

# Development of Backend Admin Dashboard for Business Project Monitoring using Scrum Method

Daffa Arya Hutomo Putra <sup>\*1</sup>, Eko Darwiyanto <sup>#2</sup>, Rio Nurtantyana <sup>#3</sup>

<sup>\*#</sup> *School of Computing, Telkom University, Bandung, Jawa Barat, Indonesia*

<sup>1</sup>[aryahutomodaffa@student.telkomuniversity.ac.id](mailto:aryahutomodaffa@student.telkomuniversity.ac.id)

<sup>2</sup>[ekodarwiyanto@telkomuniversity.ac.id](mailto:ekodarwiyanto@telkomuniversity.ac.id)

<sup>3</sup>[nurtayak@telkomuniversity.ac.id](mailto:nurtayak@telkomuniversity.ac.id)

## Abstract

Monitoring business projects is very important to ensure project performance goes according to plan. PT Gerbang Sinergi Prima (GSP) is one of the subsidiaries of the Pension Fund of PT PLN (Persero), which is engaged in Information and Communication Technology. Currently, GSP monitors project performance in a conventional way, which creates several problems, such as difficulties in monitoring project progress and development. This study aims to develop a backend as an admin dashboard for a business project monitoring system using the Scrum method. Unlike other studies, this study also tests the results obtained with three tests, namely Black-Box Testing, Unit Testing, and Performance Testing. As a result, we built a backend admin dashboard for business project monitoring that supports GSP in real-time monitoring the progress of business projects. Furthermore, testing will also be conducted on the backend admin dashboard. The Scrum method was implemented with excellent results. Feature functionality testing using black-box testing gets a percentage of 100%. API testing using unit testing receives a percentage of 100%. The average response time obtained after executing performance testing is 2896.72ms, which means the results are feasible. Overall, the process of development and testing received excellent results.

**Keywords:** Admin Dashboard, Backend, Black-Box Testing, Performance Testing, Scrum, Unit Testing.

## Abstrak

*Monitoring* proyek bisnis sangat penting dilakukan untuk memastikan kinerja proyek berjalan sesuai rencana. PT Gerbang Sinergi Prima (GSP) merupakan salah satu anak perusahaan Dana Pensiun PT PLN (Persero) yang bergerak di bidang Teknologi Informasi dan Komunikasi. Saat ini, GSP melakukan pemantauan kinerja proyek dengan cara konvensional, yang menimbulkan beberapa tantangan, seperti kesulitan dalam memantau kemajuan dan perkembangan proyek. Penelitian ini bertujuan untuk mengembangkan *backend* sebagai *dashboard* admin untuk sistem *monitoring* proyek bisnis dengan menggunakan metode Scrum. Berbeda dengan penelitian lainnya, penelitian ini juga menguji hasil yang didapatkan dengan tiga pengujian, yaitu *Black-Box Testing*, *Unit Testing*, dan *Performance Testing*. Hasilnya, dibangun sebuah *backend* admin *dashboard* untuk *monitoring* proyek bisnis yang mendukung GSP dalam memonitoring progress proyek bisnis secara *real-time*. Selanjutnya, pengujian juga akan dilakukan pada *backend dashboard* admin. Metode *Scrum* diimplementasikan dengan hasil yang sangat baik. Pengujian fungsionalitas fitur menggunakan *black-box testing* mendapatkan persentase 100%. Pengujian *API* menggunakan *unit testing* mendapatkan persentase 100%. Rata-rata response time yang didapat setelah menjalankan *performance testing*, yaitu 2896.72ms, yang berarti hasilnya layak. Secara keseluruhan, proses pengembangan dan pengujian mendapatkan hasil yang sangat baik.

**Kata Kunci:** Admin Dashboard, Backend, Black-Box Testing System, Performance Testing, Scrum, Unit Testing.

## I. INTRODUCTION

Information technology is growing very fast nowadays. This rapidly growing information technology can make it easy for companies to get the information needed quickly. Information Technology has a critical and strategic role [1]. Information technology can make monitoring more efficient [2]. One of the utilizations of information technology, namely technology for monitoring project performance in companies. PT Gerbang Sinergi Prima Bandung (GSP) is a pension fund company which is one of the subsidiaries of the Pension Fund of PT PLN (Persero) in the ICT sector. Based on the results of interviews that have been carried out with the company, there are problems related to project monitoring, which still needs to be done with conventional ways such as business project monitoring using Microsoft Excel. So, the making of this admin dashboard is urgent to facilitate the monitoring of business projects that will and will have been carried out by GSP.

Business Project Monitoring performance can be done in various ways, including making a monitoring dashboard. A monitoring dashboard is an information system that can provide information about the performance of a business project in a company [3]. Business project monitoring in real-time on the monitoring dashboard can overcome delays in project work [4]. During monitoring dashboard development, predicting changes is difficult as needs evolve. One method that can adapt to changes or additions to the needs of the monitoring dashboard is the Agile development method with the Scrum pattern [5]. With Scrum, developers can plan and work on business projects more flexibly when changes must be made during the work [6]. Based on study [5], the Scrum method makes it very easy for developers to get feedback from the user on the features that have been built due to its flexibility so that the development process does not take a long time to complete.

Dashboard development also requires a model to facilitate developers. In this study, the model used is the Unified Modeling Language (UML). UML is a standard language that is quite popular for describing, designing, and analyzing systems on dashboards [7]. Scenarios when users use the system built can also be addressed with UML [8]. There are several types of UML, such as class diagrams, use case diagrams and entity relationship diagrams. Each diagram has a different function in describing the system being used.

Testing for the admin dashboard backend is an essential step in product development. Testing ensures functionality and non-functionality related to the admin dashboard backend that has been developed [9]. The tested admin dashboard backend requires in-depth testing techniques to make the results obtained promising [10]. This study uses three types of testing, namely black-box testing, unit testing, and performance testing. Black-box testing aims to test the functionality of the backend dashboard admin built [11]. Then, unit testing aims to test whether there are errors in the API that has been built [12]. Performance testing aims to test the software's performance towards user requests on the backend admin dashboard [13].

