Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

Improving Campus Security with Web-Based Motorcycle Parking Information System using Personal Extreme Programming Method

Sagita Wulandari*, Ida Bagus Ketut Widiartha, Royana Afwani

Departement of Informatics Engineering, Faculty of Engineering, University of Mataram, Mataram
Jl. Majapahit No.62, Gomong, Kec. Selaparang, Mataram City, West Nusa Tenggara, Indonesia
Email: [1,*]sagitaawd@gmail.com, [2]widi@unram.ac.id, [3]royana@unram.ac.id
Correspondence Author Email: sagitaawd@gmail.com
Submitted: **07/07/2024;** Accepted: **17/07/2024;** Published: **20/07/2024**

Abstract−In this modern era, motorcycles are a primary means of transportation used by many people in Indonesian society, including students who use them to access campus. However, the increase in motorcycles also raises safety issues and the availability of parking spaces on campus. The parking system needs to be improved at the Faculty of Engineering, University of Mataram, because it does not use an automatic checking system, resulting in concerns about vehicle safety. This research aims to build a Web-based Motorcycle Parking Information System to store real-time rider data and detect faces and police numbers through CCTV. The Personal Extreme Programming (PXP) method is used as a development method because of its high level of flexibility and its ability to adapt to changing needs. This system was developed using the Bootstrap framework for the front end and Laravel for the back end, with MySQL Workbench as a server to manage the database. System testing uses the Black Box Testing method to ensure each unit or module functions according to predetermined needs. The test results show that all features function validly and by the expected specifications. This research provides an effective solution for improving parking security and parking space management on campus, thus providing a sense of comfort and security for students and all parking users at the Faculty of Engineering, University of Mataram.

Keywords: Information System; Motorcycle; Parking; Personal Extreme Programming; Black Box Testing

1. INTRODUCTION

Two-wheeled motorized vehicles, especially motorcycles, are one of the main transportation most widely used by people in Indonesia [1], [2]. This two-wheeled vehicle is the most frequently used alternative vehicle because of its very practical and efficient use. Most students choose to use motorbikes to access campus because of its convenience [3], [4]. Based on data from the Indonesian Central Bureau of Statistics (BPS), in 2022, the number of two-wheeled vehicles in Indonesia reached 125,305,332 units [5]. This number has increased by 4.38% from the number of vehicles in 2021, which is 120,050,112 units [6]. This number proves that every year, Indonesia experiences an increase in the use of two-wheeled motorized vehicles.

As the use of motorcycles increases, parking security and the availability of parking spaces on campus are becoming more important issues [7]. Many colleges now face the challenge of providing adequate parking spaces for students. The unavailability of sufficient parking spaces often creates new problems, such as illegal parking that can disrupt campus traffic and increase the risk of theft and vehicle damage [8], [9].

A parking lot is one of the public facilities needed by many people to leave their vehicles safely [10], [11]. Parking security is one of the important aspects that must be considered, especially in the university area. Safe parking can provide a sense of comfort for all parties who use it, especially students, staff, and other campus visitors. In addition to creating a sense of comfort, safe parking can also prevent theft, damage, or accidents that can harm parking users. To obtain this security, an accurate system is needed to meet the needs and create a sense of security for vehicle owners [4]. Furthermore, the security of parking lots is a major concern, considering the high risk of stealing motorcycles in campus parking areas [12], [13]. The frequent theft of vehicles raises concerns for vehicle owners, especially in public places such as campuses. Therefore, an effective solution is needed to overcome this problem, either through the addition of parking spaces or by improving the existing parking security system [14], [15].

Nowadays, public facilities, such as parking lots at the Faculty of Engineering, University of Mataram, have not used an automatic checking system. Therefore, riders are free to park their vehicles anywhere without knowing whether the parked vehicles belong to Mataram University residents or belong to outsiders who visit Mataram University. The unsupportive parking lot makes the owner of the vehicle worry about security and order in the parking area. Based on the observations that have been done, it is known that the available motorcycle parking area is 2389,72 m2. With a parking area of 2389,72 m2, the number of two-wheeled vehicles that can be accommodated is 1194 vehicles. These vehicles are dominated by the motorbikes whose owners are mostly students.

Several previous studies have designed vehicle parking information systems using the Waterfall method [13], [16], [17], [18], [19] which is a traditional system development method. The Waterfall method is a software development method that uses stages that must be completed sequentially, starting from requirements analysis, system design, implementation, testing, and maintenance [20]. Although this method has been widely used and has a clear structure, it is often less flexible in dealing with changing needs during the development process. However, not many have used the Personal Extreme Programming (PXP) method in designing vehicle parking

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

information systems. Meanwhile, PXP has a high level of flexibility that allows developers to return to each stage of development as needed [21], [22], [23]. The PXP method, which is an adaptation of the Extreme Programming (XP) methodology, emphasizes rapid iteration, intensive communication, and continuous adaptation towards change. This method is particularly suitable for projects that require rapid response to changes and adjustments based on user feedback [22]. In addition, this research provides a real-time detection element that has not been widely discussed in previous research. The designed system is able to detect the rider's face and vehicle license plate directly and match the data with the existing database. This real-time detection is used to ensure the identity of motorists entering and exiting the parking area, thus increasing the security and efficiency of the designed parking information system.

Based on the problems that have been described, the purpose of this research is to build a web-based motorcycle parking information system that will store riders' data in real-time. The development of this information system will implement an integrated model that is able to detect the face and police number of the rider through CCTV installed at the parking entrance. The Personal Extreme Programming (PXP) method will be used as the development method, which is a software development method designed for individual developers. PXP allows developers to work independently with high flexibility and continuous iteration, ensuring each stage of development can be personally customized and effective in meeting system needs [24].

2. RESEARCH METHODOLOGY

In this research, the method that will be used as a system development method is the Personal Extreme Programming (PXP) method. Personal Extreme Programming (PXP) is a software development methodology adapted for execution by individual software developers [21]. This method has a high degree of flexibility that allows developers to return to each stage of development as needed. In addition, this method supports rapid system development by reducing formal procedures that allow developers to focus on the result efficiently [23].

