Sistemasi: Jurnal Sistem Informasi
Volume 14, Nomor 2, 2025: 917-931
ISSN:2302-8149
e-ISSN:2540-9719

Development of Web-App Using Agile Scrum Method at PT. Stechoq Robotika Indonesia

[1]Sadr Lufti Mufreni*, [2]Ahmad Mushawwir
[1,2]Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Aisyiyah Yogyakarta
[1,2]Jl. Siliwangi (Ring Road Barat) No. 63, Indonesia
*e-mail: sadr@unisayogya.ac.id

(***received***: 16 February 2025, ***revised***: 24 February 2025, ***accepted***: 25 February 2025)

Abstrak

The increasing number of Jaminan Kecelakaan Kerja (JKK) claims over the past five years underscores the urgency of enhancing workplace safety measures. One possible solution is the adoption of Automated Guided Vehicles (AGVs) to streamline warehouse operations and minimize the risks associated with manual handling. In this study, the AGV is owned by the stakeholder, PT. Stechoq Robotika Indonesia (STECHOQ), which is responsible for AGV customization and the development of its control system through a web-based application. The web-app was developed using ReactJS, TypeScript, and Tailwind CSS, adopting the Agile Scrum methodology The web-app development followed a two-sprint approach, with each sprint lasting one week. A total of six key features were implemented Login, Dashboard, Station Management, AGV Management, Task Management, and Robot Control. Iterative Black Box Testing was conducted on these six features throughout both sprints, confirming their successful operation without any issues. Additionally, this study modified the Agile Scrum methodology by merging the sprint retrospective with the sprint review phase, enhancing efficiency while aligning with the existing workflow. The objective of this study is to develop a web-based application capable of controlling the stakeholder's AGV.

Keywords: agile scrum, AGV, reactjs, rest api, web application

1 Introduction

Based on data from BPJS Ketenagakerjaan, the number of Jaminan Kecelakaan Kerja (JKK) claims has continued to increase in recent years. In 2019, the number of JKK claims was recorded at 182,835 cases, which then increased to 221,740 claims in 2020, 234,370 claims in 2021, and 297,725 claims in 2022. This trend continued in 2023, where from January to November, the number of JKK claims had reached 360,635 cases [1]. The increasing number of claims underscores the challenges in ensuring a safe work environment. The high number of workplace accidents not only negatively impacts workers' well-being but also affects company productivity. Research shows that improving workplace safety contributes to increased employee productivity and work efficiency [2], [3]. By lowering accident risks, employees are able to work more effectively, minimizing operational disruptions.

The application of technology can be a key element in supporting workplace safety, particularly through the implementation of the Industry 5.0 concept. Industry 5.0 refers to a new era of industrialization where humans collaborate with technology and machines [4]. Thus, human-machine collaboration in Industry 5.0 opens new opportunities to improve workplace safety, such as through the implementation of robotics technology across various industries. The adoption of robotics technology in industries not only enhances productivity and flexibility but also improves worker safety, health, and well-being [4], [5].

One type of robot used in industries is the Automated Guided Vehicle (AGV). Automated Guided Vehicles (AGVs) are mobile robots used in industries to transport goods from one location to another [6]. AGVs are employed in modern warehousing and production environments to enhance greater efficiency, flexibility, and reduce operational costs, ultimately supporting increased productivity and the systematization of material flow [7]. Nevertheless, AGV management requires a reliable control system to maximize operational efficiency and minimize material handling costs [8].

ISSN:2302-8149
e-ISSN:2540-9719

In this study, the AGV robot is implemented in an industrial setting, specifically within the stakeholder environment of PT. Stechoq Robotika Indonesia, the research partner.

PT. Stechoq Robotika Indonesia (STECHOQ) is a company specializing in robotics, playing a role in customizing stakeholder-owned AGVs and developing their control systems through web-based applications. The web-based application developed by STECHOQ not only enables AGV control by operators but also serves as a platform that facilitates users in obtaining accurate and relevant information [9]. While the AGVs used belong to stakeholders, STECHOQ customizes both the hardware and software procurement aspects. Currently, the process of moving goods in stakeholder warehouses still relies on human labor or equipment such as forklifts. With the implementation of customized AGVs controllable via a web-based application, it is expected that operational efficiency in stakeholder warehouses will be improved.

The web-based application for robot control is built using various technologies, including the ReactJS library, TypeScript, and Tailwind CSS. ReactJS is chosen because it offers a virtual Document Object Model (DOM) and server-side rendering [10], which help speed up the rendering process and can impact the user experience. The web-based application development adopts the Agile Scrum method. The Agile Scrum framework enables faster, more efficient, and flexible implementation because its approach prioritizes iteration or rapid delivery, aligning with the evolving software functionality [11]. The web-based application is tested using the Black Box Testing method, which focuses on functionality [12], aiming to assess the user experience in meeting expected performance outcomes

2 Literature Review

A study conducted by [13] foucses on the development of a web-based application for employee attendance at Glints is built using React.js for the frontend, while the backend utilizes NodeJS and ExpressJS. This project is part of the Industrial Project Exploration (IPE), which is a component of the Merdeka Belajar Kampus Merdeka (MBKM). In this study, two sprints were conducted, each lasting two weeks. Testing results from the first sprint revealed a failure in the account registration feature, which was subsequently fixed in the second sprint. The implemented improvements proved successful, ensuring that all testing activities in the second sprint were completed without issues. The research demonstrated that the applied Agile Scrum methodology successfully enabled the development of new features during the application development process without disrupting the core features being developed.

Subsequent research by [14] explores the development of a web-based online survey platform using the Agile Scrum methodology. This study resulted in several User Interface (UI) components that were developed iteratively. The project emphasizes the implementation of React.js as the primary framework, enabling continuous iteration and improvement throughout the development process. indicated that applying Agile Scrum in frontend development using ReactJS can improve development team efficiency, enhance website aesthetics, and ensure periodic information updates.

While the previous studies focus on using ReactJS for various web applications, another study by [15] as a startup in research, Ubaform requires the development of a website that offers users an interactive experience, fast rendering for building document management, quiz, and online survey features, as well as a modern user interface. The development of the web application utilizes various packages provided by ReactJS, including react-router-dom, Material UI, zxcvbn, react-multi-carousel, Syncfusion, and react-icons. These packages are implemented on the Home Page, Login, Register, Dashboard Page, Builder Page, and Play Page. In this research, found that using ReactJS in developing the frontend for the Ubaform startup's Single Page Application (SPA) facilitates developers and saves time.

[16] explores the development of an e-commerce website platform designed for small businesses. In small-scale businesses, operations such as order recording and sales transactions are typically managed manually. These challenges make it difficult for business owners to identify products that are popular among customers. Developing an application using the Scrum method assists the development team and small-medium enterprise management in defining system requirements and facilitates the development team in obtaining reviews of the built features, enabling a faster deployment process.

