies used to develop these systems [6]. The Waterfall model is "a model that provides a sequential or sequential software life flow approach starting from analysis, design, coding, and testing" [7]. Based on this definition, it can be concluded that SDLC is a description of the system development life cycle that is used to develop a system starting from the manufacturing process to conversion and the waterfall method is one of the system development methods.

#### 1.1.4 Odoo

Odoo is an open-source web-based ERP application that can be used as an Information System application for processing business transactions [3]. From the statement above, it can be concluded that Odoo is an open-source software or business application (ERP). Odoo has very complete features ranging from CRM, Sales, Purchase, Accounting, Inventory, Project Management, Events, Payroll, Websites, e-Commerce, and others that function for transactions or business needs.

#### 1.1.5 XML (Extensible Markup Language)

XML is technically defined as a meta-markup language that provides a certain format for documents that have structured data [8]. Based on this information, XML is used to describe the document as a meta-markup language with a specific format with structured data. In this study, the author uses XML as a language to set the document format.

### 1.1.6 Python

Python is a dynamic interpretive, object-oriented and semantic programming language. Python has high-level data structures, dynamic typing and dynamic binding [9]. Based on the above understanding, Python is a high-level programming language whose syntax is easy to understand by users and its data structure is high-level. So, users will find it easier to understand the Python programming language.

### 1.1.7 UML (Unified Modeling Language)

UML (Unified Modeling Language) is a visual modeling method for object-oriented system design tools, or definition of UML is a language that has become a standard in the visualization, design and documentation of software systems [1].

UML Class Diagram is a class that describes the structure and explanation of classes, packages, and objects and their relationships with each other such as containment, inheritance, association and others [10]. In this study, UML Class Diagrams are used to describe the relationship between classes with the attributes in each class that is related to each other. Its function is to make it easier to find out the interconnected relations and cardinality used in the Dynamic Approval Matrix Application.

UML Use case diagrams describe the relationship between actors and the activities they can perform on the application [10]. Based on this information, it can be concluded that the Use Case Diagram is a diagram that shows the activities that can be carried out by users and what functions can be performed by users who act as actors.

UML Activity Diagram is a diagram that describes the workflow or activity of a system or business process that exists in the software [11]. Based on this definition, it can be concluded that the Activity Diagram is a diagram that describes the workflow of a system that is being designed. This diagram shows the business processes of the system.

## II. METHODOLOGY

### 2.1 Data Collection Method

Descriptive research data collection with a qualitative approach was carried out with interviews conducted via zoom meeting by asking several questions about the current system with all its shortcomings to find out

existing problems, direct observation of the object being studied, namely the system running on the DAM application and TDCR application, literature study as a data collection technique to find data through books, internet, journals, and books related to research topics to assist in building systems and writing final project reports with journal references.

### 2.2 Problem Analysis Method

The data obtained through data collection, this problem is analyzed using the PIECES method with a classification framework of Performance, Information, Economy, Control, Efficient, and Service.

### 2.3 System development method

Website-based Dynamic Approval Matrix Application design will be developed using the Waterfall system development method, namely through the Requirement Analysis, Design, Implementation, and Testing process.

The programming language chosen in designing this website-based Dynamic Approval Matrix Application is to use XML, Python as the programming language used and PostgreSQL as the database.

## III. RESULTS AND DISCUSSION

### 3.1 Odoo Multicompany Configuration

Before creating the ADAM application, need to configure the multi-company website system by adding a new company to the website system. In Odoo, this configuration can be done by logging in with email and password, opening the settings menu, and on the company tab, pressing the manage company button.

Then creating a new company. After the company has been created, go back to settings and then press the save button. So, the new company was successfully created and the website system has a switch feature to another company.