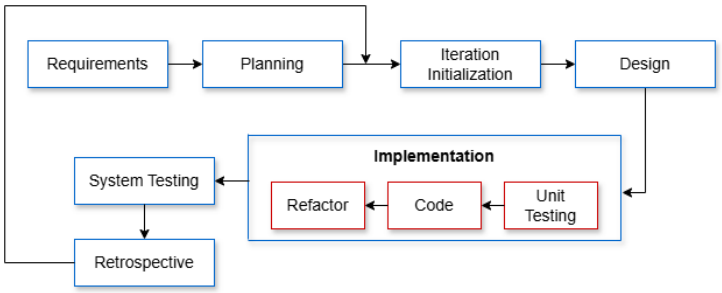

Figure 1. Stages of Personal Extreme Programming (PXP) Method

2.1 Requirements

Requirements is the initial stage in the Personal Extreme Programming method which is performed for thorough information gathering through interviews, observations, and document analysis to understand user needs for the system to be developed. By understanding these needs, developers can create clear and detailed user stories, which will be the basis for the next stages of development. User stories are a way of documenting system requirements in the agile development methodology [24].

The importance of the requirements stage in the PXP method lies not only in the collection of information, but also in the active involvement of users in the development process. This ensures that the system being developed truly fits the needs and can provide an effective solution to the problems that are faced. Thus, the requirements stage becomes a strong foundation to ensure the success of a PXP-based software development project.

2.2 Planning

The planning stage aims to determine the overall functionality to be developed in the system and design the development schedule. At this stage, the main features to be implemented and the prioritization of development based on user needs have been identified at the requirements stage [24]. At this stage, user stories are grouped based on the priority scale of working on features and estimated time in the form of story points. Story point is a method to determine the size of each feature in the form of user stories or the amount of work specifically described. Determination of story points is based on the level of urgency and complexity [21].

2.3 Iteration Initialization

Iteration initialization is a stage where predefined functionality is described in detail using the Unified Modeling Language (UML). At this stage, developers begin to detail each planned feature in the form of diagrams and models, which provide a visual representation of how the system will function as a whole. UML is a commonly

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

used tool in object-oriented system development, providing a visual modeling language that allows system developers to blueprint their vision in a standardized way. The advantage of UML is that it is easy to understand and comes with an effective mechanism for communicating system design between web developers, reducing the risk of misunderstandings and ensuring that all parties have a consistent understanding of the system architecture [22].

2.4 Design

At this design stage, the database design is made using the Entity Relationship Diagram (ERD), which functions to manage and organize the database of the information system to be created. ERD helps in mapping the relationship between entities and attributes, ensuring an efficient and easily accessible data structure [25]. The result of this design stage is a clear and detailed ERD, which describes the relationships between entities such as users, vehicles, and parking spaces, as well as the relevant attributes. This ERD becomes the main guide in the development of the database that supports the overall functionality of the system.

2.5 Implementation

The implementation stage involves coding the system using a collaboration of HTML markup language, PHP programming language, and Bootstrap framework. Bootstrap is a powerful and intuitive front-end framework using HTML, CSS, and Javascript for faster and easier web-based application development [26]. In addition, this web development uses Laravel as its backend framework. Laravel is a PHP framework known for its simplicity, reliability, and a variety of features that make it easy to develop web applications. Laravel provides various tools and libraries to simplify common tasks such as routing, sessions, caching, and authentication and allows developers to write clean and maintainable code [27]. The database will use MySQL Workbench. After coding is complete, the code is tested in unit testing. If there are errors, a correction or refactor is made at the problematic stage. If there are no errors, proceed to the next unit [23].

2.6 System Testing

The system testing stage involves the functionality testing of the system with the Black Box Testing method, where units or modules are executed and the results are observed. Black Box Testing is a software testing method in which web developers and clients test only in terms of functional specifications without looking at the design and program code [28]. Testing is done by creating all test cases for system functions that match the required specifications. The test cases are evaluated based on whether the results match the desired output [24].

2.7 Retrospective

In the retrospective stage, the system is evaluated, and decisions are made based on the iterations that have been carried out. If there are still errors, then improvements are made starting from the iteration initialization stage. This stage accommodates reflection on the development process that has been carried out, allowing the team to identify strengths, weaknesses, and opportunities for future improvement. This evaluation is important to ensure that the methods used are effective in the application development process, as well as ensuring that each iteration contributes to the overall success of the project [29].

3. RESULT AND DISCUSSION

3.1 Requirements

At the requirements stage, a discussion process was carried out with the Faculty of Engineering, Mataram University, to discuss the features that must be available in this information system. In addition, observations were made of the total parking space area at the Faculty of Engineering, which is 2389.72 m2, so that the maximum number of two-wheeled vehicles that can be accommodated is 1194 motorbikes. The conclusion of this discussion summarizes all the needs that must exist in a Web-based Motorcycle Parking Information System, which is then used as a user story. User stories in this system can be seen in Table 1.

Table 1. User Story

Code	Title	Description	Acceptance Criteria
US-01	Sign Up	As a new user, I would like to be able to sign up for a new account so that I can access the system.	Users can access the signup page from the main page, fill in the registration form with their name, username, email, gender, and password, and submit data to sign up. The system will verify the data and create a new account, and the user will receive a success notification after successfully signing up.
US-02	Sign In	As a registered user, I would like to be able to sign in to the system so that I	Users can access the sign-in page from the main page, enter a valid email or username and password, and the system will verify the credentials and allow

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

Code	Title	Description	Acceptance Criteria
		can manage my parking data.	access if they are valid. Users will be redirected to the dashboard upon successful sign in, and will get an error message if the email or password is incorrect.
US-03	Dashboard	As a user, I would like to be able to view a dashboard that provides a summary of the latest information regarding the data of motorists entering the parking area.	Users can access the dashboard page after successfully signing in. This dashboard displays important information related to the rider's data today.
US-04	View data of all admins	As a superadmin, I want to see a list of all admins so that I can manage admin accounts.	Users can access the admin list page after logging in, and the system displays a list of all admins with details such as name, username and role.
US-05	Add admin data	As a superadmin, I want to add new admin data so that the new admin can access the system.	Users can access the admin addition form from the admin list page, fill in the form with name, username, email, gender and password, and submit the form to save the new admin data. The new admin will appear in the admin list after being successfully added, and the user will get a success notification after successfully adding the admin.
US-06	Editing admin data	As a superadmin, I want to edit admin data so that admin information remains accurate and up-to-date.	Users can access the edit admin page from the admin list, change their name, username, email, gender or password, and submit a form to update admin data. The changes will appear in the admin list once saved, and the user will get a success notification after successfully editing the admin.
US-07	Deleting admin data	As a superadmin, I want to delete admin data so that I can manage admin access.	Users can access the delete option from the admin list, the system will ask for confirmation before deleting admin data. After confirmation, the system will delete the admin data and the deleted admin will no longer appear in the admin list. The user will get a success notification after successfully deleting the admin.
US-08	View admin data details	As a superadmin, I want to view admin details so that I can verify admin information.	Users can access admin details from the admin list, and the system displays complete admin information such as name, username, email, gender and role.
US-09	View data of all riders	As a superadmin and admin, I would like to see a list of all riders so that I can manage the rider data.	Users can access the list of riders page after signing in, and the system displays a list of all riders with details such as NIM, name, department and vehicle number.
US-10	Add rider data	As a superadmin, I want to add a new rider so that the new rider can use the parking service.	Users can access the rider addition form from the rider list page, fill in the form with NIM, name, username, department, gender, and vehicle number, and submit the form to save the new rider data. The new rider will appear in the rider list after being successfully added, and the user will get a success notification after successfully adding the rider.
US-11	Edit rider data	As a superadmin, I would like to edit rider data so that rider information remains accurate and up-to-date.	Users can access the edit rider page from the rider list, change NIM, name, username, department, gender, and vehicle number, and submit the form to update rider data. The changes will appear in the rider list once they have been saved, and the user will get a success notification after successfully editing the rider.
US-12	Delete rider data	As a superadmin, I want to delete rider data so that I can manage rider access.	Users can access the delete option from the rider list, the system will ask for confirmation before deleting the rider data. After confirmation, the system will delete the rider data and the deleted rider will no