 ISSN:2302-8149
e-ISSN:2540-9719

Finally [17] discusses the development of a web-based information system designed for managing training program data, participant attendance records, participant details, and invoice data, which were previously not optimally managed by the company. The website was built using ReactJS, MySQL, Java, and JavaScript, with the project completed in eight sprints over 33 days. The system includes multiple user roles, namely admin, user, and guest, and consists of 21 product backlogs, covering pages such as Home Page, Login Page, Order Page, and History Admin Page. This inefficiency has affected the effectiveness and efficiency of training program execution. The website is built using ReactJS, MySQL, Java, and JavaScript. The research aims to assist the company in processing training data according to the scheduled timeline of the training program.

Table 1 summarizes the focus areas, technologies used, and key findings of the studies discussed above, providing a comparative overview of their contributions to web application development using ReactJS and Agile Scrum methodologies.

Table 1. Analysis of related studies and implementation

Study	Focus Area	Technology Used	Key Findings
[13]	Employee Attendance System	ReactJS, Node.js, ExpressJs	Improved attendance tracking and iterative feature development
[14]	Online Survey Platform	ReactJS	Enhanced UI development and iterative design improvements
[15]	Document Management & Survey System	ReactJS (Package Material UI)	Efficient SPA development and enhanced UI/UX
[16]	E-commerce Platform for SMEs	ReactJS	Facilitated iterative system improvements and faster deployment
[17]	Training Program Management System	PERN (PostgreSQL, ExpressJS, ReactJS, NodeJS)	Improved training data management and system efficiency
This Study	AGV Control System	ReactJS (Package Shadcn UI), TypeScript, and Tailwind CSS	Adaptation of Agile Scrum by integrating Sprint Retrospective within Sprint Review for streamlined feedback and continuous improvement in AGV control system development. Black Box Testing confirmed system functionality without constraints.

Unlike previous studies that focus on general web application development using ReactJS [13], [14], [15], [16], [17] this research specifically explores the application of the Agile Scrum methodology for developing an AGV control system using ReactJS, TypeScript, and Tailwind CSS. Additionally, this study leverages modern UI/UX principles with the utilization of the shadcn/ui package in React, which accelerates the development process. The system is evaluated through Black Box Testing to ensure full functionality in an industrial setting. The web-based application to be developed will enable administrators to control AGV robots via a dashboard, assigning tasks through the Task Management feature. Furthermore, administrators can add, read, update and delete data within the Station Management, AGV Management, and Task Management features. This study primarily focuses on frontend development, ensuring that the interface is not only functional but also visually intuitive, seamlessly integrating with backend APIs. The development process incorporates Agile Scrum with an adaptation that integrates Sprint Retrospective within Sprint Review to streamline feedback and optimize iteration cycles. The objective of this research is to create a robust web-based AGV control system that leverages modern technologies while adopting the Agile Scrum approach to optimize the development lifecycle.

ISSN:2302-8149
e-ISSN:2540-9719

3 Research Method

The method used in this research adopts the Agile Scrum methodology, which is one of the System Development Life Cycle (SDLC) methodologies. Agile is a software development methodology that focuses on an iterative work process, allowing teams to quickly adapt to changes in the project. However, no modifications are made until the Sprint concludes, unless the change is considered significant enough to halt the Sprint [18]. With this approach, the development process can be completed in a shorter time without compromising quality. Another definition of Agile Development is a rapid software development method that accommodates changing requirements within a relatively short time [19], [20].

Agile has several frameworks, one of which is Scrum. Scrum is a software engineering method that adopts the principles of the Agile approach, focusing on strong team collaboration, incremental product development, and an iterative process to achieve the final outcome [21]. Figure 1 illustrates the stages of software development using the Agile Scrum methodology.

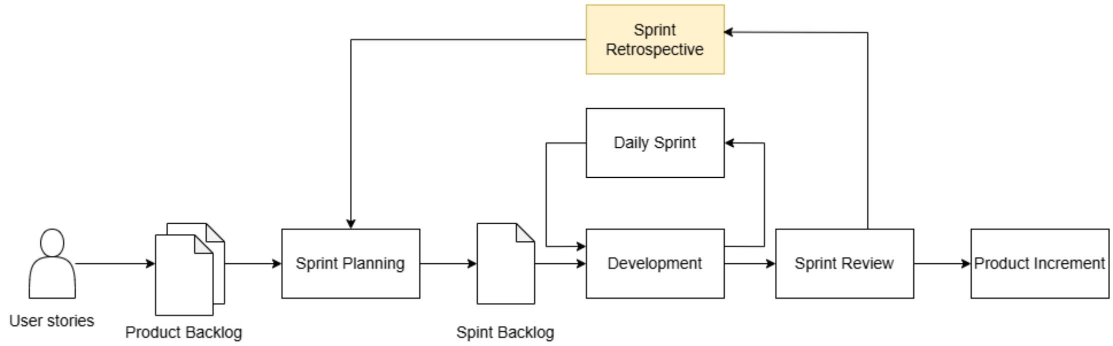

Figure 1. Agile scrum method

In this study, the sprint retrospective was combined with the sprint review, rather than being held as a distinct activity, to better align with the company's existing workflow and reduce the time spent in separate meetings. This adjustment was made to streamline the process, ensuring that the team could efficiently manage their time while maintaining focus on both progress reporting and improvement discussions in one session. The adaptation of the Agile Scrum methodology helped the team sustain development efficiency by integrating feedback and improvement discussions within a single session, thereby fostering a quicker response to issues and improving overall team dynamics. Figure 2 illustrates the Agile Scrum workflow as adapted by the company, where the sprint retrospective occurs immediately after reporting the progress of a sprint cycle during the sprint review.

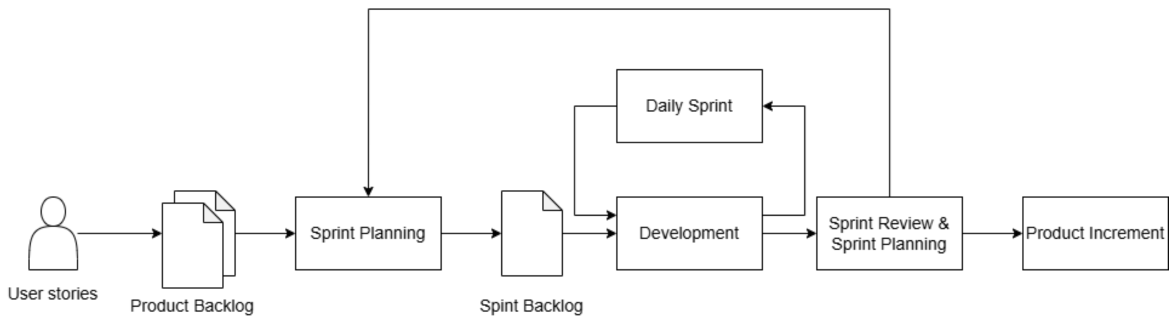

Figure 2. Adopted agile scrum method

3.1. User Stories and Product Backlog

User stories are brief, simple descriptions focused on users regarding a feature or function to be developed [22]. The results of the user stories help identify the problems faced by users. In addition to serving as a communication tool, user stories play an essential role in setting priorities for creating the product backlog [23]. The product backlog is a prioritized list of features and requirements that will be worked on during product development. At this stage, the researcher conducted discussions with

ISSN:2302-8149
e-ISSN:2540-9719

STECHOQ employees to determine the data specifications and features of the web application to be developed.

