J-COSINE (Journal of Computer Science and Informatics Engineering)
Vol. 9, No. 1, June 2025 E-ISSN:2541-0806
Accredited Sinta-4 by RISTEKDIKTI Decree No. 79/E/KPT/2023 P-ISSN:2540-8895

REST API Development of the Mataram Scout Information System for Member Management and Activity Reporting using Scrum Method

I Desak Putu Diah Anggiani*, Budi Irmawati, Royana Afwani
Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: diahanggiani08@gmail.com, budi-i@unram.ac.id, royana@unram.ac.id

Corresponding Author

The development of a web-based Scout Information System for Kwartir Cabang Mataram aims to improve member and activity data management through a REST API. Existing processes using Google Forms and Excel are inefficient due to manual input, fragmented storage, and lack of real-time access, causing errors and delays. To address this, the proposed system integrates all levels of the scout organization within Mataram City into a centralized platform. Using the Scrum method, it was developed through five sprints and refined based on stakeholder feedback. The REST API includes 25 tested endpoints, ensuring seamless front-end back-end integration and future mobile compatibility. Black-box testing confirmed endpoints reliability. A feasibility survey involving 23 scout representatives gave an average score of 81.74% ("Good"), showing approval and indicating that the system successfully improves data handling compared to the previous manual workflows. These results demonstrate enhanced operational efficiency and reduced administrative delays.

***Key words*: Scout Information System, REST API, Scrum Method, Data Management, Black-Box Testing.**

I. Introduction

Scout Movement has played a vital role in Indonesia youth development, to grow values such as independent, self-confidence, leadership, and solidarity [1]. Officially established on August 14, 1961, the Scout Movement has grown into one of the largest youth organizations in the country [2]. According to Indonesia's Law No. 12 of 2010 concerning the Scout Movement (Undang-Undang Nomor 12 Tahun 2010 tentang Gerakan Pramuka), scouting is a non-formal education aimed at nurturing individual potential, fostering self-discipline, and building essential life skills to prepare the next generation to contribute meaningfully to make a significant contribution to the nation [3].

In Mataram City, scouting activities within school-based scout units (gugus depan) have experienced significant growth, driven by the increasing number of members and events organized. Kwartir Cabang Mataram, the scout organization at the city level, currently oversees 6 kwartir ranting (the scout organization at the sub-district level) and a total of 255 gugus depan, with more than 20,000 registered members. However, Kwartir Cabang Mataram does not yet have a centralized database to manage members data and activities. Discussions with the scout master of SDN 26 Mataram, who also serves as the secretary of Kwartir Cabang Mataram, revealed that the existing data management system is still inefficient. Google Forms is used by each gugus depan to collect individual member and activity data, while Microsoft Excel is used to compile the data and generate summary reports at the sub-district and city levels. Although these tools provide a digital alternative to paper-based forms, they lack features for centralized access control, real-time synchronization, and efficient data retrieval across multiple levels of the scout organization. In addition, both gugus depan and kwartir ranting are required to prepare activity reports in printed (hardcopy) format. These reports are submitted in a hierarchical manner, where gugus depan units submit their reports to the kwartir ranting, which then compiles and forwards them to the kwartir cabang. This manual, paper-based workflow is time-consuming and prone to delays, inconsistencies, and data loss. Given these inefficiencies and fragmentation, there is a clear need for a centralized, integrated system that can streamline data collection, real-time synchronization, and reporting across all levels of Kwartir Cabang Mataram.

In today's digital era, leveraging information technology is essential for improving organizational efficiency, especially in managing member data and activity reporting. Digital transformation offers a promising solution to overcome the data management challenges faced by Kwartir Cabang Mataram. To implement this transformation, a web-based scout information system is being developed to support the multi-level organizational structure of Kwartir Cabang Mataram. As part of a collaborative project, the front-end of the system is developed by another member of the research team. This study specifically focuses on developing the back-end system using REST architecture, which represents a core component of digital transformation by replacing fragmented, manual workflows with integrated, real-time digital services that improve data accessibility, and operational transparency.