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

Code	Title	Description	Acceptance Criteria
			longer appear in the rider list. The user will get a success notification after successfully deleting the rider.
US-13	View rider details	As a superadmin and admin, I want to see the details of the rider's data so that I can verify the rider's information.	Users can access the rider details from the rider list, and the system displays the rider's full information such as NIM, name, username, department, gender, and vehicle number.
US-14	Viewing rider statistics	As a superadmin or admin, I would like to be able to view parking usage statistics for riders so that I can understand trends and usage patterns of parking facilities.	Users can view which includes information on the number of riders registered in each course along with data related to their gender. In addition, the statistics also present information on available and used parking capacity, providing a complete picture of the efficient use of parking spaces in the Faculty of Engineering.
US-15	View rider details	As a superadmin and admin, I want to see the details of the rider's data so that I can verify the rider's information.	Users can access the rider details from the rider list, and the system displays the rider's full information such as NIM, name, username, department, gender, and vehicle number.
US-16	Add a vehicle data	As a superadmin, I want to add a new vehicle so that the new vehicle can be recorded in the parking system.	Users can access the vehicle addition form from the vehicle list page, fill in the form with the vehicle number, select the NIM owner from the rider list, and submit the form to save the new vehicle data. The new vehicle will appear in the vehicle list after it is successfully added, and the user will get a success notification after successfully adding the vehicle.
US-17	Edit vehicle data	As a superadmin, I want to edit vehicle data so that vehicle information remains accurate and up-to-date.	Users can access the edit vehicle page from the vehicle list, change the vehicle number and NIM, and submit a form to update the vehicle data. The changes will appear in the vehicle list once successfully saved, and the user will get a success notification after successfully editing the vehicle.
US-18	Delete vehicle data	As a superadmin, I want to delete vehicle data so that I can manage existing vehicle data.	Users can access the delete option from the vehicle list, the system will ask for confirmation before deleting the vehicle data. After confirmation, the system will delete the vehicle data and the deleted vehicle will no longer appear in the vehicle list. The user will get a success notification after successfully deleting the vehicle.
US-19	View vehicle data details	As a superadmin and admin, I would like to view vehicle data details so that I can verify vehicle information.	Users can access vehicle details from the vehicle list, and the system displays complete vehicle information such as vehicle number, owner NIM, name, and department.
US-20	View Parking Space Information	As a user, I want to be able to view information about parking spaces to find out the remaining available parking spaces.	As a user, I want to be able to easily view information about parking spaces at the Faculty of Engineering, Mataram University. This helps me to know the total number of parking spaces, parking spaces that are still available, and those that have been used after successfully logging into the system.
US-21	Sign Out	As a system user, I would like to be able to sign out of my account so that I can secure my account when I am done using the system.	Users can sign out of their account by clicking the ”Sign Out” button available on each page. After clicking the button, the system will disconnect the user's session and redirect them to the Sign In page. After signing out, users will not be able to access pages that require authentication without signing in again.

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

This research is an advanced project [30] that integrates two machine learning models, namely face detection and license plate detection models. In this web, these models are used to record and match the face and license plate of the vehicle when entering the parking area with the face and license plate of the vehicle when the rider exits the parking area. The engines used in this web are the result of other sub-researches, which are part of a large integrated research. The models used include TensorFlow for face detection, and YOLO (You Only Look Once) for license plate detection. Both models are operated simultaneously in this system to ensure accurate and efficient identification of riders and their vehicles, thus supporting user safety and convenience in parking lots. In this system, the detection is used to record the motorists who enter and exit the parking area on US-03. Then, the data will automatically update the information about the remaining parking spaces on US-20.

3.2 Planning

The planning stage involves designing the features that will be included in the system, based on the analysis of user needs from the previous stage. Some of the features to be created include a registration feature for admin users, a login feature that can be accessed by admin and superadmin users, a dashboard menu, a rider data menu, vehicle data, admin data that can perform CRUD, and statistical data from registered riders. At this stage, user stories are grouped based on the priority scale of working on features and estimated time in the form of story points, which are determined based on the level of urgency and complexity with a scale of "Low", "Medium", and "High". In addition, iteration initialization is carried out in the system development process which is divided into 7 iterations. Details of each iteration can be seen in Table 2.