3.2. Sprint Planning

Sprint planning is the stage that follows the creation of the product backlog. The purpose of sprint planning is to determine the priority of tasks to be completed first [24]. The result of sprint planning is the sprint backlog, which is a list of items that the development team must complete within the sprint cycle. At this stage, the researcher discussed with the team to determine which items from the product backlog would be worked on first.

3.3. Daily Sprint and Development

The daily sprint is a communication within the team to share work status, monitor progress, and collaborate on resolving potential challenges or obstacles [25]. The daily sprint serves as a means to accelerate product progress in achieving sprint goals [26]. After the daily sprint, the next stage is development, which involves executing specific tasks as outlined in the sprint backlog. This phase is divided into several stages, including architectural design, database design, interface design, architectural implementation, database implementation, interface implementation, and system integration. In this phase, the researcher focused on system design, interface design, interface implementation, and system integration. The system integration in this research uses a Representational State Transfer (REST) based Application Programming Interface (API).

3.4. Sprint Review

In this phase, the completed features are demonstrated to obtain feedback from the product owner and to ensure that the developed features meet the product owner's requirements. At this stage, the researcher conducted feature testing with team members first to identify errors and bugs early, ensuring that the application's functionality runs stably before being presented to the product owner, which in this case are the STECHOQ employees, during the sprint review stage.

3.5. Sprint Retrospective

The goal of the sprint retrospective is to evaluate the experiences during the sprint and identify improvement steps for future work processes [27] with the expectation of maximizing work quality and team collaboration in the next sprint. In this research, the sprint retrospective was not conducted as the researcher adapted to the culture of the research partner.

4 Results and Analysis

4.1. User Stories and Product Backlog

User stories are necessary to understand the issues currently faced by the product owner. In this study, the alignment of system requirements was delegated to STECHOQ employees. The researcher held discussions with STECHOQ employees as the basis for creating the product backlog. The product backlog in Table 1 was obtained through discussions between STECHOQ employees and the team. Based on Table 1 there are 6 features or product backlogs that will be worked on during the web-based application development process.

Table 1. Product backlog

Feature	Description	Priority
Login	Admin can log in using the provided username and password	High
Dashboard	Displays information summarizing the total data of AGV types, stations, and batteries, as well as robot control	Medium
Robot Control	Admin can control the AGV Lidar robot, AGV Line Follower, and both robots	High
Station Management	Admin can view all Station data, including station name, coordinates, etc., add, modify, and delete station data	High
AGV Management	Admin can view all AGV data, including AGV name, AGV robot type, etc., add, modify, and delete AGV data	High
Task Management	Admin can view all Task data, including AGV name, destination station, etc., add, modify, and delete AGV data	High

 ISSN:2302-8149
e-ISSN:2540-9719

4.2. Sprint 1

4.2.1 Sprint Planning

In the first sprint planning, the team held a meeting to determine the product backlog to be worked on during the first sprint. The team consisted of the Scrum Master, Mechanical Engineer, Hardware Engineer, Electrical Engineer, and Software Engineer. The Software Engineer in this project included two Backend Programmers and one Frontend Programmer. In this sprint, the iteration was set to two cycles, with each iteration lasting one week. The result of the sprint planning was the sprint backlog, which in this study focused on frontend development. The main focus of the frontend work was to build the User Interface (UI) and integrate the REST API provided by the Backend Programmer. Table 2 shows the sprint backlog for the Frontend Programmer, which contains a list of tasks based on the product backlog in Table 1.

Table 2. Frontend Sprint Backlog (Sprint 1)

Description	Details	Priority
Designing the User Interface (UI) for Login, Dashboard, Station Management, and AGV Management	Designing UI/UX using Figma	Low
Slicing the UI design	Converting UI design into ReactJS and Tailwind CSS	Medium
Integrating the Login, Dashboard, AGV Management, and Station Management APIs	Connecting frontend UI with backend REST API using TypeScript	High
Testing Login, Dashboard, AGV Management, and Station Management features	Conducting feature testing with the team and STECHOQ employees	High

4.2.2. Daily Sprint and Development

On each active day, a daily sprint was conducted to present the progress of the sprint backlog being worked on by the team. The daily sprint lasted for 5-15 minutes and served as a coordination tool to ensure that the tasks in the sprint were on track. For example, the researcher, as the frontend programmer, presented the work that had been completed, such as designing the Unified Modeling Language (UML), UI design, slicing the design into ReactJS and Tailwind CSS, integrating the REST API, and testing features. Initial testing was conducted by the team to ensure functionality before presenting it in the sprint review. The development stage began with the design of the Unified Modeling Language (UML). UML was used to describe the software design to be built [28]. The UML diagrams created in this study included use case diagrams, sequence diagrams, and activity diagrams. The use case diagram shown in Figure 3 illustrates the admin's access rights when interacting with the web application. The admin must first log in to access the Dashboard, Station Management, AGV Management, and Task Management features.

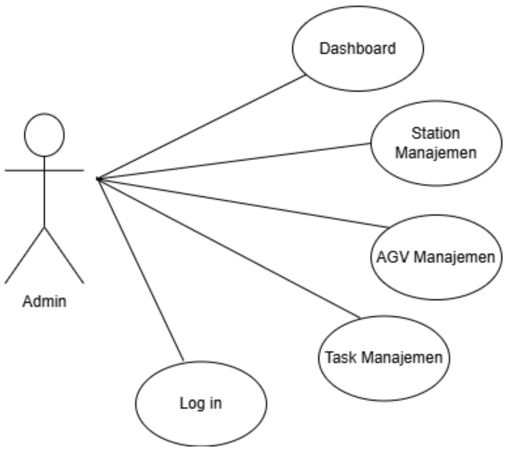

Figure 3. Use Case Diagram Sprint 1

ISSN:2302-8149
e-ISSN:2540-9719

The sequence diagram shown in Figure 4 illustrates the interaction scenario of the admin on the web application. The admin fills in the username and password inputs. If the inputs are incorrect, such as missing fields, an error message will appear. If the admin fills in all the input fields and the username and password match, the admin will be directed to the Dashboard page. If there is an error in the username or password, a message will appear on the admin page.

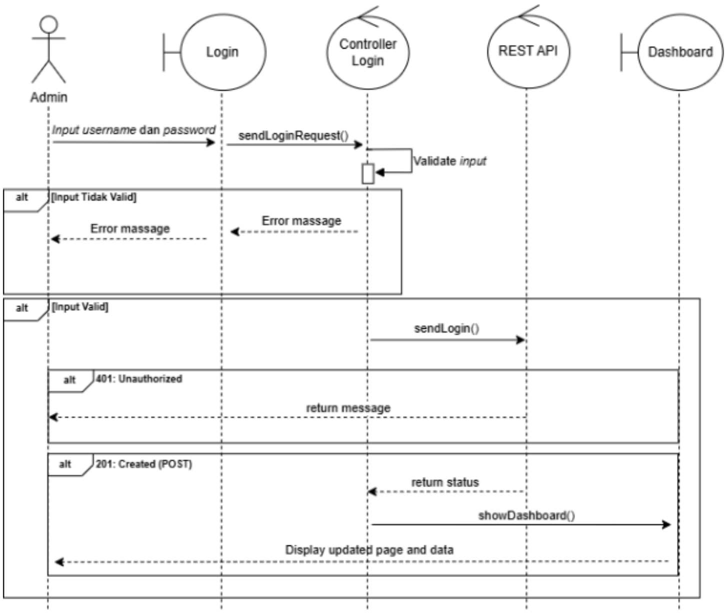

Figure 4. Sequence diagram web

Figure 5 shows the activity diagram when an admin accesses the web application. Admin can visit the web and fill out a login form to access several menus within it. These include Station Management, AGV Management, and Task Management menus.