This study aims to discuss the backend development of the admin dashboard that will be used to monitor business projects at GSP using the Scrum method. In addition, this study will also analyze the quality of the backend admin dashboard development results using several tests. In addition, this study has differences with other studies in the testing of dashboards that have been made. This dashboard will be tested with three tests, which are Black-Box Testing, Unit Testing, and Performance Testing.

## II. LITERATURE REVIEW

### *A. Business Project Monitoring*

Business Project Monitoring is very important in a project [14]. Business project monitoring is the activity of monitoring the progress of a business project. These activities must be carried out regularly to check the progress of the business project being executed as scheduled or not. Monitoring a business project that needs to be organized can prevent delays in completing the project being worked on [15]. In addition, business project monitoring can also identify problems when working on projects and corrective actions that can be taken. Therefore, the Company can solve the issues that arise immediately.

### *B. The Development of Monitoring Dashboard with Laravel framework*

A monitoring dashboard is a display system that contains information viewed by users. The monitoring dashboard displays the data and information needed to achieve a goal [3]. The monitoring dashboard represents data and information in the same system [16]. Visualization of information on the monitoring dashboard must be designed effectively so that users can understand the information presented on the monitoring dashboard. Based on a previous study, the monitoring dashboard created will have a function as a display containing information so that everyone can access it quickly and easily understand it [3].

In the development of a monitoring dashboard, there are many frameworks that can be used. Frameworks are needed so that development can be more structured [17]. One of the frameworks, namely Laravel. Laravel uses the MVC concept, which is quite popular with developers [18]. MVC stands for Model, View, and Controller. The model is used to represent the database. The view is used to represent the display that users can see. The controller is used to represent the HTTP process the user requests. Each of these is linked to each other. In managing the data contained in the admin dashboard backend, there are many systems that can be used, such as MySQL. MySQL is a popular system used by software developers because of its ease of use. MySQL is a data storage system that uses the SQL language [19].

### *C. Backend Admin with Application Programming Interface (API)*

There are several parts to developing a website, for example Application Programming Interface (API). The backend is the part that is responsible for managing the server and database that will be used on a website [20]. Managing data contained in the database, such as adding, changing, and deleting data, is the responsibility of a backend. Furthermore, the backend also needs to develop the APIs. API is an interface that can be used as a link between client and server [21]. APIs can simplify developers with simple operations, such as GET, POST, PUT, and DELETE. In addition, data security in the database is also an important aspect that must be taken seriously by the backend.

### *D. The advantage of the Scrum method*

The Scrum method is one of the popular agile development tools utilized by software developers. Scrum is usually used in challenging software development [22]. The method is flexible when there are additions or changes in product functionality [23]. Three significant roles are in the scrum method such as, product owner, scrum master, and development team [24]. The responsibility of the product owner is to optimize the product created. The Scrum master is aware of the work of the Scrum team. The development team will develop the product that has been discussed with the product owner. Several activities are executed in Scrum, which are product backlogs and sprints. In the Sprint, there are sprint planning, daily Scrum, sprint review, and Sprint retrospective [25]. Each of these stages has different tasks to perform during the execution. Each of these phases has different tasks to perform during execution. The scrum method focuses on iterative development, so these phases will continuously repeat until the final product has fulfilled the product owner's needs.

### *E. The software model using Unified Modeling Language*

Unified Modeling Language is a model that can be utilized to structure the design of a software system [26]. UML can help users understand the system's design to be developed. There are many types of UML, such as activity diagrams, entity relationship diagrams, use case diagrams and class diagrams. These diagrams facilitate system development and make it more structured. With the use of UML, the software developed is expected to match the user's needs [27].

#### F. Software testing

Testing is the final step in dashboard development. Dashboard testing is an activity that discovers errors that may occur when the dashboard is finished [11]. These errors could be fatal if the developer fails to fix them. So, developers need to ensure that the dashboard built is as expected and can run well [28]. To prevent errors, various kinds of tests can be applied to the dashboard, such as black-box testing, unit testing, and performance testing. Black-Box Testing can be used to test based on functional requirements [11]. Unit Testing can be used to test the program code that was built effectively or not [12]. Performance testing can be used to test how long the response time required by users. There are optimal response time criteria to be considered by users, which is 8 seconds or 8000ms [29].

### III. RESEARCH METHOD

#### A. System Development Method

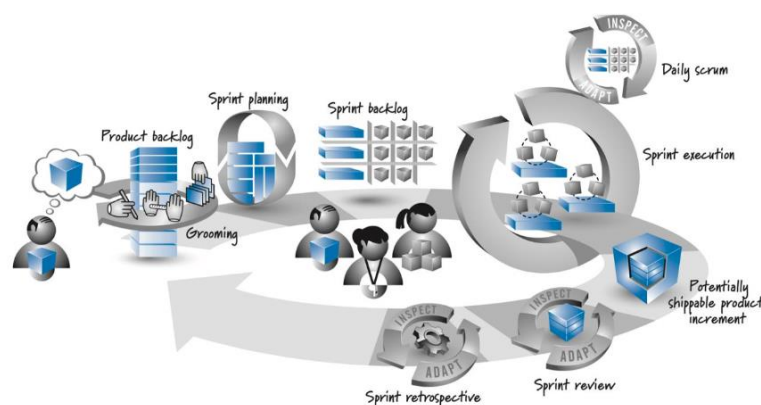


Fig. 1. Scrum Phases [30]

Based on the analysis of the current problems at GSP, the developer decided to use the Scrum method, which is expected to facilitate the developer in the development of the backend admin dashboard. The scrum method was selected due to the simplicity of adaptability with changes requested by the user compared to other methods. The products created using the Scrum method are also high quality [30]. The scrum method is also highly transparent regarding work progression, making it easier for the development team to accomplish the tasks collaboratively. This backend admin dashboard development will also use MySQL database and Laravel Framework. Based on Figure 1, the following is an explanation of the phases executed by the developer.