### 3.2 Current System

The system that is running at the company is described using the UML Use Case, UML Activity, and UML Class. The following diagram is the UML diagram of the current system:

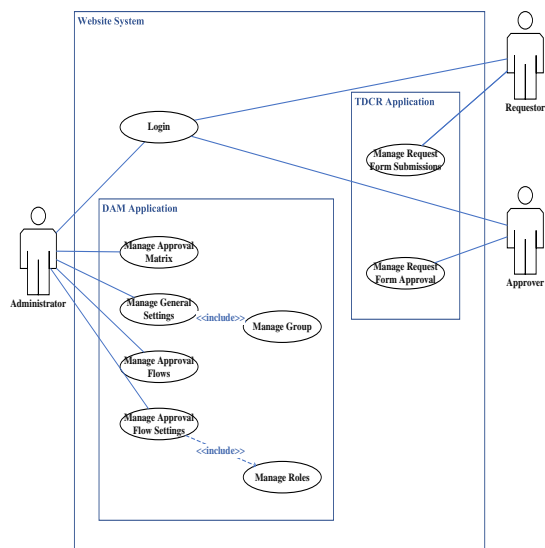


Figure 1. Use case diagram of the current system

Figure 1 Administrators, requestors, and approvers can log in. In the Dynamic Approval Matrix (DAM) Application, administrators can manage the approval matrix, manage general settings, manage approval flow, and manage approval flow settings.

Manage general settings including managing groups in them. Manage approval flow settings including management roles in it. In the TDCR application, the requestor can manage the request form. Meanwhile, the approver can manage the approval of the request form.

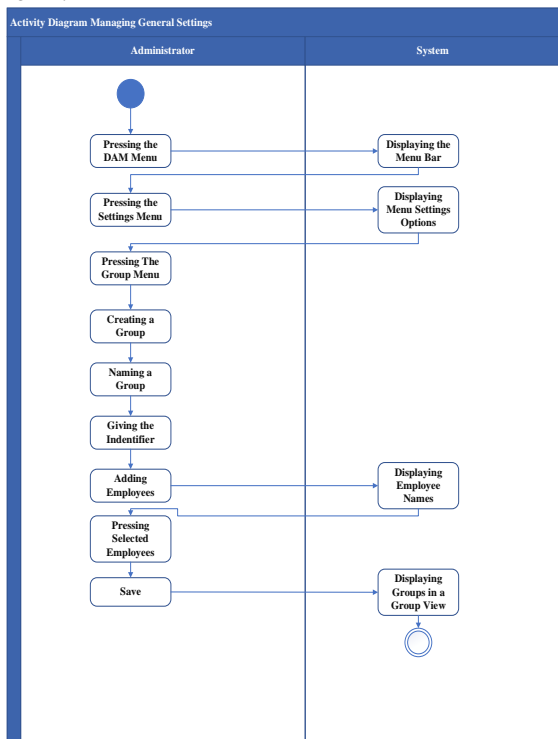


Figure 2. Activity diagram managing general settings

Figure 2 is an Activity Diagram that explained that after login, the administrator can press the DAM menu. Next, the system will display a menu bar. Then the administrator can press the settings menu and the system will display the settings menu options.

The settings menu has a submenu, namely groups. Administrators can hit the group menu and create a new group. Administrators can name groups, and give identifiers.

Furthermore, administrators can add employees to the group. Then, the system will display the name of the employee. Administrators can select employees by pressing the name of the selected employee. Then the administrator can save by pressing the save button and then the system will display the group in the list view.