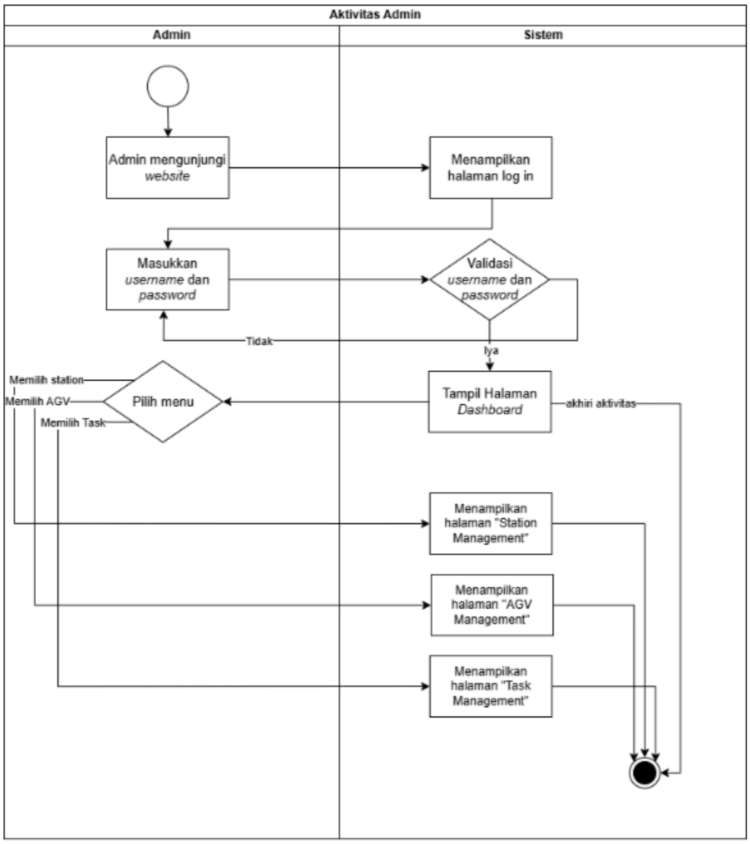

Figure 5. Activity diagram web

ISSN:2302-8149
e-ISSN:2540-9719

After completing the UML design, the next phase is User Interface (UI) design as shown in Figure 6. The UI design was created using Figma and includes several pages, such as Login, Dashboard, Station Management, and AGV Management pages.

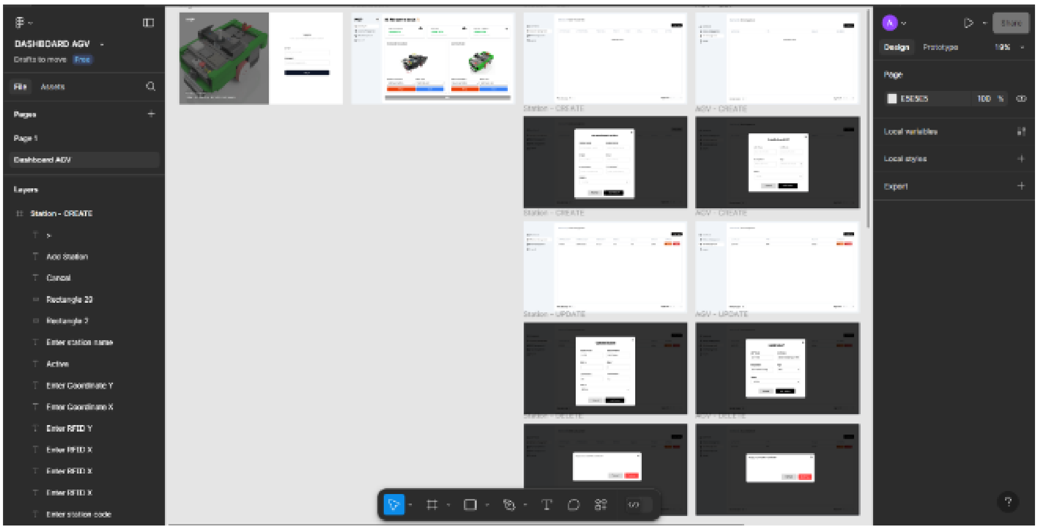

Figure 6. UI display of sprint backlog 1

The next phase involves slicing the UI design into ReactJS library using Tailwind CSS, followed by REST API integration from the Backend Programmer using TypeScript. The design slicing process and REST API integration were carried out gradually, starting from the Login page, Dashboard, Station Management, to the AGV Management page. Figure 7 shows the results of slicing the UI design into ReactJS and Tailwind CSS, along with REST API integration for the Dashboard page.

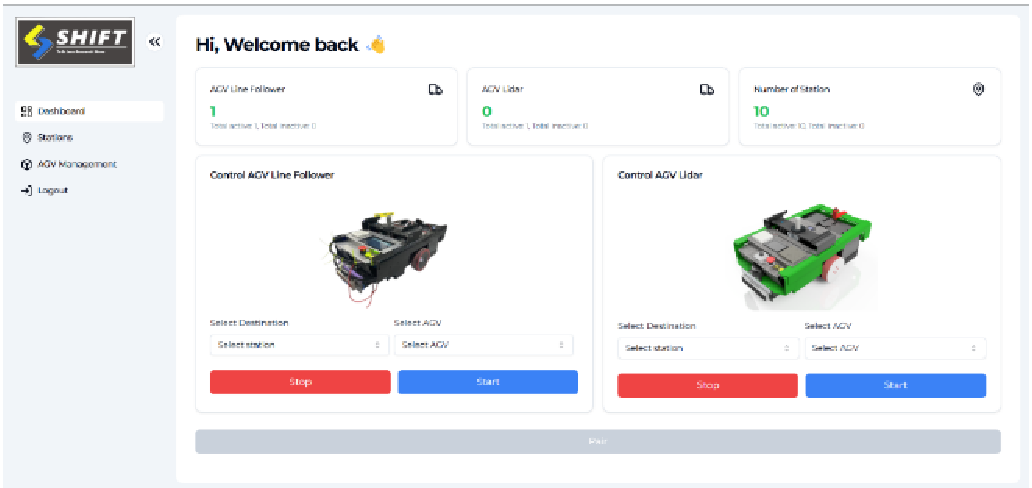

Figure 7. Display of sliced dashboard page

4.2.3 Sprint Review

At this stage, the team presented the development progress together with STECHOQ employees. Sprint reviews were routinely held every week to report on the progress of completed sprint backlogs, constraints and solutions during the development process, and plans for the next sprint. Sprint backlog result demonstrations were conducted at this stage. Then, testing was performed after the sprint review. The test results in Table 3 on the next page show the successful testing of all features contained in the first sprint backlog. Successfully tested features include Login, Dashboard, Station Management, and AGV Management.