Table 2. Story Point

Code	Title	Urgency	Complexity	Story Point
	Iteration 1			
US-01	Sign Up	High	Medium	3
US-02	Sign In	High	Medium	3
US-21	Sign Out	Medium	Low	2
	Total Points			8
	Iteration 2			
US-03	Dashboard	High	Medium	3
US-04	View data of all admins	Medium	Medium	3
US-05	Add admin data	Medium	Medium	3
US-06	Edit admin data	Medium	Medium	3
	Total Points			12
	Iteration 3			
US-07	Delete admin data	Medium	Medium	3
US-08	View admin data details	Medium	Low	2
US-09	View data of all riders	Medium	Medium	3
	Total Points			8
	Iteration 4			
US-10	Add rider data	Medium	Medium	3
US-11	Edit rider data	Medium	Medium	3
US-12	Delete rider data	Medium	Medium	3
	Total Points			9
	Iteration 5			
US-13	View rider details	Medium	Low	2
US-14	View rider statistics	Medium	High	5
US-15	View rider details	Medium	Medium	3
	Total Points			10
	Iteration 6			
US-16	Add a vehicle data	Medium	Medium	3
US-17	Edit vehicle data	Medium	Medium	3
US-18	Delete vehicle data	Medium	Medium	3
	Total Points			9
	Iteration 7			
US-19	View vehicle data details	Medium	Low	2
US-20	View Parking Space Information	High	High	5
	Total Points			7
	Total			**63**

3.3 Iteration Initialization

In the third stage of the Personal Extreme Programming (PXP) development method, namely Iteration Initialization, use case diagrams are made based on user stories that have been identified at the requirements stage and story points that have been determined at the planning stage. This use case diagram will consist of three actors,

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

namely superadmin, admin, and rider. This diagram helps visualize the interaction between users and the system and describes the main functionality to be developed. Use Case Diagram of Web-Based Motorcycle Parking Information System can be seen in Figure 2.

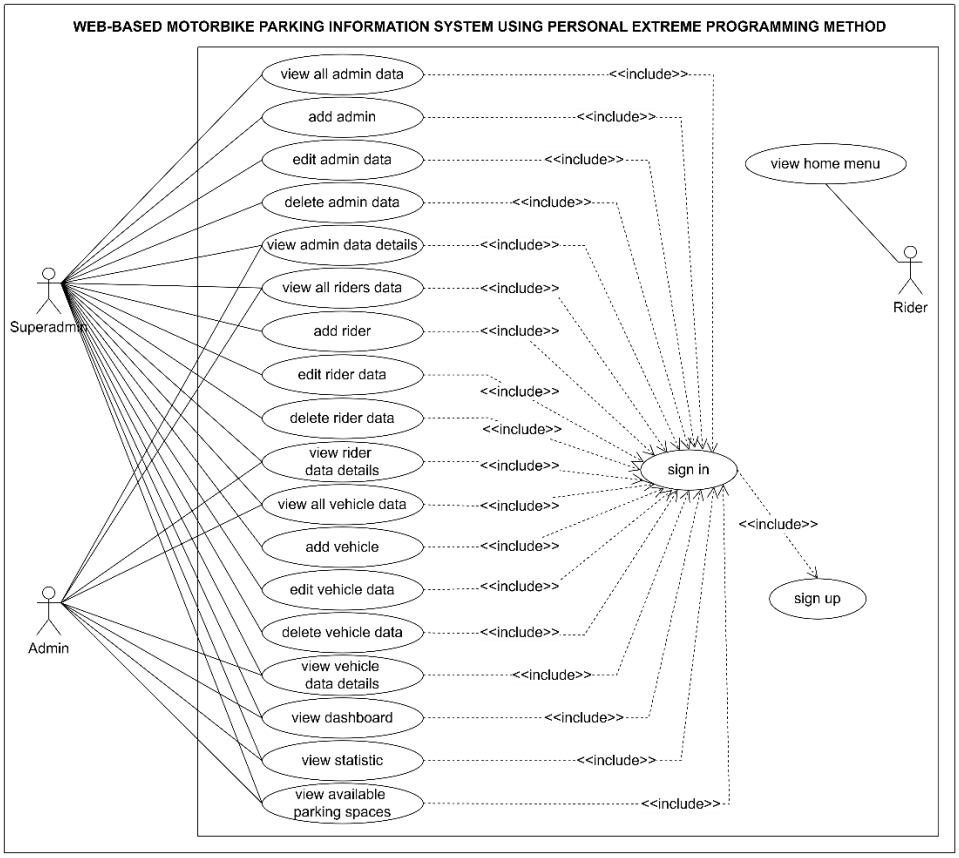

Figure 2. Use Case Diagram of Web-based Motorcycle Parking Information System

3.4 Design

In the design stage, a database system display design is carried out using the Entity Relationship Diagram (ERD) which will be used in accordance with the required features. The database of this Web-Based Motorcycle Parking Information System consists of 5 entities consisting of users, riders, motorcycle, visit, and parking area. Details of the ERD that has been made are in Figure 3. below.

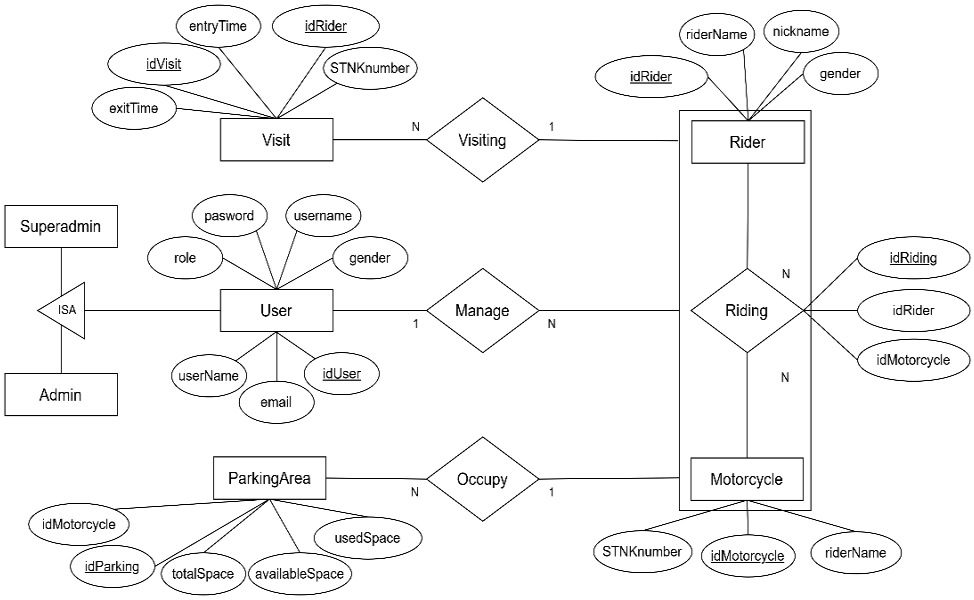

Figure 3. ERD of Web-based Motorcycle Parking Information System

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

3.5 Implementation

The implementation stage includes three processes: writing program code, unit testing, and refactoring. The technology used to create this system is the Laravel framework with the PHP programming language, HTML, CSS, JavaScript markup languages with the Bootstrap framework, and MySQL Workbench as a web server.