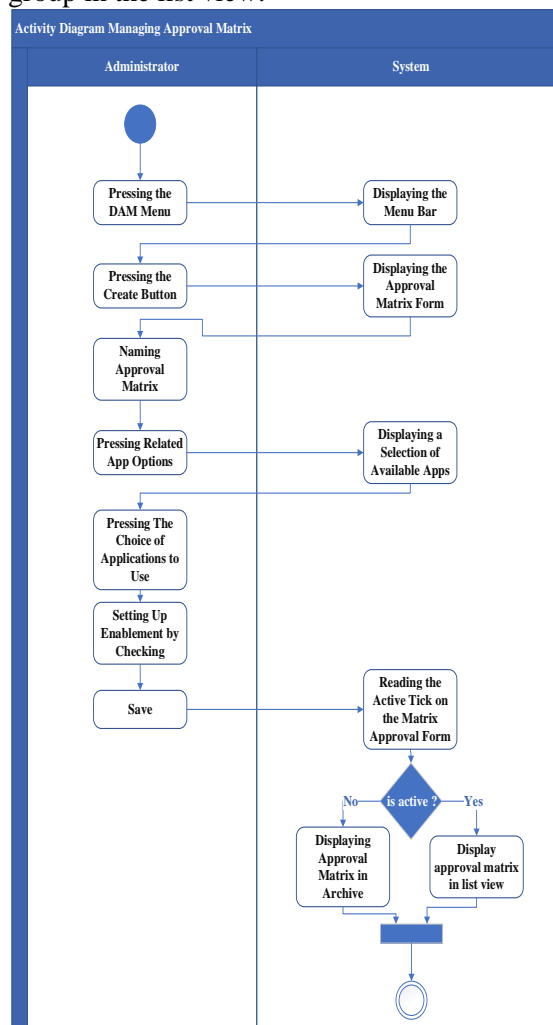


Figure 3. Activity diagram managing approval matrix

Figure 3 is an Activity Diagram that explained that the administrator can press the DAM menu. Then the system will display a

menu bar and the administrator can press the available create button. Then, the system will display the approval matrix form.

Administrators can name the approval matrix and then press to select the application to be associated with. Next, the system will display the available application options and then the administrator can press the application option to be used and set the activation by placing a check mark.

After filling is complete, the administrator can press the save button and the system will read about the activation on the approval matrix form. If it is active, the system will display the approval matrix in the list of approval matrix forms.

If the tick on the approval matrix form is inactive, the system will display the approval matrix form on the archive list.

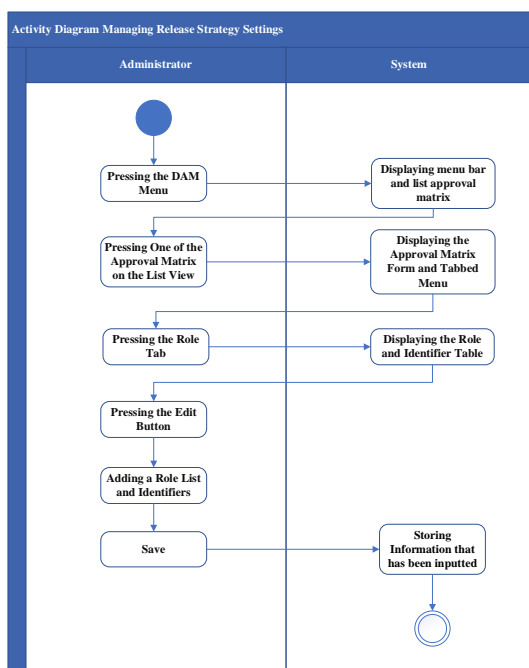


Figure 4. Activity diagram managing release strategy settings

Figure 4 is an Activity Diagram that explained that the administrator can press the DAM menu and the system will display a menu bar and an approval matrix list. Next, the administrator can press one of the approval matrices on the list display and the system will display the approval matrix form. The administrator can then press the roles tab and the system will display the roles table. Then the administrator can press the available edit button and add a list of roles and identifiers. After filling is complete, the administrator can press the save button.

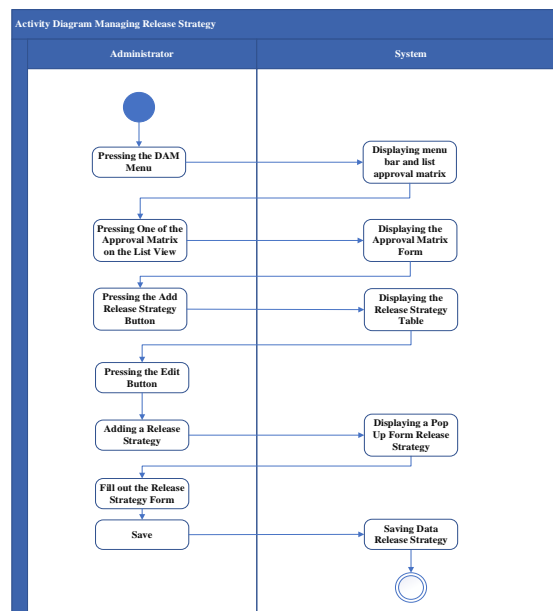


Figure 5. Activity diagram managing release strategy

Figure 5 is an image that explained Administrators can enter the application by pressing the DAM menu. The system will display a menu bar and a list of approval matrix. Then, the administrator can create a new version for release strategy. To edit the release strategy table, the administrator can press the edit button, then add the release strategy. Administrators can fill out the release strategy form and saving inputted data.