The REST API, built with the Next.js framework, enables seamless data integration and real-time access across organizational levels through well-designed endpoints, data processing, database integration, and back-end services to support front-end features. Operating over the HTTP protocol, REST is widely applicable both on the internet and within internal network environments [4]. REST simplifies interactions between distributed systems, improving data consistency and accessibility. The system is developed using the Scrum method, which emphasizes iterative development through sprints, continuous feedback, and collaboration among cross-functional teams [5]. This approach offers flexibility to adapt to changing requirements and helps deliver a system aligned with user needs. Until now, no REST API-based scouting information system has been designed specifically to support the three-tier Scout Movement structure in Mataram. This research addresses this gap by developing back-end features for managing member data and activity reporting with role-based access control, real-time synchronization, centralized database integration, and streamlined reporting workflows. With the implementation of this system, it is expected that the management of scout data in Kwartir Cabang Mataram will become more efficient.

II. Literature Review

In recent years, various studies have been conducted on the development of web-based information systems. As a result, a literature review of previous research relevant to the topic has been carried out to serve as a theoretical foundation and comparison.

Study [6] highlights the use of the Scrum method in developing an information system for retail sales at Rabbani Shoes. The research implemented Scrum to build four main modules: user login, product management, user management, and transaction management. The successful implementation of all planned features according to the product backlog underscores Scrum's effectiveness in providing structured project management.

In line with the focus on the agile Scrum methodology, study [7] discusses the development and implementation of the back-end of a financial reporting system website at SMK Multistudi High School. The back-end was developed using JavaScript, designed to handle business logic and data operations, and supported by MySQL as the database management system. Through the application of Scrum, the web-based financial reporting system successfully delivered features for user registration, authentication, supplier data management, inventory control, sales tracking, and profit and loss reporting.

Another study [8] focuses on the digitization of inventory management through a web-based system developed for Widarapayung Kulon Village Hall. The system utilized a REST API built with Node.js to integrate the front-end and back-end. The REST API was designed to efficiently support CRUD (Create, Read, Update, Delete) operations on inventory data. The development process followed the waterfall model, which was chosen due to the project's well-defined requirements and minimal changes expected during the development.

Study [9] discusses the development of a back-end system for the NTB Mall MSME registration platform and an attendance application. Using an agile approach, the project employed Next.js as the full-stack framework, integrating front-end interfaces with API routing. Furthermore, Prisma ORM facilitated efficient CRUD operations and ensured data integrity through structured schemas and controlled database migrations. In addition, Postman was used to validate API endpoints prior to deployment. Overall, the integration of these tools resulted in a scalable and maintainable system well-suited for managing MSME and attendance data effectively.

In Study [10], the development of a web-based booking system for Potret Kecilmu addressed challenges in manual booking management. The study proposed a more efficient booking system built using Next.js and Prisma ORM, adopting an agile development methodology. The findings indicate a 46.15% simplification of the booking process, measured by comparing the number of steps before and after system implementation, along with notable improvements in booking management efficiency, a reduction in operational errors, and enhanced user experience.

Study [11] discusses the development of a web-based application to monitor and control Automated Guided Vehicles (AGVs) at PT. Stechoq Robotika Indonesia. The system was developed with a REST API back-end integrated with a ReactJS front-end to enable real-time communication between the web application and AGVs. The REST API facilitated efficient data exchange and control, making it a key component in the system's functionality. The application was developed using the Agile Scrum methodology, ensuring that the system met user needs through continuous improvements and rapid adaptability. By conducting regular sprints and integrating sprint reviews and retrospectives, the development team was able to refine API functionalities based on feedback from the front-end team and end-users. This iterative approach facilitated better synchronization between back-end and front-end modules, ensuring that data exchange and features were reliable and responsive. Key features implemented included login, dashboard, station management, AGV management, task management, and robot control, all of which successfully operated without issues as confirmed by iterative black-box testing.