1) *Product Backlog*: Activities performed by the development team during this phase include identifying and analyzing the needs needed to solve the problem. The development team also discussed the required needs with the product owner. Then, the development team will list requirements that must be fulfilled during the work.

- 2) *Sprint Planning*: After listing the product backlog items, the development team will have an internal conversation to plan the sprint activities that will be executed. The development team will identify the phases of the work. The development team will also estimate the time needed during the progress. The sprint backlog will include the tasks and time estimation for each sprint.
- 3) *Sprint*: According to the structured sprint backlog, the developer starts the work in detail in this phase. For each day, the development team will also perform a daily scrum. Daily scrum will discuss the daily activities of the development team and discuss if there are problems when working. In addition, the daily scrum development team will check the progress of the work being done or done by the development team. These activities will keep repeating until the work on the sprint backlog is complete.
- 4) *Sprint Review*: After all activities on the sprint backlog are completed, it will move to the sprint review phase. In the sprint review, the development team will recheck all the work completed by the sprint backlog.
- 5) *Sprint Retrospective*: At this phase, the development team will conduct an evaluation based on the previous phases that have been executed. The development team will discuss what went on during the sprint implementation and what problems appeared during the work. In addition, the development team will discuss the following plans. This sprint activity will be repeated until the entire product backlog has been compiled, the product owner accepts the resulting product, and there are no problems or errors in the resulting product.

After all sprints are completed, the monitoring dashboard that has been developed will be tested using black-box testing, unit testing, and performance testing. The functionalities of all features contained in the monitoring dashboard will be tested using black-box testing. Then, the built API will be tested using unit testing. In addition, there is also an analysis of response time testing using performance testing. The analysis will determine whether the built monitoring dashboard has met the standard time required by the system to manage user requests. After all the tests are carried out and good results are obtained, the development team will conduct a product demo for the product owner.

*B. Scrum Team and the Framework*

Table I explains the Scrum team that will develop the project to solve the problem. There are three sections, which are product owner, scrum master, and development team. GSP, as a product owner, has a problem with business project monitoring. Mr. Eko Darwiyanto, as a scrum master, helps with mentoring and support during Scrum. Then, Bulan Ayu, Daffa Arya, and Rizky Sampoerna are the development teams responsible for their respective tasks in this project. For this study, the project is related to developing a backend admin dashboard that will be used to manage project data at GSP. The sprint process was performed four times with each sprint of 21 days.

TABLE I  
 SCRUM TEAM AND THE FRAMEWORK

Product Owner	PT Gerbang Sinergi Prima (GSP)
Scrum Master	Mr. Eko Darwiyanto
Development Team	1. Ms. Bulan Ayu Sari Muntai (Development of Mobile Application for Project Monitoring PT Gerbang Sinergi Prima) 2. Mr. Daffa Arya Hutomo Putra (Development of Backend Admin Dashboard for Business Project Monitoring using Scrum Method) 3. Mr. Rizky Sampoerna (Development of User Dashboard frontend for Project Monitoring PT Gerbang Sinergi Prima)
Project	Development of Backend Admin Dashboard for Business Project Monitoring using Scrum Method
Sprint	4 times (1 sprint run for 21 days)

#### IV. RESULTS AND DISCUSSION

##### A. Functional Requirements Results

Table II shows several of the functional requirements that are required to overcome existing problems based on the results of discussions that have been held with the product owner. Functional requirements include services that will be provided in this dashboard system, as well as a list of features that can be done by the admin on this dashboard.

TABLE II  
FUNCTIONAL REQUIREMENTS RESULTS

No	Functional Requirement	Description
1	The system can perform login for admin	Only user who have the admin role can enter the dashboard.
2	The system can display the main dashboard page	This main dashboard page contains 4 buttons that read the amount of data from projects, services, customers, and users table. There is also a To-Do List that will be used to record things that need to be done and there is also a calendar to check the date.
3	The system can display project pages, project details, services, customers, and users.	Each page has various data. Data on individual page can be managed, such as adding, changing, and deleting data.
4.	The system can display the notification page and activity logs	both pages have their own uses for recording changes
5.	The admin can process project data, project details, customers, services, users, notifications, and activity logs.	Data in the database can be added, changed, and deleted by admin
6.	The admin can check notifications	Notifications delivered by the system can be seen by the admin
7.	The admin can view and delete all activity logs	Activities performed can be recorded and viewed by the admin. In addition, all activities can also be deleted by the admin.

##### B. System Design Results

1) *Use Case Diagram*: Based on the results of preparing functional requirements, developers will design use case diagrams to represent the functionality [26]. This use case diagram illustrates the connection between the user and the system. On this dashboard, there is only one user, namely the admin. The admin can process data on the dashboard. In addition, the admin can also check the activity history in the dashboard and check notifications when project data and project details are added, changed, or deleted. Figure 2 illustrates the use case diagram on the backend of this admin dashboard.