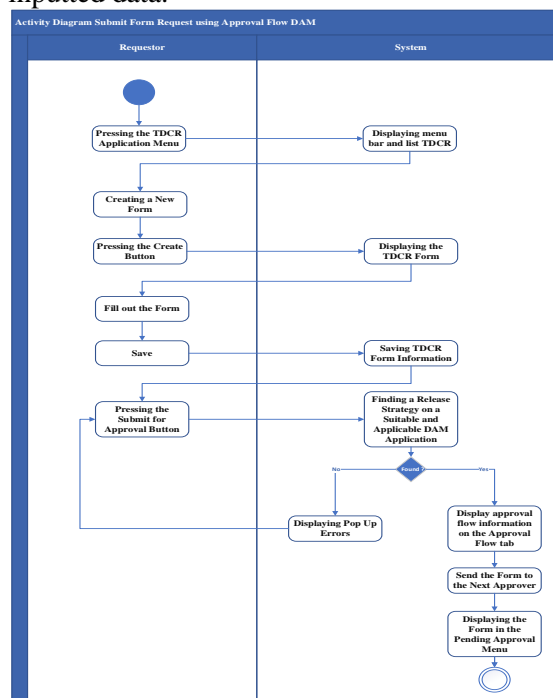


Figure 6. Activity diagram submit form request using approval flow DAM



Figure 6 is an Activity Diagram image of submitting a request form using the DAM approval flow. In this diagram, it is explained that the requestor creates a new form by pressing the available create button. The system will display the TDCR form. Then, the requestor can fill out the form.

Then the requestor can press the submit for approval button to send the request form. Then, the system will find the appropriate and applicable release strategy for the DAM application.

If the release strategy is found, the system will display the approval flow information on the approval flow tab. Next, the form will be sent to the approver in accordance with the applicable release strategy.

If the release strategy for the DAM application is not found, it will display an error pop-up. Then the requestor must press the submit for approval button again until the release strategy is found.

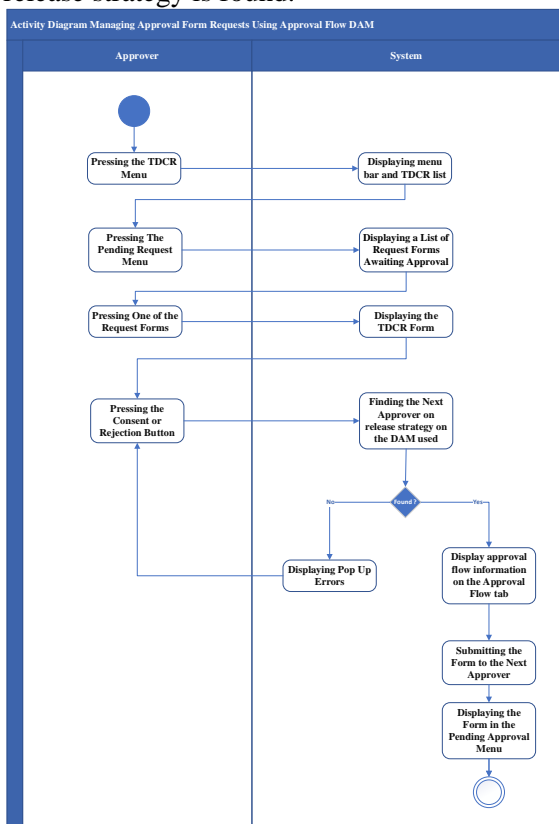


Figure 7. Activity diagram managing approval form requests using approval flow DAM

Figure 7 explains that the approver can find the form that needs approval on the pending approval tab. Approvers can approve the form by pressing the approve or decline buttons provided.

Then, the system will find the next approver in the release strategy of the DAM application used.

If the release strategy is found, the system will display the approval flow information on the approval flow tab. Next, the form will be sent to the next approver according to the release strategy used.

If the release strategy for the DAM application is not found, it will display a pop-up error. Then the approver must press approve and decline again until the release strategy is found.