Several studies have highlighted the significance of applying the Agile Scrum methodology, particularly in the development of web-based information systems. Research [6], [7], [11] show that Scrum's iterative approach facilitates rapid adaptation to user needs, improves team collaboration, and ensures the successful delivery of system features. Additionally, studies [8], [11] demonstrate the effectiveness of using REST APIs for seamless communication between the back-end and front-end, enabling real-time data exchange and control. Furthermore, research [9], [10] highlight the use of Next.js and Prisma ORM in system development, ensuring smooth integration and efficient database management through structured schemas and controlled migrations. While the "Ayo Pramuka" app has been developed, it primarily operates at the national level and does not address the operational needs at the city level. This system lacks support for city-specific administrative workflows, as described by the secretary of Kwartir Cabang Mataram during the interview. This study addresses that gap by focusing on a back-end system specifically designed for the three-tier Scout Movement structure in Mataram.

This research focuses on the development of a REST API for the Mataram Scout Information System. The system is built using established technologies, including Next.js for structuring server-side API routes, REST API for facilitating communication between system components, and Prisma ORM for efficient and structured interaction with the database. This approach ensures maintainability and adaptability to meet the evolving needs of the Scout organization in Mataram City.

III. Research Methodology

The research method acts as a roadmap for the entire information development process, starting from the planning phase to implementation. Scrum is an agile development method that organizes work into iterative cycles called sprints, allowing teams to deliver incremental improvements [5]. This method encourages collaboration among team members and stakeholders, with regular reviews to adapt to changing requirements throughout the development process. This research applies the Scrum method in the development of the REST API for the Mataram Scout Information System, implementing each phase systematically to ensure structured progress and continuous improvement.

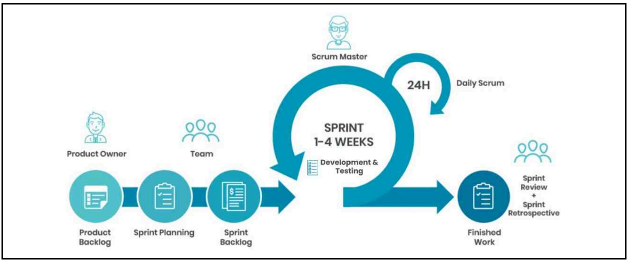


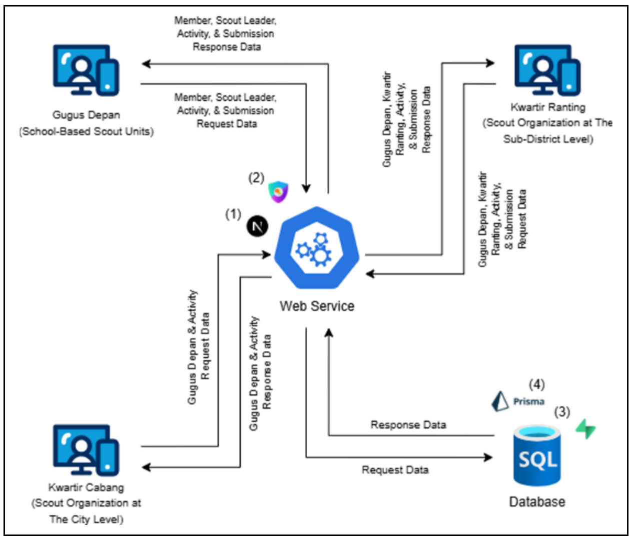
Fig. 1. Scrum Methodology

As shown in Fig. 1, the development process is guided by the product backlog, which contains a prioritized list of REST API features and system requirements such as user authentication endpoints, member data management endpoints, and activity reporting endpoints. These were gathered through interviews with Mataram City scouting representatives. The backlog remains dynamic, allowing updates based on evaluations at the end of each sprint. During the sprint planning phase, the team held collaborative discussions to select and break down backlog items into smaller, actionable tasks, each with an estimated duration based on team consensus. These selected items were then organized into a sprint backlog. Each sprint spans one to four weeks and involves development activities such as REST API endpoint creation, database integration, and testing. Progress during the sprint is tracked using project management tools to ensure transparency and accountability. A detailed overview of the REST API development durations can be found in the sprint backlog table. The system is developed as a full-stack application, with the back-end, which is the focus of this paper, being developed using Next.js, and the front-end handled by another team. Both components are built and integrated together to ensure seamless communication between the REST API and the front-end interface. At the end of the sprint, a sprint review is conducted, to demonstrate the functional system increment to stakeholders. Their feedback on whether the developed features meet their needs is used to refine the product backlog and guide future development [5]. Finally, a sprint retrospective allows the team to reflect on their performance, identify challenges, and discuss improvements for future sprints. This continuous feedback loop ensures the team evolves along with the project's needs, leading to more effective and efficient development over time.