 ISSN:2302-8149
e-ISSN:2540-9719

Table 3. First sprint test results

Testing Activity	Output	Conclusion
Input username and password	Admin successfully logs in and will display Dashboard page	[√] Success [ ] Failed
Click Dashboard menu	Shows AGV and station data summary and robot control display	[√] Success [ ] Failed
Click Station Management menu	Shows station data summary	[√] Success [ ] Failed
Click Add New button on Station Management page	Modal or pop-up appears containing input fields and can add data and display it on Station Management page	[√] Success [ ] Failed
Click Update button on Station Management page	Modal or pop-up appears containing input fields from updated data and can save changes and display them on Station Management page	[√] Success [ ] Failed
Click Delete button on Station Management page	Modal or pop-up confirmation appears and can delete data and display it on Station Management page	[√] Success [ ] Failed
Click AGV Management menu	Shows AGV data summary	[√] Success [ ] Failed
Click Add New button on AGV Management page	Modal or pop-up appears containing input fields and can add data and display it on AGV Management page	[√] Success [ ] Failed
Click Update button on AGV Management page	Modal or pop-up appears containing input fields from updated data and can save changes and display them on AGV Management page	[√] Success [ ] Failed
Click Delete button on AGV Management page	Modal or pop-up confirmation appears and can delete data and display it on AGV Management page	[√] Success [ ] Failed

The testing results confirm that core functionalities, including user authentication, station management, and AGV management, operate correctly without errors. Successful create, read, update, and delete data operations ensure smooth data handling, while responsive UI interactions enhance usability. The system effectively displays relevant data, allowing administrators to manage tasks efficiently.

4.3. Sprint 2

4.3.1 Sprint Planning

The second sprint backlog contains the remainder of the product backlog that wasn't completed in the first sprint. Uncompleted features were included in the second sprint backlog as presented in Table 4. Table 4 presents the uncompleted features, namely Task Management along with AGV robot integration.

Tabel 4. Frontend sprint backlog (sprint 2)

Description	Details	Priority
Create UI design for Task Management	Design UI/UX for website using Figma	Low
UI design slicing	Translate UI design to ReactJS and Tailwind CSS code	Medium
Task Management API Integration and AGV Robot Control	Connect Backend API url with sliced UI using TypeScript	High
Testing Task Management features and AGV Robot Control	Perform feature testing with team and STECHOQ employees	High

ISSN:2302-8149
e-ISSN:2540-9719

4.3.2 Daily Sprint

On each working day, the team conducts daily sprints lasting about 5-15 minutes to provide updates on progress achieved and discuss potential obstacles. In the second daily sprint, the researcher as Frontend Programmer conveyed what would be worked on in the development stage. Several things discussed included UML design, such as use case diagrams, sequence diagrams, and activity diagrams for Task Management features, UI design, UI design slicing along with API integration with Backend Programmer for Task Management features and AGV Robot Control.

Testing of Task Management features and AGV Robot Control was also part of the discussion in the meeting. After the daily sprint, the team immediately proceeded to the development stage. Starting with UML design for sprint backlog 2. The UML designed was for the Task Management feature. Figure 8 shows the use case diagram for Task Management and AGV Robot Control. Figure 8 explains that on the Dashboard page, admin can control the AGV robot and on the Task Management page, admin can view data, add data, modify data, and delete data.

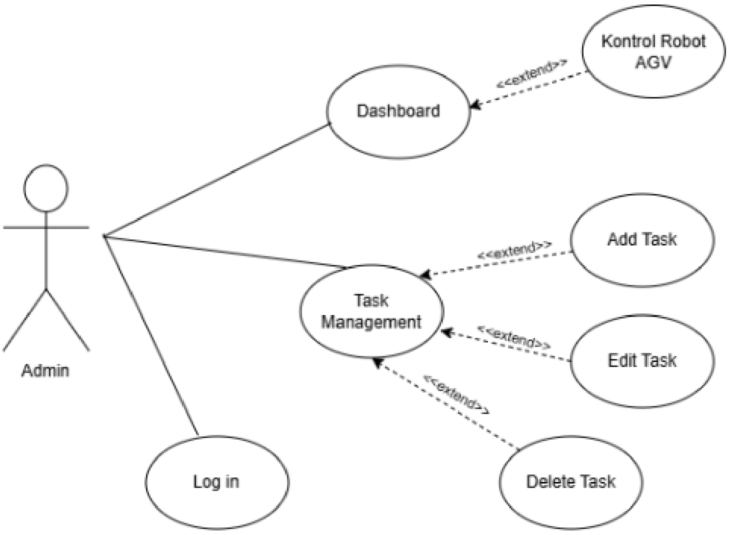

Figure 8. Use case diagram sprint 2

Figure 9 shows the sequence diagram for the Task Management feature. To access the Task Management menu, admin needs to click on the menu, after which the Task Management data list will be displayed on the admin page.

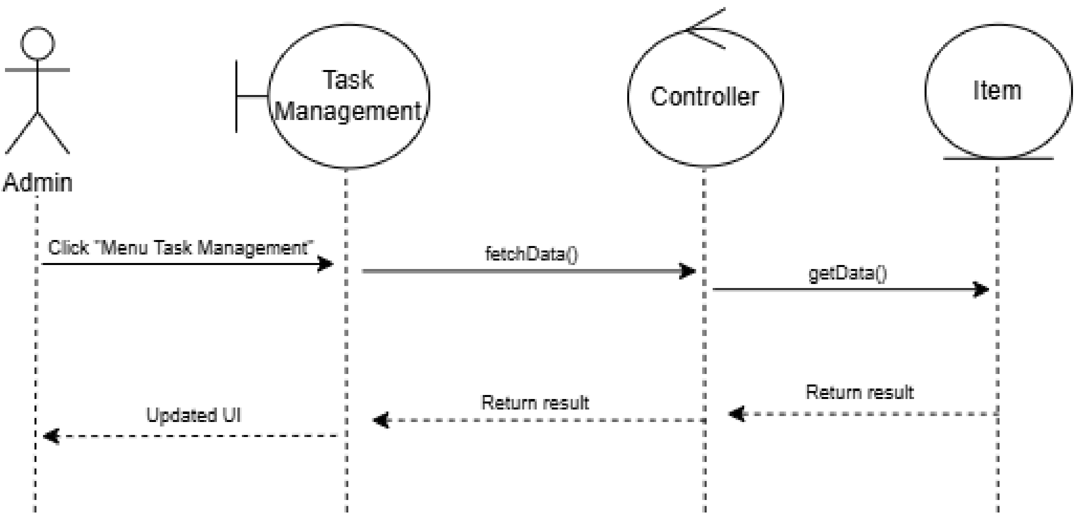

Figure 9. Sequence diagram task management

Figure 10 on the next page shows the Activity Diagram, which depicts the activities that will be performed by the admin in the Task Management feature. After the admin selects the Task Management menu, they can view the Task Management data list. Task data is successfully added or

ISSN:2302-8149
e-ISSN:2540-9719

modified when the data type matches the input form; if it doesn't match, it will remain in the input form and display an error message in the input field with the mismatched data type.