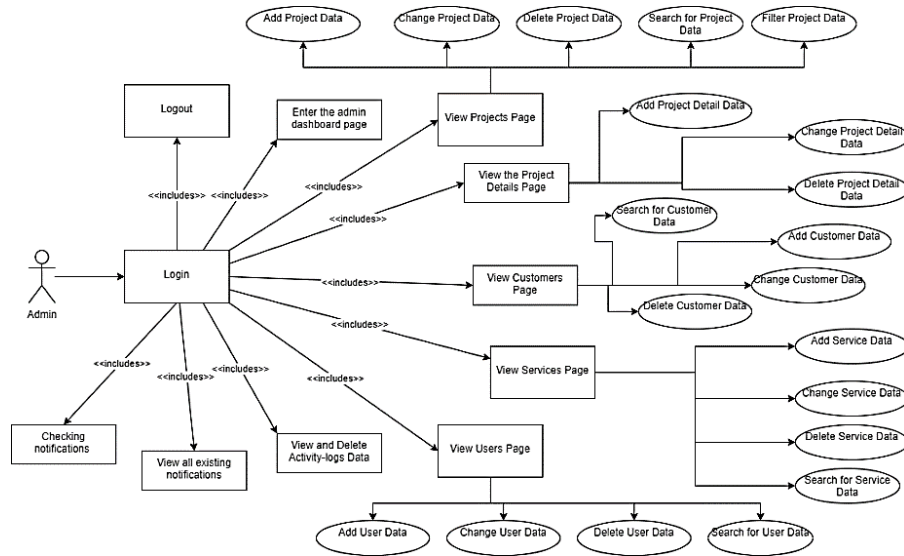


Fig. 2. Use Case Diagram

2) *Activity Diagram*: Based on the results of the preparation of functional requirements, the developer will design an activity diagram. An activity diagram is the flow of a system on the dashboard [27]. This activity diagram aims to help users with the activities that will be carried out when accessing the dashboard. Figure 3 illustrates one of the activity diagrams that have been made. The figure illustrates one of the activity diagrams that has been designed. The figure explains the flow of activities that will be carried out when the admin logs into the dashboard.

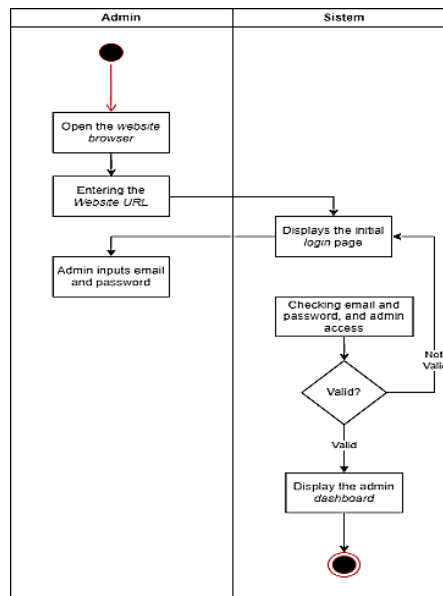


Fig. 3. Activity Diagram

3) *Class Diagram*: The developer will create a class diagram based on the results of preparing functional requirements. The class diagram describes the class structure used to develop this dashboard [26]. This dashboard development uses the Laravel framework, so there are models, views, and controllers. This

class diagram is divided into two parts, namely the model and controller. The Model Class Diagram is used to describe the management of data in the database, while the controller class diagram is used to describe the HTTP process related to requests from users.

4) *Entity Relationship Diagram*: Based on the results of the preparation of functional requirements, the developer will create an entity relationship diagram. ERD is a diagram that illustrates the relation of one class to another class in the database. There are eight classes that are related to each other in the development of this dashboard. Figure 4 illustrates the ERD used.

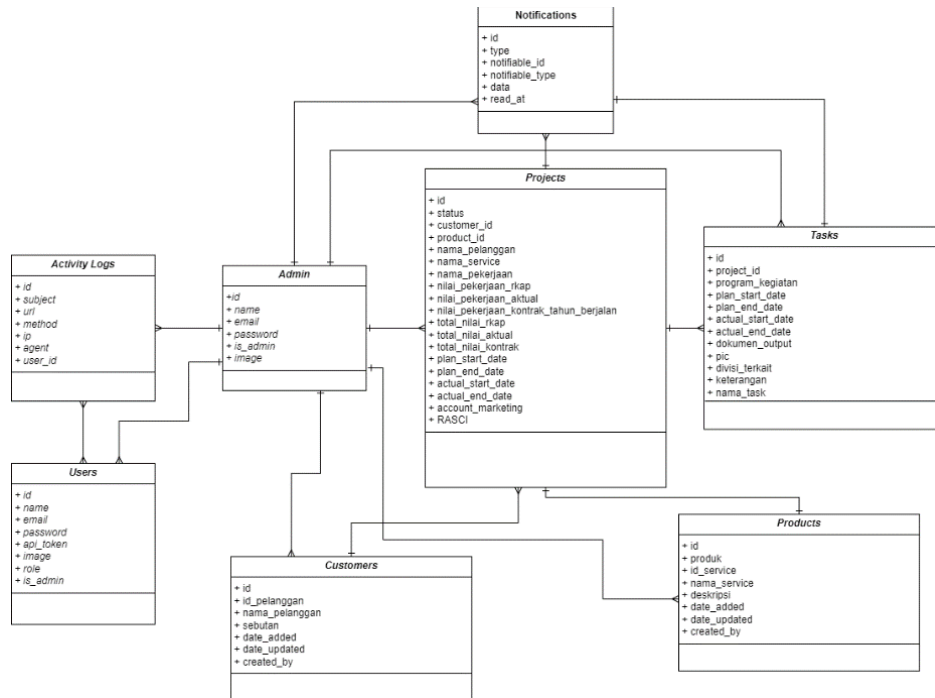


Fig. 4. Entity Relationship Diagram

C. Scrum Implementation Results

1) *Product Backlog*: In implementing scrum, the initial stage must be to compile a product backlog. The product backlog is created by analyzing existing functional requirements [24]. All work in the product backlog must be completed within the period discussed. Table III describes the tasks that must be done.

TABLE III  
FUNCTIONAL REQUIREMENTS RESULTS

No	Task List	Period
1	Create a display for login, and build APIs for login and logout	March 17, 2024 -
2	Create a display for the main dashboard and features on the main dashboard	
3	Create a display for the services page, and build a CRUD API for the services table	
4	Create a display for the users' page, and build a CRUD API for the users table	
5	Create a display for the customers page, and build a CRUD API for the customers table	



6	Create a display for the projects page, and build a CRUD API for the projects table	June 12, 2024
7	Create a display for the project details page and build a CRUD API for the project details table based on the project name.	
8	Create a notification feature based on the projects and project details table and create an API to view notifications.	
9	Create an activity-logs feature to log activities conducted by admins.	

2) *Sprint Planning*: Based on the list of tasks in the product backlog, developers create a sprint backlog for each sprint. This Sprint Backlog is a group of tasks that the developers must accomplish during the sprint. Table IV describes some of these tasks and the durations for each sprint.