This research adopts black box testing to verify the functionality of the REST API without examining its internal code structure [12]. The testing process is carried out at the end of each sprint, following the completion of features listed in the sprint backlog. This ensures that each implemented endpoint functions correctly based on predefined system requirements [13]. Various inputs, including valid and invalid data, are used to evaluate how well the REST API handles different scenarios. For example, testing scenarios include creating new member records, retrieving member lists by unit level, and submitting activity reports. Valid inputs ensure all required fields are filled in correctly, while invalid inputs include missing parameters, incorrect data formats (e.g., invalid NTA or date), or attempts to access non-existent resources. These cases reflect real-world interactions, such as a scout master submitting an activity report without a date or updating a member that does not exist. Postman is used to execute HTTP methods (GET, POST, etc.) and validate outputs such as response codes and messages. Test results are documented to identify deviations, which are then addressed in the next sprints as part of the Scrum workflow.

IV. Result and Implementation

The system is developed using the Scrum method. The Scrum team consists of one product owner from the Scout organization in Mataram City, one scrum master, and a developer team. I am part of the developer team as a back-end developer and take on the role of scrum master due to the small team size. The implementation process began with initial system design, including the creation of the system architecture and Entity Relationship Diagram (ERD), based on requirements gathered from interviews with Mataram City scouting stakeholders. These designs provided a clear technical foundation for subsequent development.


Member, Scout Leader, Activity, & Submission Response Data
Member, Scout Leader, Activity, & Submission Request Data
Gugus Depan (School-Based Scout Units)
Kwartir Ranting (Scout Organization at The Sub-District Level)
Gugus Depan, Kwartir Ranting, Activity, & Submission Response Data
Gugus Depan, Kwartir Ranting, Activity, & Submission Request Data
(2)
(1)
Web Service
Gugus Depan & Activity Request Data
Gugus Depan & Activity Response Data
(4)
Prisma
(3)
Response Data
Request Data
SQL
Database
Kwartir Cabang (Scout Organization at The City Level)
Fig. 2. System Architecture of Mataram Scout Information System

As shown in Fig. 2, the system adopts a web-based client-server architecture with a REST API approach. REST is an architectural style that uses standard HTTP methods, such as GET, POST, PUT, PATCH, and DELETE, to perform operations on resources [4]. These resources are represented by URLs and can be manipulated through interactions between the client and server. The system involves three main types of users: the gugus depan, the kwartir ranting, and the kwartir cabang. Each type of user interacts with the back-end services via a web interface to manage data such as members, scout masters, activities, and activity submissions.

Requests from users are processed by the back-end service, which is developed using the Next.js framework (1) and functions as the REST API server. For security and access control, the system integrates NextAuth (2), which supports stateless authentication using JSON Web Token (JWT). Upon receiving a request and passing through authentication and authorization checks, the back-end service performs read/write operations on Supabase (3), which provides the PostgreSQL database. The interaction between the back-end and the database is facilitated by Prisma ORM (4), allowing for easy database queries and seamless integration with TypeScript.

To meet the system's requirements, the database is designed with ten tables that store information from various entities, illustrated in Fig. 3. This design includes data for the gugus depan, kwartir ranting, kwartir cabang, member and scout master management, rank history, activity reporting, as well as NTA submissions and verifications. The database is built using PostgreSQL, a powerful relational database management system, to ensure data integrity, scalability, and efficient querying. This design aims to ensure that the system development process is more efficient and structured.