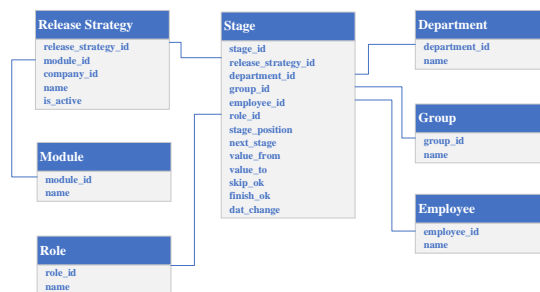


Figure 8. Class diagram current system

Figure 8 is a Class Diagram of a running system that shows the relationship between classes in which it contains fields. There are six classes used in the current running system.

The first class is the employee class which stores employee data. Furthermore, the second class is the group class which will store the groups used in the approval flow. The third class is the department class which stores the available departments. The fourth class is the role class which will store the identifier data for the employees involved in the approval flow.

Next is the module class which stores application data available on the system and the last class is the release strategy class which stores the release strategy data that has been created by the administrator.

### 3.3 Requirement Analysis

**Functional Requirements** This system consists of the main information that is interconnected and supports each other. Functional Requirements in this study are login, Settings, and Release Strategy funds.

**Non-Functional System Requirements** for this system are supporting requirements in operating a website-based Dynamic Approval Matrix Application. Requirements in this study are Laptop as hardware and some software, namely Microsoft Windows 10 Pro 64-bit, PyCharm Community Edition 2020 3.3, Python 3.7 dan PgAdmin 4.

### 3.4 Design

Based on the System Requirements Analysis, the author then designed the system with functional model of the system using UML diagrams.

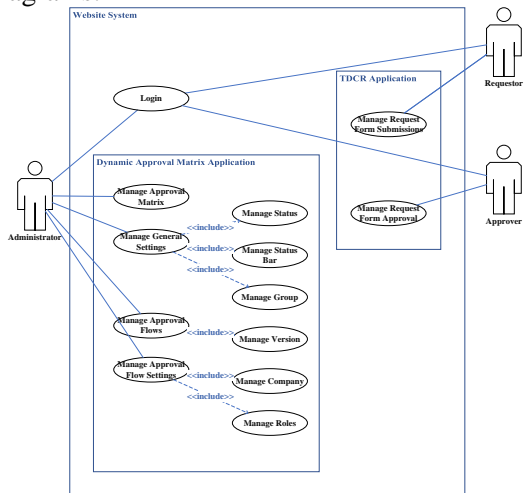


Figure 9. Use case diagram of current system

Figure 9 This image explains that there are 4 additional use cases in manage general settings including manage status and status bar. In the manage approval flow, there is an additional use case that includes manage version. In the manage approval flow settings there is an addition including manage company.

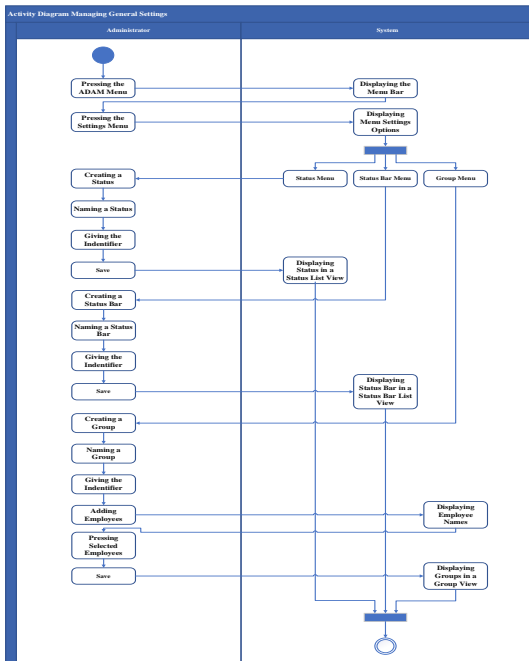


Figure 10. Activity diagram managing general settings

Figure 10 explains that there are two new activities, namely the activity on the status tab and the status bar tab.

For the Activity Diagram Managing Approval Matrix is the same activity as the old at Figure 3, the difference is that in the new system, the administrator uses the ADAM application.