The process of writing program code involves creating a system with features that have been described at the requirements stage. If a problem is found in the features or units that have been done, a refactor process is carried out to correct the errors found in the unit testing. This implementation involves writing program code for each user story that has been created previously, with three stages, namely creating unit tests, writing code, and refactoring. Here are some views of the system page.

3.5.1 Sign In Page

This Sign In page can only be accessed by users who have the roles of “Superadmin” and “Admin”. On the sign in page, users are asked to enter the email or username that has been registered previously along with the password. If the login process is successful, the user will be directed to the dashboard menu. However, if the login process fails, the user will be given a message stating that the login was unsuccessful.

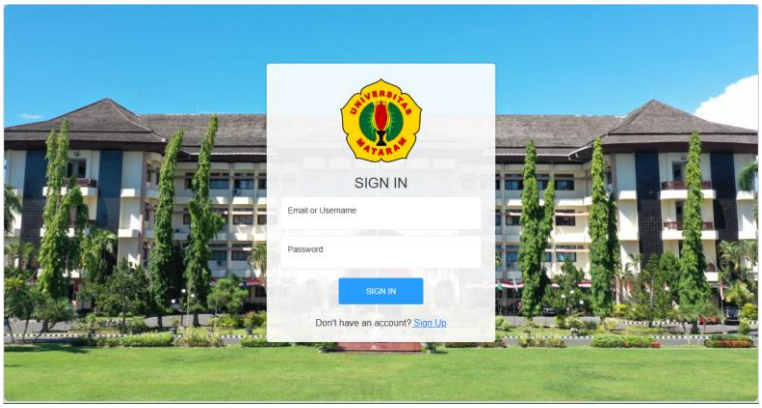

Figure 4. Sign In Page

3.5.2 Dashboard Page

This dashboard page displays data on motorists entering the parking area of the Faculty of Engineering, University of Mataram on this day. The information presented includes the time of entry, name, NIM, departement, as well as the motorcycle plate number of the rider. This page is designed to be accessible by superadmins and admins, allowing them to monitor and manage rider data more efficiently. The main purpose of this page is to ensure safety and order in the Faculty of Engineering parking area, as well as facilitate the process of monitoring vehicles entering and exiting.

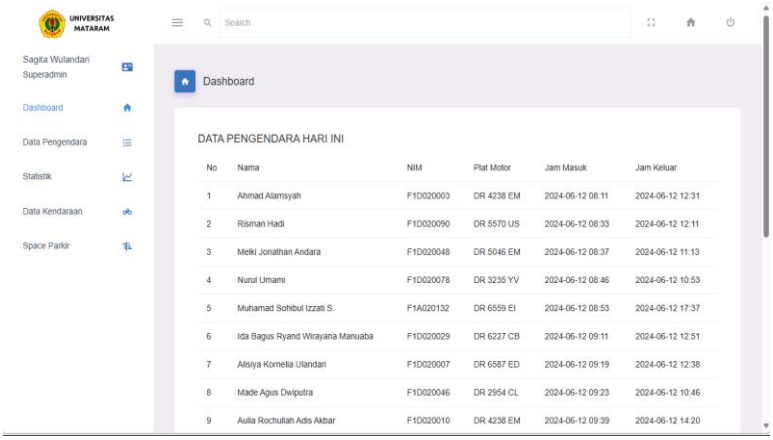

Figure 5. Dashboard Page

3.5.3 Admin Page

On the admin data menu, there is detailed information about all registered admins. The admin data display will differ depending on the user's role, superadmin and admin. Superadmins have full access to all available features, including the ability to add, edit, delete, and view other admin details. On the other hand, admins have limited access to only the features to view their own data details. This difference distinguishes the role of an admin and superadmin in the system.

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

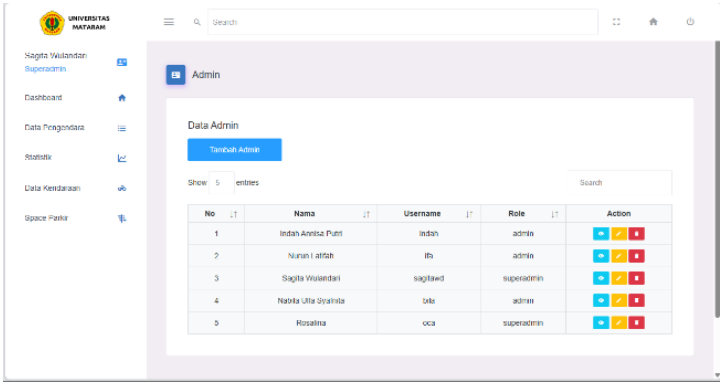

Figure 6. Admin Page

3.5.4 Rider Page

On the rider data menu, there are data details of all registered riders. The rider data display is different for superadmin and admin. Superadmins have full access to available features, such as add, edit, delete, and show rider details. However, admins have limited access only to show rider details. This difference marks the role of an admin and superadmin.

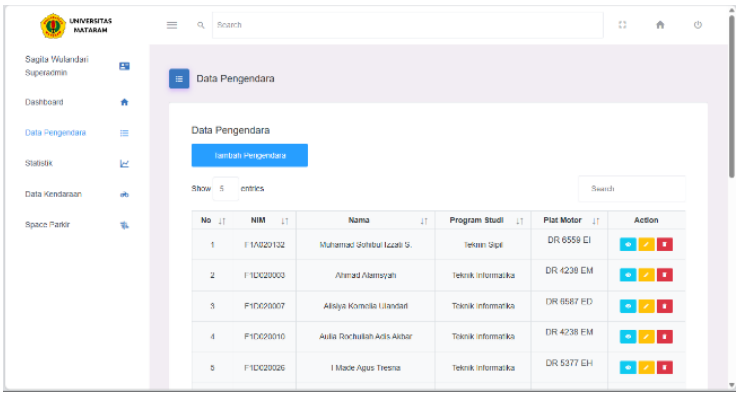

Figure 7. Rider Page

3.5.5 Statistic Page

This page displays rider statistics in graphical form, which includes rider distribution by major and gender. In addition, there is also a graph depicting the usage of parking spaces in the parking area. With this visualization, admins and superadmins can easily analyze parking usage patterns and rider characteristics, supporting better decision-making in parking facility management.