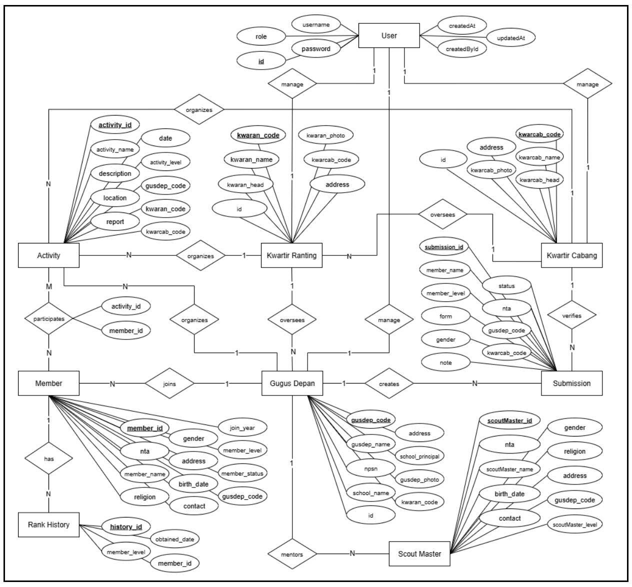


Fig. 3. ERD of Mataram Scout Information System

A. Product Backlog

In Scrum, the product backlog is a dynamic list of prioritized system requirements maintained by the product owner. It outlines the project scope and guides sprint planning. The backlog is continuously updated to reflect shifting user needs and sprint review outcomes, ensuring development remains aligned with project goals.

TABLE I. PRODUCT BACKLOG

Domain	Feature Description	Priority
Registration	A feature that allows admins to create a user account within the system.	High
User Management	A feature that allows users to manage other users accounts within their respective administrative regions.	High
Authentication	A feature that facilitates user authentication by verifying credentials for system access.	High
Member	A feature that allows gugus depan users to manage member data.	High
Scout Master	A feature that allows gugus depan users to manage scout master data.	High
Activity	A feature that allows users to manage activity data.	High
NTA Submission	A feature that allows gugus depan users submit NTA requests.	Medium
Submission Verification	A feature that allows kwartir cabang users to verify NTA submissions.	Medium
Dashboard	A feature that allows users to view the development of regional scouting organization through visualized data charts.	High
Activity History	A feature that allows users to view a list of activities in which a member has participated.	High
Rank History	A feature that allows user to record and view a member's rank progression, including the rank achieved and its date.	High

Following the product backlog, it is important to note that the addition of the "rank history" and “activity history” features resulted from feedback and requests from stakeholders during the sprint review session of sprint 3. During this review, stakeholders expressed the need for the system to display each member's rank progression and past activities to support more transparent and comprehensive data monitoring. This feedback was documented using project management tool and discussed further in the subsequent sprint planning meeting. Based on the team's evaluation of priority and feasibility, these features were officially added to the sprint 4 backlog. This reflects the iterative nature of Scrum, where stakeholder input is continuously gathered and translated into actionable tasks to enhance system functionality. The product backlog in this research specifically focuses on back-end development tasks, as this study centers on the development of the REST API component of the system.

B. Sprint Planning

Sprint planning is a phase in which the developer reviews the product backlog to define the scope of work for the upcoming sprint. The features and requirements are selected based on the project's current priorities and then organized into a sprint backlog. Each selected item is broken down into smaller tasks with estimated durations. The sprint backlog serves as a roadmap, guiding the execution of tasks throughout the sprint to ensure steady progress toward meeting the system's development goals. Although some features, such as those in sprint 5, have medium priority, they are still essential to the system's completeness and functionality, and are therefore implemented accordingly. Additionally, sprint 6 was conducted following academic review, focusing on implementing improvements and adjustments based on evaluation feedback.