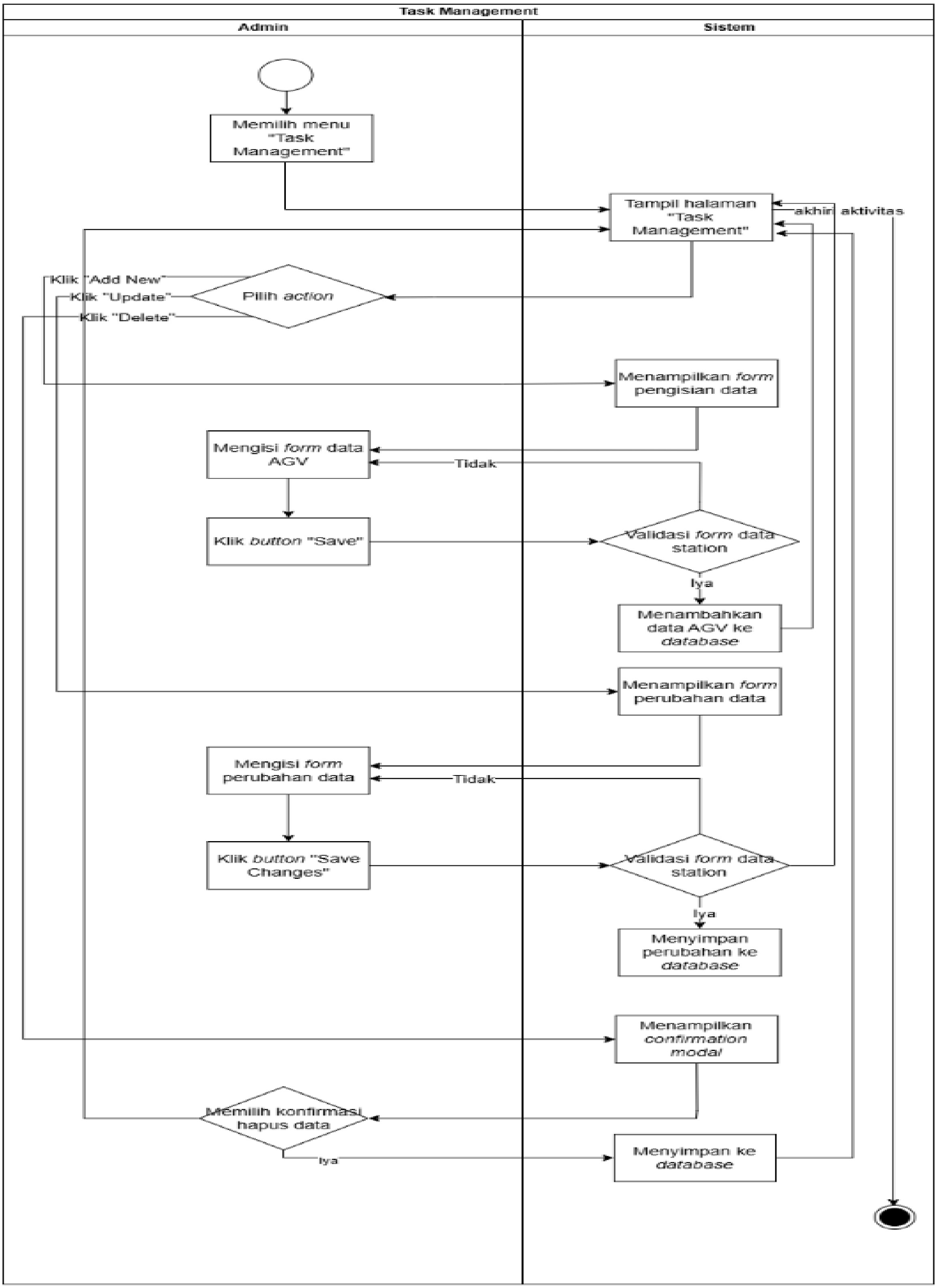

Figure 10. Activity diagram task management

After completing the UML design, the researcher created the UI design for the Task Management feature. Figure 11 on the next page shows the UI design using Figma. The page design created is similar to the pages in the Station Management and AGV Management features.

ISSN:2302-8149
e-ISSN:2540-9719

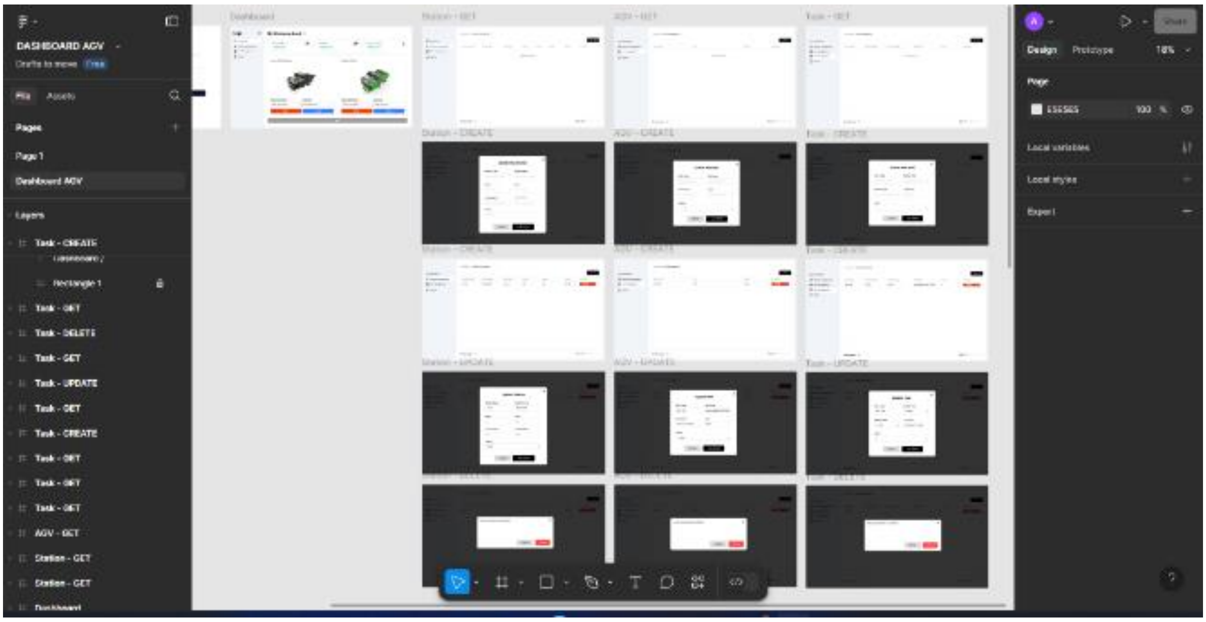

Figure 11. UI Display for Sprint Backlog 2

The next step is converting the UI design into components in the ReactJS library using Tailwind CSS, followed by the API integration process from the Backend Programmer using TypeScript. This process was carried out for the Task Management feature and Robot Control. As shown in Figure 12, which displays the implementation results of the UI design in ReactJS with Tailwind CSS, complete with API integration for the Task Management feature and Robot Control.