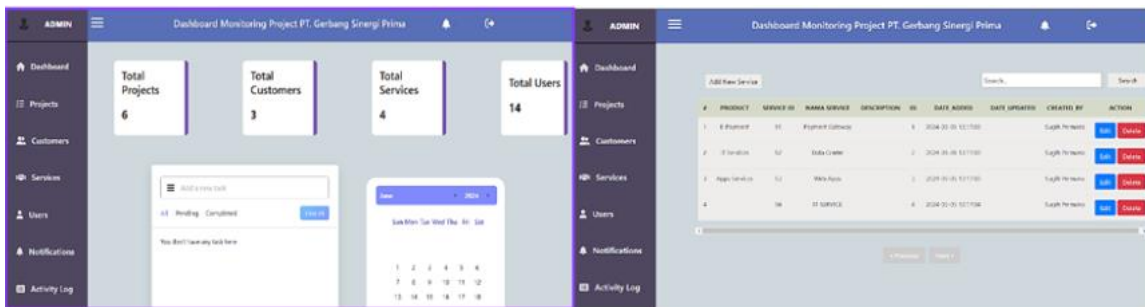
TABLE IV  
SPRINT PLANNING

Sprint	Product Backlog	Task	Period
Sprint Planning Sprint I	Create a display for login, and build APIs for login and logout	1. Connect to the database	March 17, 2024 - April 7, 2024
		2. There are fields to fill in email and password	
		3. Gives an error output if the email or password entered is incorrect	
		4. Return to login page if logout is successful	
		5. Preparing APIs for login and logout	
	Create a display for the main dashboard and features on the main dashboard	1. There are features for to-do lists, checking calendars, and features for viewing the total data of projects, customers, services, and users.	
2. There is a sidebar that is connected to each page that will be created			
3. There is a display for notification and logout features			
Sprint Planning Sprint II	Create a display for the services page, and build a CRUD API for the services table	1. Connect to the database	April 8, 2024 - April 29, 2024
		2. There are fields that match the requirements for the services page	
		3. Setting up CRUD APIs for services table	
	Create a display for the users' page, and build a CRUD API for the users table	1. Connect to the database	
		2. There are fields that match the requirements for the users' page	
		3. Setting up CRUD APIs for users table	
	Create a display for the customers page, and build a CRUD API for the customers table	1. Connect to the database	
		2. There are fields that match the requirements for the customers page	
		3. Setting up CRUD APIs for customers table	
Sprint Planning Sprint III	Create a display for the projects page, and build a CRUD API for the projects table	1. Connect to the database	April 30, 2024 - May 21, 2024
		2. There are fields that match the requirements for the projects page	
		3. Setting up CRUD APIs for projects table	
	Create a display for the project details page and build a CRUD API for the project details table based on the project name.	1. Connect to the database	
		2. There are fields that match the requirements for the project details page	
		3. Setting up CRUD APIs for project details table	
Sprint Planning Sprint IV	Create a notification feature based on the projects and project details table and create an API to	1. Notifications are sent to all registered users	
		2. Detailed and clear notification data	

	view notifications.	3. Setting up CRUD APIs for notifications	May 22, 2024 - June 12, 2024
	Create an activity-logs feature to log activities conducted by admins.	1. All admin activities are recorded in detail	
		2. Easy to read and simple to understand display	

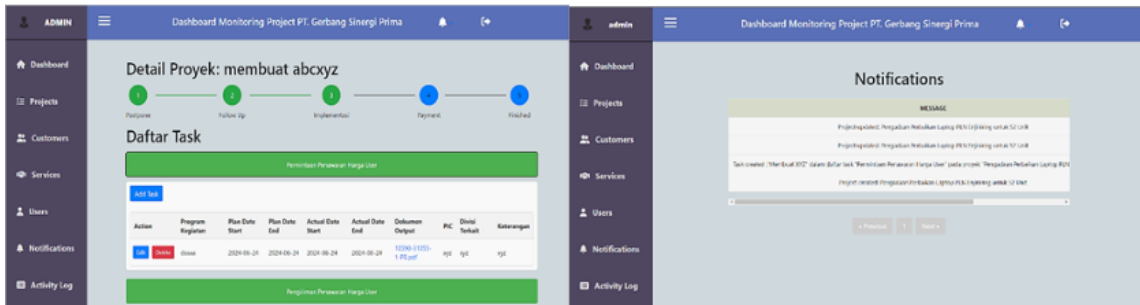
3) *Sprint*: All activities in the sprint backlog listed in the sprint planning will be done at this sprint stage. The following are the results that the developer has completed.

- a) *Sprint I*: The Sprint I task resulted in the development of login and logout views for the admin. In addition to these views, API development for login and logout was also done in sprint I. Then, the development of the main dashboard view was also carried out in sprint I. Figure 5a represents an example of a display view that has been created, the figure describes the main dashboard page, which contains several features, such as the to-do list feature and the calendar checking feature.
- b) *Sprint II*: The Sprint II task resulted in the development of the services, users, and customers pages. API creation for each of these pages was also done in sprint II. Figure 5b represents an example of the view that was developed. The figure shows that the user can see all the data in the table. In addition, users can add, change, and delete service data, as well as perform searches related to the existing data.
- c) *Sprint III*: The sprint III task resulted in the development of the projects and project detail pages. Project details are based on the job names in the projects table. API development for each of these pages was also done in sprint III. Figure 5c represents an example of the view that has been developed. The figure shows the user can see all the data in the table. In addition, users can also add, change, and delete project detail data.
- d) *Sprint IV*: The sprint IV task resulted in the development of the notification page and activity logs. API development for notifications was also done in sprint IV. Figure 5d represents an example of the view that was created. The figure shows that the user can check all available notifications.



a. Main Dashboard Page (Sprint I)

b. Services Page (Sprint II)



c. Project Details Page (Sprint III)

d. Notifications Page (Sprint IV)

Fig. 5. Sprint activities

4) *Sprint Review:* After completing the entire product backlog, the developer will double-check the results done in the sprint review. The whole sprint went very well. All work can be completed according to the required criteria. The sprint period is also in accordance with what was discussed by the development team.