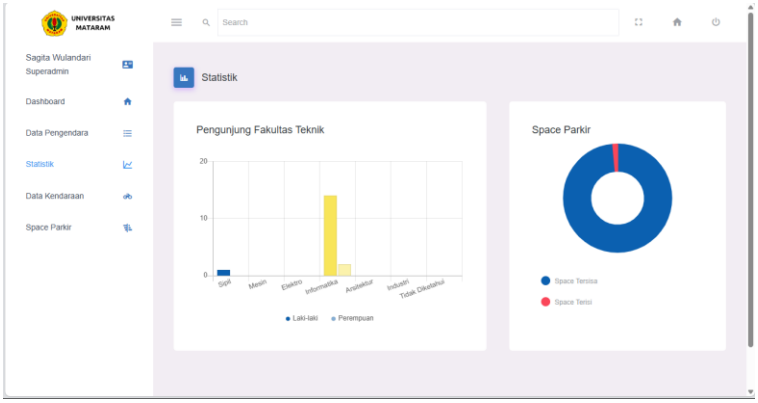

Figure 8. Statistic Page

3.5.6 Vehicle Page

On the vehicle data menu, there are data details of all registered riders. The rider data display is different for superadmin and admin. Superadmins have full access to available features, such as add, edit, delete, and show details of registered vehicles. However, admins have limited access only to show vehicle data details. This difference marks the role of an admin and superadmin.

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

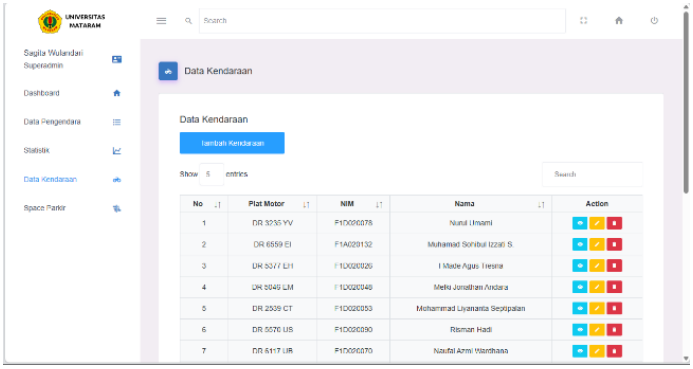

Figure 9. Vehicle Page

3.5.7 Parking Space Page

This page displays information regarding the total number of available parking spaces. This information is automatically updated based on incoming and outgoing riders. When a rider enters, the number of parking spaces will decrease by 1, and when a rider exits, the number will increase by 1. With this feature, admins and superadmins can monitor the availability of parking spaces in real-time, ensuring efficiency and effectiveness in parking area management.

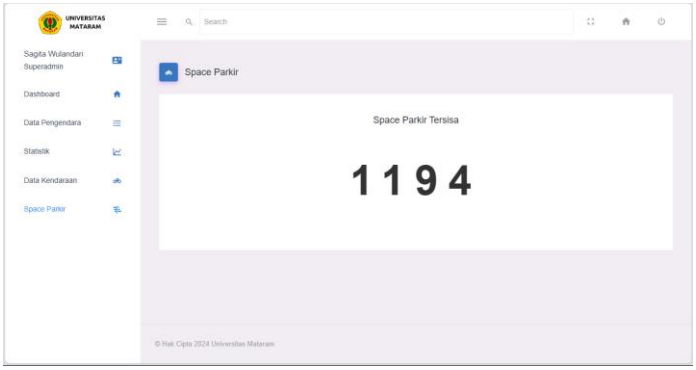

Figure 10. Parking Space Page

3.6 System Testing

At this stage, system functionality testing is carried out using the Black Box Testing method, which allows testing based on functional specifications without paying attention to internal design and program code. The testing process involves executing units or modules from various pages in the system and comparing the results obtained with the expected results to ensure conformity with the desired business process. If the test results are as expected, then the unit or module is considered to have successfully fulfilled the specified functionality. The results of system functionality testing can be seen in Table 3. below.

Table 3. Testing Results with Black Box

Tested Page	Process Design	Expected Results	Test Results
Sign Up	User fills in the registration form and submits it	New account is created and user receives a success notification	Succeed
Sign In	User enters email or username and password to login	User successfully logs in and is redirected to the dashboard	Succeed
Dashboard	Users who successfully login are redirected to the dashboard main page	The dashboard displays a summary of data information on motorists entering the parking area	Succeed
View data of all admins	The user accesses the admin list page	A list of all admins is displayed with details such as name, username, email, gender, and role	Succeed
Add admin data	The user fills in the admin addition form and submits it	The new admin data is saved and displayed in the admin list	Succeed
Editing admin data	The user accesses the edit form from the admin list and edits the data	The admin data is updated and the changes are displayed in the admin list	Succeed

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

Tested Page	Process Design	Expected Results	Test Results
Deleting admin data	User selects an admin from the list and deletes it	Admin data is deleted from the admin list	Succeed
View admin data details	User clicks the show action in the list to view the details	Admin details are displayed	Succeed
View data of all riders	The user accesses the statistics page	The statistics page displays visual data in the form of a graph of the number of riders by gender and study program and the remaining parking spaces	Succeed
Add rider data	The user accesses the riders list page	A list of all riders is displayed with details such as NIM, name, study program, and police number	Succeed
Edit rider data	User fills in the rider addition form and submits it	New rider data is saved and displayed in the rider list	Succeed
Delete rider data	The user accesses the edit form of the rider list and edits the data	The rider data is updated and the changes are displayed in the rider list	Succeed
View rider details	User selects a rider from the list and deletes it	Rider data is deleted from the rider list	Succeed
Viewing rider statistics	User clicks the show action in the list to view details	Details of the rider are displayed	Succeed
View rider details	User accesses the vehicle list page	A list of all vehicles is displayed with details such as NIM, police number, as well as owner name	Succeed
Add a vehicle data	User fills in the vehicle addition form and submits it	New vehicle data is saved and displayed in the vehicle list	Succeed
Edit vehicle data	User accesses the edit form of the vehicle list and edits the data	Vehicle data is updated and changes are displayed in the vehicle list	Succeed
Delete vehicle data	User selects a vehicle from the list and deletes it	Vehicle data is deleted from the vehicle list	Succeed
View vehicle data details	User clicks the show vehicle action in the list to view the details	Vehicle details are displayed	Succeed
View Parking Space Information	User accesses the parking space information page	User can view real-time information regarding parking space availability, including the number of available slots and status (empty or occupied)	Succeed
Sign Out	User selects the sign out option from the navigation menu	User successfully exits the system and is redirected to the login page	Succeed

Based on the results from Table 3, the results show that all the main features of the web-based motorcycle parking information system function according to the specified specifications. Of the 7 test iterations conducted, all of them run well without any failures. Therefore, the percentage of test success is 100%. This shows that the system has successfully fulfilled all functional requirements with a high level of accuracy and efficiency. Thus, the system is reliable for use in a campus environment, with the potential to be applied in other parking areas.