TABLE II. SPRINT BACKLOG

Domain	Description	Priority & Duration
Sprint 1		
Registration	Admin registers an account by filling in the required form fields.	High, 2 Day
Authentication	- Gugus depan, kwartir ranting, and kwartir cabang users receive a username and password, registered by either admin or the user in charge of their area. - Users log in and out using valid credentials.	High, 2 Days
User Management	- Admin can manage accounts for gugus depan, kwartir ranting, and kwartir cabang users. - Kwartir cabang users can manage accounts for kwartir ranting users. - Kwartir ranting users can manage accounts for gugus depan users.	High, 3 Days
Sprint 2		
Member	- Gugus depan users can manage member data within their area. - Users can view members of each gugus depan.	High, 5 Days
Scout Master	- Gugus depan users can manage scout master within their area. - Users can view scout master of each gugus depan.	High, 3 Days
Sprint 3		
Activity	- Users can manage activities they have participated in or organized. - Users can view activity reports from their respective region.	High, 5 Days
Sprint 4		
Activity History	Users can view a list of activities a member has participated in.	High, 2 Day
Rank History	- Gugus depan users can manage members rank history. The most recent rank, based on the latest recorded date is displayed in the member's history. - Users can view member's rank history.	High, 2 Days
Dashboard	Users can view data visualizations of their scout organization's progress through charts.	High, 5 Days
Sprint 5		
NTA Submission	Gugus depan users can add, update, and delete NTA submissions.	Medium, 3 Days
Submission Verification	Users with the kwartir cabang role can verify NTA submissions.	Medium, 3 Days
Sprint 6		
User Management	Users can edit the accounts they have previously created.	High, 1 Day
Member & Scout Master	Add new attributes (contact) to the member and scout master models.	High, 1 Day
Submission Verification	Submissions that are approved will be directly registered as members, while rejected submissions will receive a correction note.	High, 2 Day

C. Sprint

The sprint backlog defines a five-sprint plan, with each sprint lasting one to four weeks. Throughout these sprints, features are developed and assessed according to predefined functional standards, ensuring systematic progress and alignment with project goals. The following table provides a comprehensive overview of the API endpoints developed during the sprints, detailing their functionality.

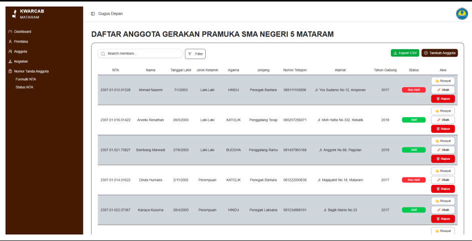
TABLE III. ENDPOINTS

Code	Method	Endpoint	Description
Sprint 1			
E001	POST	/auth/register	Allow admins to register.
E002	POST	/user/admin	Allow admins to create user accounts for gugus depan, kwartir ranting, and kwartir cabang users.
E003	GET	/user/admin	Allow admins to view the list of accounts they have created.
E004	DELETE	/user/admin/:id	Allow admin to delete user accounts.
E005	POST	/user/account	Allow users to create accounts for users within their respective regions.
E006	GET	/user/ account	Allow users to view the list of accounts they have created.
E007	DELETE	/user/ account /:id	Allow users to delete user accounts.
Sprint 2			

Code	Method	Endpoint	Description
E008	GET	/anggota	Allow viewing all members.
E009	POST	/anggota	Allow adding member.
E010	PATCH	/anggota/:id	Allow editing member.
E011	DELETE	/anggota/:id	Allow deleting member.
E012	GET	/pembina	Allow viewing all scout masters.
E013	POST	/pembina	Allow adding scout master.
E014	PATCH	/pembina/:id	Allow editing scout master.
E015	DELETE	/pembina/:id	Allow deleting scout master.
Sprint 3			
E016	GET	/kegiatan	Allow viewing all activities.
E017	GET	/kegiatan/:id	Allow viewing detailed activity.
E018	POST	/kegiatan	Allow adding activity.
E019	PATCH	/kegiatan/:id	Allow editing activity.
E020	DELETE	/kegiatan/:id	Allow deleting activity.
Sprint 4			
E021	GET	/anggota/:id /riwayat-kegiatan	Allow viewing a member’s activity history.
E022	GET	/anggota/:id /riwayat-jenjang	Allow viewing a member’s rank history.
E023	POST	/anggota/:id /riwayat-jenjang	Allow adding a new rank record to a member’s history.
E024	DELETE	/anggota/:id /riwayat-jenjang /:riwayatId	Allow deleting rank records from a member’s history.
E025	GET	/dashboard /anggotaByGender	Allow the chart to display member data based on gender.
E026	GET	/dashboard /anggotaByJenjang	Allow the chart to display member data based on rank.
E027	GET	/dashboard /anggotaByYear	Allow the chart to display total members per year.
E028	GET	/dashboard /gusdepByKwaran	Allows the chart to display total gugus depan in each kwartir ranting.
E029	GET	/dashboard /jenjangPerKwaran	Allow the chart to display total members per rank in each kwartir ranting.
E030	GET	/dashboard /kegiatanPerJenjang	Allow the chart to display total activities based on rank.
E031	GET	/dashboard /totalAnggota	Allow the chart to display total members.
E032	GET	/dashboard /totalKegiatan	Allow the chart to display total activities.
Sprint 5			
E033	GET	/ajuan	Allow viewing all submissions.
E034	POST	/ajuan	Allow adding submission.
E035	PATCH	/ajuan/:id	Allow editing submission.
E036	PATCH	/ajuan/:id/status	Allow kwartir cabang users to verify submission.
E037	DELETE	/ajuan/:id	Allow deleting submission.