5) *Sprint Retrospective:* A Sprint retrospective is a developer evaluation phase during the sprint. There are several evaluations that developers can take from the sprints that have been run, such as team communication that still needs to be improved and learning related to API development that needs to be more profound. These problems are discussed again with the development team to find a way out.

D. Testing Results

1) *Black-Box Testing:* This black-box test focuses on the functionality of the features built on this dashboard. The goal is to check whether the developed features are as expected. Table V describes the Black-Box Testing done on this admin dashboard after all sprints finished.

TABLE V  
BLACK-BOX TESTING

No	Test Class	Initial Condition	Scenario	Expected Results	Results
1	Login Page Function	Login Page	1. Fill in the correct email and password and be registered in the database as an admin	Login successfully and enter the admin dashboard page	100% Success
			2. Fill in the correct Username and password but not registered as admin in the database	The login process fails and the notification "You do not have access to this page" appears.	100% Success
			3. Username and password are incorrect and not registered in the database	The login process fails and the notification "Login Failed. Email or Password is incorrect."	100% Success
2	Dashboard Page Function	Dashboard Page	1. Pressing the logout button/icon	The logout process is successful and returns to the login page	100% Success
			2. Pressing the notifications button/icon	A display appears with the words "No notifications" if there are no notifications, and a list of notifications appears if there are notifications.	100% Success
3	Customers Page Function	Customers Page	1. Fill in the data you want to search then click the Search button	Display the searched data in the Customers table	100% Success

		Add New Customer Page	2. Fill in the required data and click the Submit button	Display the Customers page and the alert "Customer Created Successfully."	100% Success
		Edit Customer Page	3. Change one of the data and click Submit	Display the Customers page and the alert "Customer Updated Successfully."	100% Success
		Customers Page	4. Select one of the rows in the Customers table and press the Delete button.	Display the Customers page and the alert "Customer Deleted Successfully."	100% Success
4	Users Page Function	Users Page	1. Fill in the data you want to search then click the Search button	Display the searched data in the Users table	100% Success
		Add New User Page	2. Fill in the required data and click the Submit button	Display the Users page and a "User Created Successfully" alert appears.	100% Success
		Edit User Page	3. Change one of the data and click Submit	Display the Users page and the alert "User Updated Successfully."	100% Success
		Users Page	4. Select one of the rows in the Users table and press the Delete button.	Display the Users page and the alert "User Deleted Successfully."	100% Success
5	Services Page Function	Services Page	1. Fill in the data you want to search then click the Search button	Display the searched data in the Services table	100% Success
		Add New Service Page	2. Fill in the required data and click the Submit button	Display the Services page and a "Product Created Successfully" alert appears.	100% Success
		Edit Service Page	3. Change one of the data and click Submit	Display the Services page and the alert "Product Updated Successfully."	100% Success
		Services Page	4. Select one of the rows in the Services table and press the Delete button.	Displays the Services page and the alert "Product Deleted Successfully."	100% Success
6	Project Page Function	Projects Page	1. Fill in the data you want to search then click the Search button	Display the searched data in the Projects table	100% Success
		Create New Project Page	2. Fill in the required data and click the Submit button	Return to the Projects page and the notification "Project created successfully."	100% Success
		Edit Project Page	3. Make changes to the data you want to change then click the Update button	Return to the Projects page and the notification "Project updated successfully."	100% Success
		Projects Page	4. Clicking the Delete button in one of the rows in the Projects table on the Projects page	A display appears with the words "Project deleted successfully."	100% Success
		Projects Page	5. Clicking on one of the data in the job name column in the projects table	Display the project details page according to the selected job name data	100% Success
		Detail Projects Page	6. A pop-up display of adding a new Task appears, enter the data then click the Submit button	The newly registered task appears in one of the selected "Task Lists"	100% Success
			7. Edit Task pop up display, make changes to the data and click the Save Changes button.	The data in the table changes according to the changes that have been made.	100% Success
			8. Pressing the Delete button on one of the tasks and an alert appears "Are you sure you want to delete this task?" then click Ok.	"Task successfully deleted" alert appears	100% Success

7	Notifications Page Function	Dashboard Page	1. Clicking on "Notifications" in the sidebar	Display the Notifications page with all data	100% Success
8	Activity Log Page Function	Activity Logs Page	1. Click the delete all button on the Activity Log page, then a pop up appears and click the Delete All button.	Display the Activity Log page with all table data deleted	100% Success

Based on the explanation in Table V above, all features tested, starting from the login feature, main dashboard page, customers page, users page, services page, projects page, project details page, notifications, and activity log page, it can be concluded that all tested features have achieved 100% success. All features ran adequately, and no errors were found.

2) *Unit Testing*: Unit Testing focuses on testing API units built on the admin dashboard backend. This test aims to check whether the API created is by its function or not. This test uses PHPUnit. PHPUnit is a reasonably popular feature provided by Laravel. Figure 6 is one example of the results of unit testing that the developer has run.

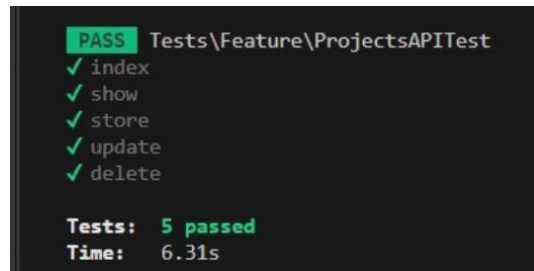


Fig. 6. API Projects Unit Testing Results

This test focuses on the Create, Read, Update, and Delete APIs in the projects table that has been built. In the study there were 28 APIs with different functions with status codes 200 / 201 explain that the API has succeeded in making requests. In average, the testing process required for unit testing this API project is 6.31 seconds. Based on a total of 28 APIs built, all APIs tested using unit testing received a "PASS" result with a percentage of 100%. Which means no errors were found in the coding. It can be concluded that the APIs can function correctly.