3.7 Retrospective

In the retrospective stage, discussions and analysis of the iteration result data with stakeholders are carried out to find out whether there are still problems in the system. If there are still problems or discrepancies with the wishes of stakeholders, then rework will be carried out from the iteration initialization stage to retrospective. This stage also includes an analysis of the system development process, especially the estimated development time that has been determined through story points at the planning stage.

4. CONCLUSION

This research implements a Web-based Motorcycle Parking Information System at the Faculty of Engineering, Mataram University using the Personal Extreme Programming (PXP) method which is known for its flexibility in accommodating changing needs. This method consists of several stages including Requirements, Planning, Iteration Initialization, Design, Implementation, System Testing, and Retrospective. This system is designed to solve the problem of security and management of parking lots on campus in the absence of a previous automatic checking system. Features developed include user management with logins for admin and superadmin, rider data

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

management, vehicle data, and available parking space information. CCTV technology is integrated to detect the faces and license plates of motorcyclists in real time, the results of which are stored and matched with existing databases. This detection process combines two machine learning models, namely face detection and vehicle license plate detection. This research adopts the Black Box Testing method to validate the performance of the system, ensuring all features function according to the set specifications with accurate and efficient results. Overall, the implementation of this system has successfully improved the security and efficiency of parking management in the campus environment, with the potential to be widely applied in educational institutions and other public parking areas, through continuous development and improvement. This web-based parking information system not only provides an effective solution for parking management on campus but also increases the level of security with the use of real-time face and license plate detection technology. It is shown that this system can be a useful model for similar applications in various other environments.

REFERENCES

[1] Acuviarta and A. M. P. Permana, "Analisis Faktor Yang Mempengaruhi Permintaan Sepeda Motor di Kota-Kota Besar Jawa Barat," Jurnal Riset Ilmu Ekonomi, vol. 2, no. 3, pp. 171–180, Jan. 2022, doi: 10.23969/jrie.v2i3.41.
[2] S. R. K. Irawan, E. C. Putri, D. Situngkir, F. Yusvita, and M. D. R. Rusdy, "Factor Related to Two-Wheeled Safety Riding Behavior at Production Planning Workers at PT. Yuasa Battery Indonesia in 2021," Journal of Vocational Health Studies, vol. 6, no. 2, pp. 107–117, Nov. 2022, doi: 10.20473/jvhs.v6.i2.2022.107-117.
[3] R. Ananda and M. Amin, "Use of KTP to Activates Start Motorcycle Engine with Module RC-522," JURTEKSI (Jurnal Teknologi dan Sistem Informasi), vol. 9, no. 3, pp. 515–520, Jun. 2023, doi: 10.33330/jurteksi.v9i3.2495.
[4] F. Apri Wenando, R. Pratama Santi, J. Rahmadoni, L. Nur Irsyad, and S. Ramadhani Putri, "Sistem Informasi Parkir Elektronik pada Kampus Universitas Andalas Berbasis Website," Jurnal Teknologi Informasi dan Ilmu Komputer (FASILKOM), vol. 13, no. 1, pp. 61–71, Apr. 2023.
[5] Badan Pusat Statistik, "Jumlah Kendaraan Bermotor Menurut Provinsi dan Jenis Kendaraan (unit), 2022," Badan Pusat Statistik. Accessed: May 27, 2024. [Online]. Available: https://www.bps.go.id/id/statistics-table/3/VjJ3NGRGa3dkRk5MTlU1bVNFOTVVbmQyVURSTVFUMDkjMw==/jumlah-kendaraan-bermotor-menurut-provinsi-dan-jenis-kendaraan-unit-.html?year=2022
[6] Badan Pusat Statistik, "Jumlah Kendaraan Bermotor Menurut Provinsi dan Jenis Kendaraan (unit), 2021," Badan Pusat Statistik. Accessed: May 27, 2024. [Online]. Available: https://www.bps.go.id/id/statistics-table/3/VjJ3NGRGa3dkRk5MTlU1bVNFOTVVbmQyVURSTVFUMDkjMw==/jumlah-kendaraan-bermotor-menurut-provinsi-dan-jenis-kendaraan-unit-.html?year=2021
[7] A. A. M. Suradi, Mushaf, M. F. Rasyid, A. R. Johanis M, and A. Bahtiar, "Penerapan Teknologi QR Code Pada Sistem Informasi Parkir Berbasis Android," Jurnal sistem Informasi dan Teknologi Informasi, vol. 12, no. 1, pp. 100–110, Apr. 2023.
[8] Januriad, B. S. Widhi Dharma, and Y. Utami, "Evaluasi Kebutuhan Ruang Parkir Sepeda Motor di Area GEDUNG FKIP dan FST Universitas Qomarul Huda Badaruddin Bagu," Al-Qlu: Jurnal Matematika, Teknik dan Sains, vol. 1, no. 1, pp. 1–13, Jan. 2023, [Online]. Available: https://jurnal.yalamqa.com/index.php/aqlu
[9] A. F. Baharuddin, M. Y. Jinca, and J. Rahim, "Pengembangan Penataan Perparkiran Universitas Hasanuddin Di Fakultas Teknik, Fakultas MIPA Dan Fakultas Farmasi," Jurnal Ilmiah Ecosystem, vol. 21, no. 3, pp. 625–633, Dec. 2021, doi: 10.35965/eco.v21i3.1256.
[10] R. Oktaviani, Nazwirman, Djamaludin, and V. Septiana Windyasari, "Aplikasi Sistem Parkir Kendaraan Bermotor Menggunakan Teknologi Radio Frequency Identification (RFID) di Universitas Islam Syekh Yusuf Tangerang," Jul. 2020.
[11] F. Dwi Amanda, Masril, and S. Dewi, "Analisis Karakteristik Parkir Kendaraan Bermotor di Kawasan Plaza Ramayana Kota Bukittinggi," Ensiklopedia Research and Community Service Review, vol. 1, no. 1, Oct. 2021, [Online]. Available: http://jurnal.ensiklopediaku.org
[12] B. Ballen Saputra, A. Budianto, and R. Riswadi, "The Role Of The Polri Criminal Reserve Unit In Revealing Cases Of Motorcycle Theft (Case Study Of Polres Semarang)," Journal Research of Social Science, Economics, and Management, vol. 2, no. 10, pp. 2371–2378, May 2023, doi: 10.59141/jrssem.v2i10.455.
[13] E. Rahmawati, "Rancang Bangun Sistem Informasi Parkir Sepeda Motor Berbasis Dekstop," Journal Computer Science, vol. 1, no. 2, 2022.
[14] S. R. Siregar, Azlan, and Pristiwanto, "Aplikasi Sistem Informasi Parkir Kampus Menggunakan Model Single Channel Single Phase Service," Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer), vol. 23, no. 2, pp. 171–180, Feb. 2024, [Online]. Available: https://ojs.trigunadharma.ac.id/index.php/jis/index
[15] H. Marcos and M. Reza, "Implementasi IoT Pada Rancang Bangun Aplikasi Mobile Sistem Keamanan Dan Pelacak Sepeda Motor," Jurnal Teknik Informatika dan Sistem Informasi, vol. 8, no. 1, pp. 170–180, Mar. 2021, [Online]. Available: http://jurnal.mdp.ac.id
[16] E. Anjarwani, H. I. Illina, and N. Agitha, "Sistem Informasi Daya Tampung Area Parkir Pada Pusat Perbelanjaan Untuk Meningkatkan Layanan Penggunaan Parkir (Studi Kasus: Lombok Epicentrum Mall)," J-COSINE (Journal of Computer Science and Informatics Engineering), vol. 6, no. 1, pp. 1–9, Jun. 2022, [Online]. Available: http://jcosine.if.unram.ac.id/
[17] M. Irfan Maulana, R. Ramdhan Priatna, and M. Yusuf, "Rancang Bangun Aplikasi Parkir Pada PT. Indostorage Solusi Teknologi Berbasis Web Menggunakan Metode Waterfall," 2023. [Online]. Available: https://jurnalmahasiswa.com/index.php/jriin
[18] A. Prayitno, B. Syachputra, and I. Arrela, "Perancangan Sistem Informasi Parkir Di Universitas Dinamika Bangsa Berbasis Web," Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM) , vol. 3, no. 2, pp. 667–674, Sep. 2023, doi: 10.33998/jakakom.v3i2.