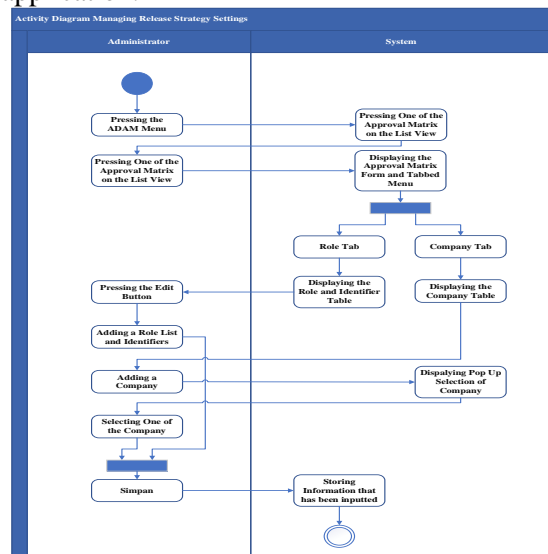


Figure 11. Activity diagram managing release strategy settings

Figure 11 explains that there is one new activity, namely the activity on the company tab.

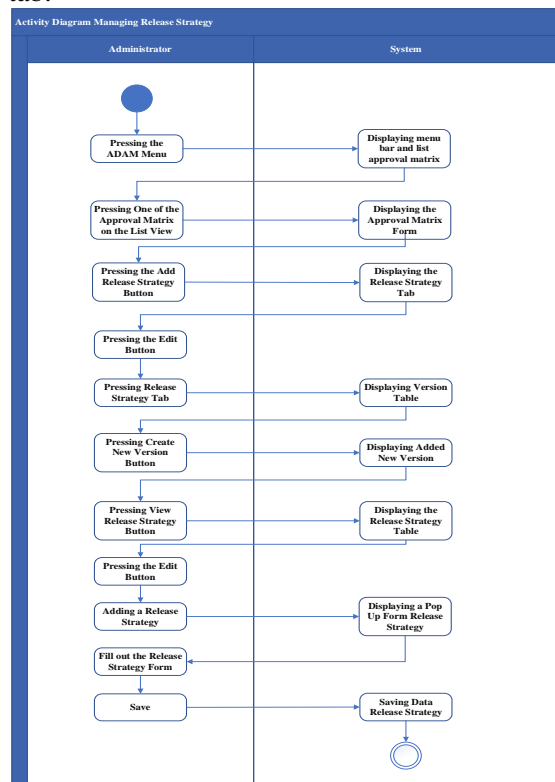


Figure 12. Activity diagram managing release strategy

Figure 12 explained that there are two new activities, namely adding the tab release strategy and setting the version.

For activity diagram submit form requests using approval flow ADAM and managing approval form requests using approval flow ADAM is the same activity as the old one in Figure 7, the difference is in the new system, the system will look for a release strategy on the ADAM application.

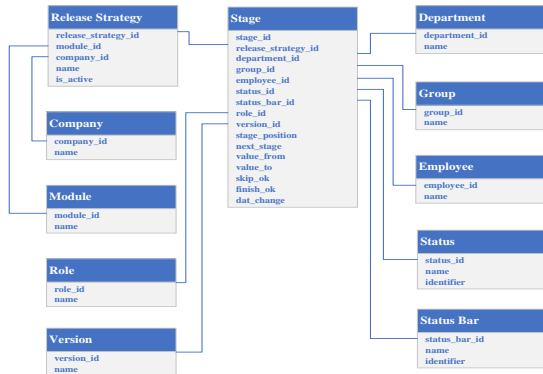


Figure 13. Class diagram current system

Figure 13 This image explains that there are 4 additional classes in the new system. Company class to store company information so that it can be multi-company. Version class to store the version of the release strategy. Status class and status bar to save status and status bar is used as needed.

### 3.5 Implementation

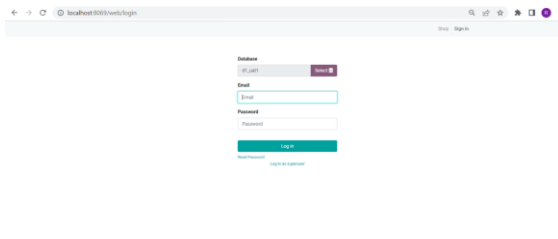


Figure 14. Login page

Figure 14 For login by entering your email and password.