3) *Performance Testing*: Performance Testing focuses on checking the response time to data appearance, addition, change, and deletion features. This Performance Testing aims to calculate the average response time when the admin uses these features. In this performance testing scenario on the admin dashboard, testing will focus on the projects table. The reason for choosing the projects table is because this table is the main feature in the built admin dashboard. For example, table VI is the result of the analysis for adding data process to the project table.

Table VI describes five trials of the data addition feature in the projects table. The average value of the five trials is 2896.72ms. The average value is obtained from the total time divided by the number of trials. In detail, the minimum value obtained is 2320.52ms and the maximum value obtained is 3774.57ms within five trials.

TABLE VI  
PERFORMANCE TEST RESULTS OF THE CREATE PROJECTS TABLE FEATURE

Number of Trial	Total Time
1	3147.24ms
2	2376.21ms
3	3774.57ms
4	2320.52ms
5	2865.04ms
AVERAGE	2896.72ms

In the previous study, the response time of more than 8 seconds or 8000ms makes users uncomfortable and makes them forget what the system is doing [29]. From the test above, the feature scenario with the most complicated process (i.e. the process of add the data) compared to other features does not get a response time of more than 8000ms. The features tested have a response time that is still acceptable and good enough.

## V. CONCLUSION

In this study, the development of the backend admin dashboard for business monitoring project at GSP using the Scrum method went very well. Regarding the contribution of this study, we also tested with three mechanisms, such as Black-Box Testing, Unit Testing, and Performance Testing, which are different from other studies that mostly only use one of them. The implementation of the Scrum method in the development of the backend admin dashboard shows promising results. The Scrum method is compelling enough to be used to develop a complex dashboard due to its level of flexibility. In the implementation, The Laravel Framework uses the concept of Model, View, and Controller so that the writing of program code is arranged more neatly. The application of the Scrum methodology provides outstanding progress. It achieved a 100% success rate in performing functional testing of various features using black box testing. API testing done by unit testing also achieved a 100% success rate. The average response time recorded during performance testing was 2896.72ms, which indicates satisfactory performance. In summary, the development and testing process achieved excellent results and the product owner can use the backend dashboard admin well without errors. Despite all the tests that were carried out getting excellent results, security testing of the API built has yet to be carried out in more depth on this dashboard. Therefore, further research is needed to perform security testing on APIs that are more in-depth on this backend admin dashboard. By performing security testing on the API, the system built can provide maximum protection to the available data.

## ACKNOWLEDGMENT

I want to thank PT Gerbang Sinergi Prima for allowing me to conduct this study and for their helpful support. This study would not have been possible without the help and support of PT Gerbang Sinergi Prima.

## REFERENCES

- [1] Yusar Sagara. "The Effect of Implementation Accounting Information System and Competence of Human Resources on the Quality of Financial Reporting," *Research Journal of Finance and Accounting*, vol. 6, no. 10, pp. 111-115. 2015.
- [2] Lanto Ningrayati Amali, Muhammad Rifai Katili, Sitti Suhadal, and Tri Alfandra Labuga. "Business process monitoring system in supporting information technology governance," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2884-2891, Oct. 2021. <https://doi.org/10.11591/eei.v10i5.3147>
- [3] Desi Mega Paramita, and Ahmad Nurul Fajar. "Analysis of Network Performance Management Dashboard," *International Journal of Mechanical Engineering and Technology (IJMET)*, vol. 10, no. 3, pp. 952-963, Mar. 2019.