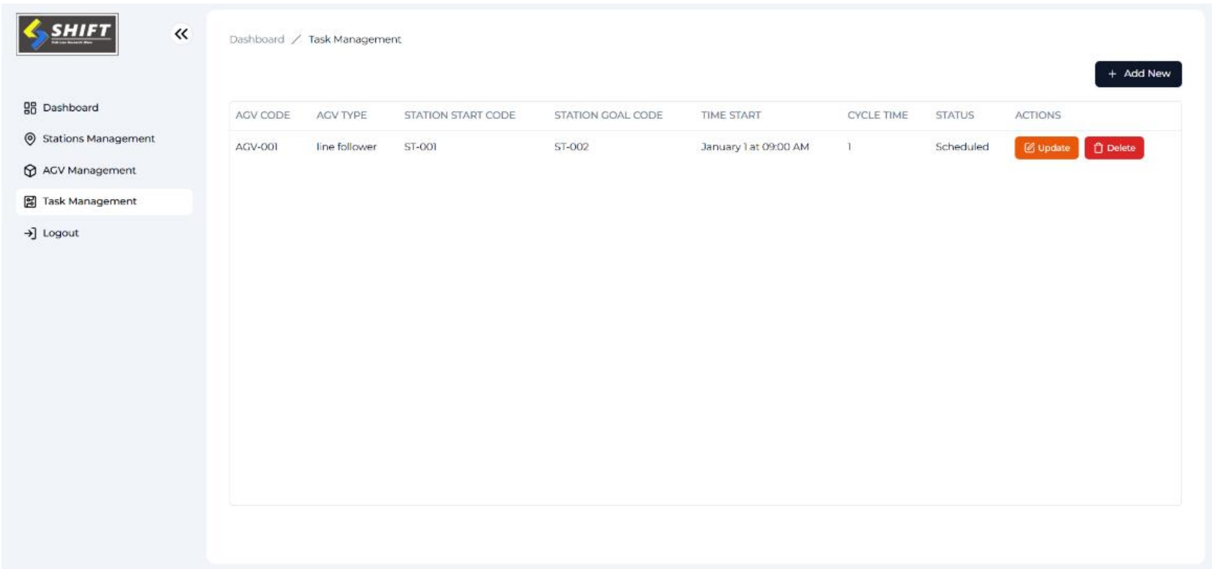

Figure 12. Display of sliced task management feature

4.3.3 Sprint Review

The completed sprint backlog was then presented together with the team and STECHOQ employees to receive feedback on the implemented features. Testing of Task Management and Robot Control features was conducted after the sprint review. Table 5 shows the test results for the Task Management and Robot Control features. Where the Task Management and Robot Control features can be run without errors or bugs.

Table 5. Second sprint test results

Testing Activity	*Output*	Conclusion
Click Dashboard menu	AGV robot and Station data summary appears. Robot control feature display is present	[√] Success [ ] Failed
Click Station and AGV robot	Station list and AGV robot list appears	[√] Success [ ] Failed
Click Start button	Robot successfully starts moving when button is clicked	[√] Success [ ] Failed
Click Stop button	Robot successfully stops when button is clicked	[√] Success [ ] Failed
Click Task Management menu	Task data summary appears	[√] Success [ ] Failed
Click Add New button on Task Management page	Modal or pop-up appears containing input fields and can add data and display it on Task Management page	[√] Success [ ] Failed
Click Update button on Task Management page	Modal or pop-up appears containing input fields from updated data and can save changes and display them on Task Management page	[√] Success [ ] Failed
Click Delete button on Task Management page	Modal or pop-up confirmation appears and can delete data and display it on Task Management page	[√] Success [ ] Failed

These successful tests demonstrate that the core features of the system task management and robot control function as intended, ensuring seamless interaction between users and the AGV system. The ability to add, update, and delete tasks without issues suggests that the system effectively handles data operations. Additionally, the robot control functionality, including the ability to start and stop the robot, performed reliably, indicating that API communication between the Frontend Programmer and Backend Programmer is stable.

Challenges encountered during the second sprint involved the integration of robot control through the API provided by the backend. One major issue was the change in the payload structure implemented by the Backend Programmer, which caused discrepancies in communication between the frontend and backend. To address this issue, the Backend Programmer and Hardware Programmer held in-depth discussions to clarify the API data requirements, ensuring seamless robot control. Once the API specifications were finalized, the Frontend Programmer who is also the author of this study refactored the code to align with the updated payload structure.

Through effective communication and collaboration among the development team, these changes were successfully implemented without disrupting system stability, ultimately improving the integration of robot control. This resolution highlights the importance of clear coordination between software and hardware teams in technology-driven projects.

5 Conclusion

This research successfully adopted the Agile Scrum methodology for web-based application development, involving user stories and product backlog, daily sprint and development, sprint review, and iterative feature testing during daily sprints and sprint reviews. The sprints in this research were conducted twice, with each sprint lasting one week. The first sprint delivered satisfactory results by successfully implementing Login, Dashboard, Station Management, and AGV Management features. All features were tested through Black Box Testing that showed success rates without any technical constraints. The second sprint results provided satisfactory test results similar to the first sprint. Features tested in the second sprint were Task Management and Robot Control.

However, adjustments were made to the adopted Agile Scrum methodology. The sprint retrospective was combined with the sprint review to align with the company's workflow,

ISSN:2302-8149
e-ISSN:2540-9719

streamlining the process and allowing the team to efficiently address progress and improvements in a single session. The success of these sprints provides a strong foundation for future development, including the integration of real-time monitoring using IoT, machine learning for optimizing AGV task assignments, and enhanced security measures for system stability. Additionally, implementing obstacle detection and avoidance using LiDAR, ultrasonic sensors, or computer vision with deep learning models could improve AGV navigation and safety. Future research could also explore the comparative effectiveness of different Agile methodologies in industrial software development and the use of cloud computing for scalable and efficient AGV fleet management.

Reference

[1] BPJS Ketenagarkerjaan, "Kecelakaan Kerja Makin Marak dalam Lima Tahun Terakhir." Accessed: Dec. 23, 2024. [Online]. Available: https://www.bpjsketenagakerjaan.go.id/berita/28681/Kecelakaan-Kerja-Makin-Marak-dalam-Lima-Tahun-Terakhir

[2] K. G. Lumenta, R. J. Pio, and S. A. P. Sambul, "Pengaruh Keselamatan dan Kesehatan Kerja Terhadap Produktivitas," *Productivity*, vol. 2, no. 2, pp. 102–107, Apr. 2021, [Online]. Available: https://ejournal.unsrat.ac.id/v3/index.php/productivity/article/view/33532

[3] M. Y. F. Kurniawan and G. Wijonarko, "Pengaruh Dampak Keselamatan dan Kesehatan (K3), Kompensasi dan Lingkungan Kerja Fisik Terhadap Produktivitas Karyawan di PT. Najatim Dickyard," pp. 1–15, Feb. 2022, [Online]. Available: http://repositori.stiamak.ac.id/id/eprint/318

[4] B. Alojaiman, "Technological Modernizations in the Industry 5.0 Era: A Descriptive Analysis and Future Research Directions," *Processes*, vol. 11, no. 5, p. 1, May 2023, doi: 10.3390/pr11051318.