Journal of Information System Research (JOSH)
Volume 5, No. 4, Juli 2024, pp 1029−1041
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v5i4.5508

[19] G. Salim and M. Feizal, "Rancang Bangun Sistem Informasi Parkir Menggunakan Laravel 8 Metode Waterfall di Institut Teknologi dan Bisnis Swadharma," OKTAL : Jurnal Ilmu Komputer dan Science, vol. 2, no. 1, pp. 271–276, Jan. 2023.
[20] L. Afuan, N. Hidayat, B. Wijayanto, and I. Cahya Febriani, "Website Employee Acceptance Information System at PT Noto Teknologi Indonesia," Jurnal Teknik Informatika (Jutif), vol. 3, no. 5, pp. 1379–1388, Oct. 2022, doi: 10.20884/1.jutif.2022.3.5.524.
[21] F. A. Melati, W. R. Pradana, and N. N. Arisa, "Pengembangan Buku Tamu Digital Pada PT XYZ Menggunakan Metode Personal Extreme Programming," Jurnal Sistim Informasi dan Teknologi, vol. 5, no. 4, pp. 73–81, Jan. 2024, doi: 10.60083/jsisfotek.v5i4.323.
[22] M. F. Riki and I. Effendy, "Aplikasi Pencarian Donor Darah Menggunakan Metode Personal Extreme Programming Berbasis Android," SMATIKA : STIKI Informatika Jurnal, vol. 12, no. 02, pp. 186–199, Dec. 2022, doi: 10.32664/smatika.v12i02.698.
[23] D. Saeful Malik and A. Zein, "Menggunakan Metode Personal Extreme Programming di Toko Surya Gemilang," Jurnal Ilmu Komputer JIK, vol. 5, no. 1, May 2022.
[24] A. Maulana Fikri and I. S. Putu Deny Arthawan P, "Rancang Bangun Sistem Informasi Buku Tamu Pada Dinas Pemuda, Olahraga dan Pariwisata Kota Balikpapan dengan Metode Persona Extreme Programming," Multitek Indonesia: Jurnal Ilmiah, vol. 14, no. 2, pp. 101–100, Dec. 2020, [Online]. Available: http://journal.umpo.ac.id/index.php/multitek
[25] F. Darmawan et al., "Rancang Bangun Sistem Informasi Pengelolaan Surat Menggunakan Metode Personal Extreme Programming (Studi Kasus: Kelurahan Sepinggan)," SPECTA Journal of Technology, vol. 6, no. 1, 2022, doi: 10.35718/specta.v6i1.
[26] Y. Anis, P. Purwatiningtyas, R. Retnowati, and E. A. N. Fajrina, "Penerapan Framework Bootstrap Dalam Sistem Informasi Rekam Medis Data Posyandu dengan Metode Waterfall," Jurnal Sistem Komputer dan Informatika (JSON), vol. 4, no. 2, p. 310, Dec. 2022, doi: 10.30865/json.v4i2.4833.
[27] S. Hendrawan, A. David Manuputty, B. Haryanto, and A. D. Manuputty, "Design of Information Systems for Research Permit Application with Agile Method and Website Based Laravel Framework," Journal of Information Systems and Informatics, vol. 2, no. 1, pp. 2656–5935, 2020, [Online]. Available: http://journal-isi.org/index.php/isihttp://journal-isi.org/index.php/isi
[28] N. Chasanah, A. I. Anggraeni, W. Ramadhanti, A. Krisnaresanti, L. R. Naufalin, and A. Dinanti, "Implementasi of Website Based E-Marketing Strategy in Organic Coconut Sugar Business Group," Jurnal Teknik Informatika (Jutif), vol. 3, no. 6, pp. 1747–1756, Dec. 2022, doi: 10.20884/1.jutif.2022.3.6.616.
[29] T. Palyus Fiqar, N. F. Azhar, and F. Azzahra, "Implementasi Metode Personal Extreme Programming dalam Pengembangan Sistem Informasi Manajemen Hak Paten pada Sentra HKI ITK," SPECTA Journal of Technology, vol. 6, no. 3, pp. 329–341, Jan. 2023, doi: 10.35718/specta.v6i3.799.
[30] M. L. Septipalan, I. B. K. Widiartha, A. Zubaidi, and M. Taufik, "Integrated Notification System for Smart Parking Security Using Bot Telegram," Jurnal Penelitian Pendidikan IPA, vol. 10, no. 5, pp. 2679–2686, May 2024, doi: 10.29303/jppipa.v10i5.7447.