- [4] Jawa Anak Gara, Rozana Binti Zakaria, Eeydzah Aminudin, Jeffryl Azniel Adzar, and Omar Sedeeq Yousif. "The Development of Real-Time Integrated Dashboard: An Overview for Road Construction Work Progress Monitoring," *Journal of Hunan University Natural Sciences*, vol. 48, no. 5, pp. 127-137, May. 2021.
- [5] Putu Adi Guna Purmana. "Scrum Method Implementation in a Software Development Project Management," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 9, pp. 198-204. 2015. <https://doi.org/10.14569/ijacsa.2015.060927>
- [6] Jerin Jose Philip, Surya Babu, Zacharia Alexander, Shilpa K. Mohanan, and Neena Joseph. "Design and Development of Agile/Scrum Dashboards," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 7. 2021.
- [7] Fajar Mahardika, and Muhammad Labib Abdillah. "Design of Unified Modeling Language Information System for Motorcycle Unit Selling and Buying Transactions using the Waterfall Method," *Journal of Information Systems*, vol. 1, no. 2, pp. 37-48, Feb. 2024. <https://doi.org/10.56211/hanif.v1i2.15>
- [8] Hatice Koc, Ali Mert Erdogan, Yousef Barjakly, and Serhat Peker. "UML Diagrams in Software Engineering Research: A Systematic Literature Review," *Proceedings*, 74(1), 13. Presented at the *7th International Management Information Systems Conference*, Online, 9-11 December 2020. <https://doi.org/10.3390/proceedings2021074013>
- [9] Enjelina Sianturi. "Boundary Value Analysis and Decision Table Testing Methods in Software Testing," *International Journal of Information Technology and Education (IJITE)*, vol. 1, no. 3, pp. 124-129, Jun. 2022. <https://doi.org/10.62711/ijite.v1i3.68>
- [10] Supriyono. "Software Testing with the approach of Blackbox Testing on the Academic Information System," *International Journal of Information System & Technology*, vol. 3, no. 2, pp. 227-233. 2020.
- [11] Putri Kusuma Ayuningtyas, Dwi Atmodjo. WP, and Pratiwi Rachmadi. "Performance and Functional Testing with the Black Box Testing Method," *International Journal of Progressive Science and Technologies (IJPSAT)*, vol. 39, no. 2, pp. 212-218, Jul. 2023. <https://doi.org/10.52155/ijpsat.v39.2.5471>
- [12] Easter Viviana Sandin, Noraniah Mohd Yassin, and Radziah Mohamad. "Comparative Evaluation of Automated Unit Testing Tool for PHP," *International Journal of Software Engineering and Technology*, vol. 3, no. 2, pp. 7-11, Dec. 2016.
- [13] Arlinta Christy Barus, Eka Stephani Sinambela, Ivani Purba, Jhonathan Simatupang, Monika Marpaung, and Nancy Pandjaitan. "Performance Testing and Optimization of DiTenun Website," *Journal of Applied Science, Engineering, Technology, and Education*, vol. 4, no. 1, pp. 45-54. 2022. <https://doi.org/10.35877/454RI.asci841>
- [14] Rosine Mukamwiza, and Dr. Khan Jean De Dieu Hakizimana. "The Role of Monitoring and Evaluation on Effective Project Implementation," *International Journal of Scientific and Research Publication*, vol. 11, no. 7, pp. 298-305, Jul. 2021. <http://dx.doi.org/10.29322/IJSRP.11.07.2021.p11537>
- [15] Moses Njiru, and Erastus Thoronjo. "Analysis of Monitoring and Evaluation Practices on Performance of Non-Governmental Organizations' Project in Kiambu County, Kenya," *Journal of Business Management and Economic Development*, vol. 2, no. 1, pp. 1-12, Jan. 2024. <https://doi.org/10.59653/jbmed.v2i01.308>
- [16] Rayhan Alya Chaerul, Widhy Hayuhardhika Nugraha Putra, and Buce Trias Hanggara. "Utilizing of the Trello API Within the Development of a Monitoring Information System Recording of Project Activities Using a Website-Based Kanban System (Case Study: Electrical Project of PT. XYZ)," *Journal of Information Technology and Computer Science (JITeCS)*, vol. 6, no. 2, pp. 146-157, Aug. 2021. <https://doi.org/10.25126/jitecs.202162289>
- [17] Syaeful Anas Aklani, and Jason Aaron Yang. "Performance Analysis Between Interpreted Language-Based (Laravel) and Compiled Language-Based (GIN) Web Frameworks," *Computer Based Information System Journal*, vol. 11, no. 1, pp. 12-16, Mar. 2023. <https://doi.org/10.33884/cbis.v11i1.6583>
- [18] Defni, and Tri Lestari. "Implementing Laravel Framework for E-Commerce: Case Study at Indonesian Farmer Shop Center," *International Journal of Advanced Computing Science and Engineering*, vol. 2, no. 1, pp. 14-20, Apr. 2020. <https://doi.org/10.30630/ijasce.2.1.46>
- [19] Rizky Maulana Fanani, and Irwan A. Kautsar. "Design and Build Features Chat on Renting Book Application Based Web Technology," *Journal for Technology and Science*, vol. 1, no. 1, pp. 117-131, Mar. 2024. <https://doi.org/10.61796/ipteks.v1i1.95>
- [20] I. H. Madurapperuma, M. S. Shafana, and M. J. A. Sabani. "State-of-Art Frameworks for Front-end and Back-end Web Development," *Proceedings of Papers, 2<sup>nd</sup> International Conference on Science and Technology*, pp. 62-67, Aug. 2022.
- [21] Mario Dudjak, and Goran Martinovic. "An API-first Methodology for Designing a Microservice-based Backend as a Service Platform," *Information Technology and Control*, vol. 49, no. 2, pp. 206-223, Feb. 2020. <http://dx.doi.org/10.5755/j01.itc.49.2.23757>
- [22] Hestri Apriliani Widowati, and Andhik Budi Cahyono. "Implementing of Scrum for Developing the Data Visualization Dashboard for the PLN UP3 Yogyakarta Project," *Journal of Social Science*, vol. 5, no. 4, pp. 1234-1243, Jul. 2024. <https://doi.org/10.46799/jss.v5i4.897>
- [23] Amat Basri, Dewi Marini Umi Atmaja, Arif Rahman Hakim, and Andreas Rino Sanjaya. "Application of Scrum Methodology in The Design of Micro, Small, and Medium Enterprise Systems: A Case Study on Laundry Services," *International Journal Software Engineering and Computer Science (IJECS)*, vol. 3, no. 3, pp. 537-544, Dec. 2023. <https://doi.org/10.35870/ijsecs.v3i3.1968>
- [24] Florencia Vega, Guillermo Horacio Rodriguez, Fabio Rocha, and Rodrigo Pereira dos Santos. "Scrum Watch: a tool for monitoring the performance of Scrum-based work teams," *Journal of Universal Computer Science*, vol. 28, no. 1, pp. 98-117, Jan. 2022. <http://dx.doi.org/10.3897/jucs.67593>
- [25] Fitrotul Khasanah, and Sarmini. "Implementation of Scrum Method for Keep Wallet Application Design," *International Journal of Informatics and Information Systems*, vol. 6, no. 3, pp. 103-113, Sept. 2023. <https://doi.org/10.47738/ijiis.v6i3.170>
- [26] Shahrin Azuan Nazeer. "Design of Strategic Management System for Northern Border University using Unified Modeling Language," (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 138-147. 2018. <https://dx.doi.org/10.14569/IJACSA.2018.091017>
- [27] Djarot Hindarto, and Mochamad Hariadi. "Information System Design at FGH Stores with Unified Modelling Language," *Journal of Computer Networks, Architecture and High Performance Computing*, vol.5, no. 2, pp. 623-633, Jul. 2023. <https://doi.org/10.47709/cnahpc.v5i2.2702>
- [28] Arief Agus Sukmandhani, Yemima M. Gaesela, Mery, Edgardo, Kevin, and Nico Fernando. "Testing and Performance Evaluation of E-Commerce Web Sites," *Journal of Business and Audit Information System (JBASE)*, vol. 6, no. 2, pp. 34-45. 2023. <https://doi.org/10.30813/jbase.v6i2.4673>

- [29] Vuk Vukovic, Jovica Durkovic, and Jelica Trninc. "Defining Performance Criteria and Planning Performance Tests for the Exam Registration Software," *International Scientific Journal of Management Information System*, vol. 9, no. 2, pp. 15-19, Jun. 2014.
- [30] Kenneth S. Rubin. "Essential Scrum: A Practical Guide to the Most Popular Agile Process," *Pearson Publisher*. 2012.