[5] A. F. Costa, M. D. S. Carvalho, M. Henriques, and P. V. Ferreira, "Strategy for the Introduction of Autonomous Driving Technologies: A Case Study in the Logistics Area of an Automotive Company," in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 337–345. doi: 10.1016/j.procs.2022.08.041.

[6] M. De Ryck, M. Versteyhe, and F. Debrouwere, "Automated Guided Vehicle Systems, State-of-the-Art Control Algorithms, and Techniques," *J Manuf Syst*, vol. 54, pp. 152–173, 2020, doi: https://doi.org/10.1016/j.jmsy.2019.12.002.

[7] M. Dobrzanska and P. Dobrzanski, "Simulation Modelling of Material Handling Using AGV," *IOP Conf Ser Mater Sci Eng*, vol. 1199, no. 1, p. 012015, Nov. 2021, doi: 10.1088/1757-899x/1199/1/012015.

[8] R. Abdurrahman, B. Setiawan, and I. Siradjuddin, "Sistem Kontrol Kecepatan Robot Automatic Guided Vehicle (AGV) dengan Metode Fuzzy Logic," vol. 18, no. 1, pp. 85–93, Apr. 2020, doi: 10.33795/eltek.v18i1.229.

[9] H. Juawdesb and S. L. Mufreni, "Rancang Bangun Website Pada Kelurahan Anggilowu," *Procedia of Engineering and Life Science*, vol. 1, no. 2, Jun. 2021, doi: https://doi.org/10.21070/pels.v1i2.974.

[10] T. K. Patra and Dr. S. Jain, "Comparison of JavaScript Frameworks: React.js and Vue.js," *International Journal of Innovative Science and Research Technology*, vol. 7, no. 9, pp. 254–258, 2022, doi: https://doi.org/10.5281/zenodo.7108920.

[11] K. Bhavsar, V. Shah, and S. Gopalan, "Scrum: An Agile Process Reengineering In Software Engineering," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* , vol. 9, no. 3, 2020, doi: 10.35940/ijitee.C8545.019320.

[12] L. Setiyani, "Pengujian Sistem Informasi Inventory Pada Perusahaan Distributor Farmasi Menggunakan Metode Black Box Testing," *Jurnal Ilmu Komputer dan Teknologi Informasi*, vol. 4, no. 1, pp. 20–27, Apr. 2019, doi: 10.36805/technoxplore.v4i1.539.

[13] I. Tahyudin and Zidni Iman Sholihati, "Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 169–176, Feb. 2022, doi: 10.29207/resti.v6i1.3793.

[14] C. Fauzan Alqodri, “Pengembangan Frontend Website Pada Platform Survey Online Menggunakan Agile Scrum,” *Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi*, vol. 3, no. 2, 2024, doi: 10.31284/p.semtik.2024-2.6239.

[15] Nasution and L. Iswari, “Penerapan React JS Pada Pengembangan FrontEnd Aplikasi StartUp Ubaform,” *Prosiding Automata (Ajang Unjuk Tugas Akhir oleh Mahasiswa Informatika)*, vol. 2, no. 2, Aug. 2021, [Online]. Available: https://journal.uii.ac.id/AUTOMATA/article/view/19532

[16] A. Wulansari, D. S. Y. Kartika, Agussalim, S. F. A. Wati, E. M. Safitri, and A. S. Fitri, “E-commerce Website Development Using Scrum Methods on Small Business ,” *IJCONSIST JOURNALS* , vol. 3, no. 2, pp. 8–12, Jun. 2022, doi: doi.org/10.33005/ijconsist.v3i2.69.

[17] H. J. S. Sitio, I. Christovita, R. K. Ahmad, and Y. Setiawan, “Web-based Application Development for Training Data Management Using ReactJS,” *Indonesian Journal of Multidisciplinary Science*, vol. 2, no. 6, pp. 2573–2588, May 2023, doi: https://doi.org/10.55324/ijoms.v2i6.461.

[18] T. Satpathy, *A Guide to the Scrum Body of Knowledge (SBOK® Guide)*, 4th ed. Arizona: SCRUMstudy™, 2022. [Online]. Available: https://www.scrumstudy.com/sbokguide/download-free-buy-sbok

[19] N. Syahputri and M. I. P. Nasution, “Pengembangan Aplikasi E-Commerce Pada Penjualan Produk HNI menggunakan Metode Agile,” *SISTEMSI: Jurnal Sistem Informasi*, vol. 13, no. 2, pp. 643–655, Mar. 2024, doi: 10.32520/stmsi.v13i2.4009.

[20] N. Hikmah, A. Suradika, R. Andi, and A. Gunadi, “Metode Agile Untuk Meningkatkan Kreativitas Guru Melalui Berbagi Pengetahuan (Knowledge Sharing),” *Jurnal Instruksional*, vol. 3, no. 1, pp. 30–39, 2021, doi: 10.24853/instruksional.3.1.30-39.

[21] Hanry Ham, “Metodologi Scrum,” BINUS University School of Computer Science. Accessed: Dec. 15, 2024. [Online]. Available: https://socs.binus.ac.id/2019/12/23/metodologi-scrum/

[22] Geeks for Geeks, “User Stories in Agile Software Development.” Accessed: Nov. 29, 2024. [Online]. Available: https://www.geeksforgeeks.org/user-stories-in-agile-software-development/

[23] Yeliz Obergfell, “Scrum User Stories: A Comprehensive Overview for Effective Scrum User Story Management.” Accessed: Nov. 29, 2024. [Online]. Available: https://www.scrum-institute.org/Scrum_User_Stories.php

[24] William Wijaya and Fenina Adline Twince Tobing, “Implementation of Scrum Method for Designing Website-Based E-commerce Application (Case Study: Putra Prabu Workshop),” *Ultimatics: Jurnal Teknik Informatika*, vol. 15, no. 2, pp. 106–114, Jan. 2024, doi: https://doi.org/10.31937/ti.v15i2.3487.

[25] A. Menzinsky, G. López, J. Palacio, and J. López, *Scrum Manager: Core Subject Area I*, 2.6.1. Scrum Manager, 2019. [Online]. Available: https://www.scrummanager.com/files/scrum_manager_en.pdf

[26] A. Andipradana and K. Dwi Hartomo, “Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum,” *Jurnal Algoritma*, vol. 19, no. 1, pp. 161–172, 2021, doi: https://doi.org/10.33364/algoritma/v.18-1.869.

[27] K. S. Rubin, *Essential SCRUM: A Practical Guide to The Most Popular Agile Process*, 1st ed., vol. 1. United States of America: Addison-Wesley, 2012.

[28] K. Nistrina and L. Sahidah, “Unified Modelling Language (UML) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru di SMK Marga Insan Kamil,” *Jurnal Sistem Informasi (J-SIKA)*, vol. 4, no. 1, pp. 17–23, 2022, [Online]. Available: https://ejournal.unibba.ac.id/index.php/j-sika/article/view/839