Figure 15. Multi-company switch

Figure 15 For switch companies just by switching companies.

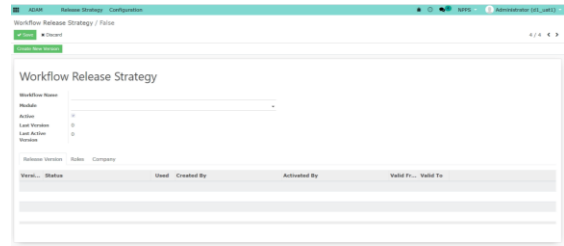


Figure 16. Form ADAM application page

Figure 16 ADAM form with release strategy, roles and company tabs.

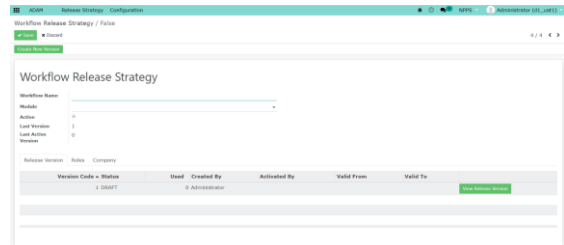


Figure 17. Version page

Figure 17 is ADAM version page with the button to create the new version.

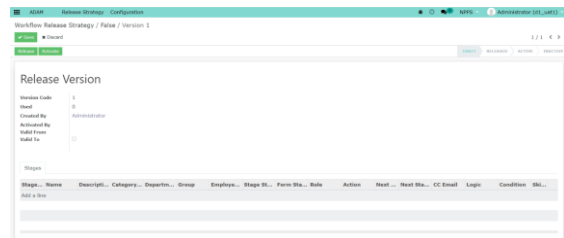


Figure 18. Inside release version page

Figure 18 is inside ADAM release version with the release strategy column.

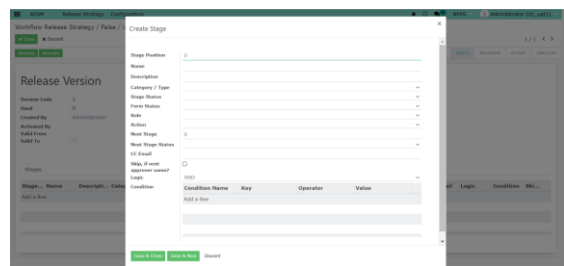


Figure 19. Release strategy form pop up

Figure 19 is release strategy form, it is used to add new actors to approval flow.



Figure 20. Configuration/settings page

Figure 20 is a tab menu configuration with 3 submenu groups, status, and status bar.



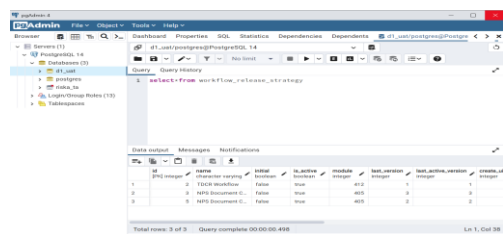


Figure 21. PostgreSQL (PgAdmin4)

Figure 21 shows an overview of the database used on the website system. Users can query the database directly on this sheet. The thing that is needed to connect PyCharm with PostgreSQL is the database port that is set in the configuration. This database port must match the port in PostgreSQL.

### 3.6 Testing

The testing process in this study was tested using a black box indicator. The following is a black box table for testing with the result:

Table 1. Black box indicator for testing

No	Description	Yes	No
1	Users can log in	✓	
2	Users can open ADAM app	✓	
3	Users can create approval matrix	✓	
4	Users can save the approval matrix	✓	
5	Users can edit the approval matrix	✓	
6	Users can create new versions	✓	
7	Users can add roles	✓	
8	Users can choose a company	✓	
9	Users can add associated apps	✓	
10	Users can add approval flow	✓	
11	TDCR application can find the appropriate release strategy	✓	
12	The TDCR application can run buttons according to the Actions set on ADAM	✓	
13	The TDCR application can find the next approver according to the flow set on ADAM	✓	

14	ADAM application can add status as per the needs of the related application	✓
15	ADAM app can add status bar as per related application needs	✓
16	ADAM application can be used by selection company	
17	Users can change the company with only a switch	✓
18	Users can activate version	✓
19	Users can activate release strategy	✓
20	Users can log out	✓

## IV. CONCLUSION

Based on the results of the multi-company system configuration, the development of the ADAM application uses the Waterfall development method with Python, XML, and PostgreSQL programming languages as the database and the testing process uses the Black Box indicator. The following conclusions are obtained:

1. From the results of the multi-company system configuration, the website system can be used by multi-companies by only switching the company without going through a re-login. That way, a system that supports multi companies makes the website system more effective.
2. From the test results on the ADAM application which is dynamic and supports multi companies with the freedom to set the approval flow, status, status bar, and version. Thus, the dynamic approval matrix application is more efficient, easy to use according to needs, and systematic in carrying out the agreed approval flow.
3. From the results of testing the TDCR application using the ADAM application, which is dynamic and supports multi-company, the system can run the company's choice of settings that can be adjusted according to the company's needs. Thus, the application of a dynamic approval matrix with multi-company settings is more efficient in carrying out

approvals on websites used by multi-company.

## BIBLIOGRAPHY

- [1] M Teguh Prihandoyo, “Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web,” *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 126–129, 2018.
- [2] M. Arwanda, “Penerapan Aplikasi Web Dinamis Untuk Sensus Pembaruan Data Kartu Keluarga Penduduk Pada Desa Sei Mencirim,” *Univ. Pembang. Panca Budi Medan*, 2019, [Online]. Available: <https://jurnal.pancabudi.ac.id/index.php/fastek/article/view/1768/1609>.
- [3] S.- Suminten, S. Amelia, and I. D. Sintawati, “Penerapan Enterprise Resource Planning Penjualan Aksesoris Berbasis Odoo,” *JSiI (Jurnal Sist. Informasi)*, vol. 6, no. 1, p. 69, 2019, doi: 10.30656/jsii.v6i1.1052.
- [4] K. Rafinska, “Aplikasi Berbasis Web dan Keunggulannya yang Bisa Anda Ketahui,” 2020. <https://www.online-pajak.com/seputar-efiling/aplikasi-berbasis-web> (accessed Jan. 06, 2021).
- [5] Jotform, “What is an approval matrix?,” 2021. <https://www.jotform.com/blog/approval-matrix/> (accessed Jan. 06, 2022).
- [6] Binus University, “Memahami System Development Life Cycle.” <https://accounting.binus.ac.id/2020/05/19/memahami-system-development-life-cycle/#:~:text=SDLC> (accessed Jun. 26, 2022).
- [7] Y. Handrianto and B. Sanjaya, “Model Waterfall Dalam Rancang Bangun Sistem Informasi Pemesanan Produk Dan Outlet Berbasis Web,” *J. Inov. Inform.*, vol. 5, no. 2, pp. 153–160, 2020, doi: 10.51170/jii.v5i2.66.
- [8] S. Mulyadi, A. Lawi, and A. Achmad, “Implementasi Operasi Relasional Extensible Markup Language ( XML ) Pada Sistem Informasi,” *Jur. Sist. Komput. STMIK Handayani, Makassar*, vol. 8, no. 1, pp. 26–36, 2017.
- [9] M. Fahmi, M. Maisyaroh, I. Komarudin, S. Faizah, and I. Fadhilah, “Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik,” *Bina Insa. Ict J.*, vol. 8, no. 1, p. 53, 2021, doi: 10.51211/biict.v8i1.1517.
- [10] M. Rizal, “Analisis Sentimen Pengguna Twitter Terhadap Objek Pariwisata di Indonesia Menggunakan Algoritma Pengolahan Deep Natural Language dari IBM Insights untuk Twitter,” *Skripsi Univ. Islam Negeri Alauddin Makassar*, pp. 1–82, 2017.
- [11] Y. Heriyanto, “Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT. APM Rent Car.,” *J. Intra Tech*, vol. 2, no. 2, pp. 64–77, 2018.