Sprint 1 of Development. The development focuses on registration, authentication, and user management domain. The /auth/register endpoint enables admin to create their own accounts. In addition, the /user/admin and /user/account endpoints allow users to create user accounts for their respective regions. Authentication is handled using NextAuth, configured with a credentials-based provider and a stateless JWT strategy. Upon successful login, the system generates a JWT that securely encodes user-specific data, including the user's role and region codes (e.g., kwartir cabang, kwartir ranting, and gugus depan). This token is used to authorize access to protected endpoints, enabling role-based access control without relying on server-side sessions.

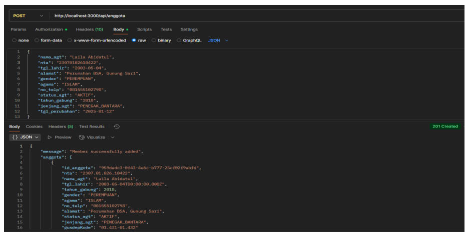
Sprint 2 of Development. The development focuses on managing member and scout master data. The /anggota and /pembina endpoints support full CRUD operations. These endpoints allow gugus depan users to manage their member and scout master records, while allowing kwartir cabang and kwartir ranting users to access the latest updates in real time, ensuring synchronized and consistent data across all administrative levels.



DAFTAR ANGGOTA GERAKAN PRAMUKA SMA NEGERI 5 MATARAM

NTA	Nama	Tanggal Lahir	Jenis Kelamin	Agama	Jenjang	Nomor Telepon	Alamat	Tahun Gabung	Status	Aksi
2307.01.012.01328	Ahmad Nazemi	7/1/2003	Laki-Laki	HINDU	Penegak Bantara	085111102856	Jl. Yos Sudarso No.12, Ampenan	2017	Non Aktif	Riwayat Ubah Hapus
2307.01.016.01422	Arvelio Renathan	28/5/2003	Laki-Laki	KATOLIK	Penggalang Terap	085257256071	Jl. Moh Hatta No.332, Kekalik	2018	Aktif	Riwayat Ubah Hapus
2307.01.021.73827	Bambang Marwadi	27/6/2003	Laki-Laki	BUDDHA	Penggalang Ramu	081457963168	Jl. Anggrek No.88, Pagutan	2019	Aktif	Riwayat Ubah Hapus
2307.01.014.01622	Dinda Humaira	2/11/2002	Perempuan	KATOLIK	Penegak Bantara	081222593638	Jl. Majapahit No.18, Mataram	2017	Non Aktif	Riwayat Ubah Hapus
2307.01.022.07367	Kanaya Kusuma	28/4/2003	Perempuan	HINDU	Penegak Laksana	081234888101	Jl. Bagik Manis No.23	2017	Aktif	Riwayat Ubah Hapus

Fig. 4. Member Page Integration Result



```
http://localhost:3000/api/anggota
```

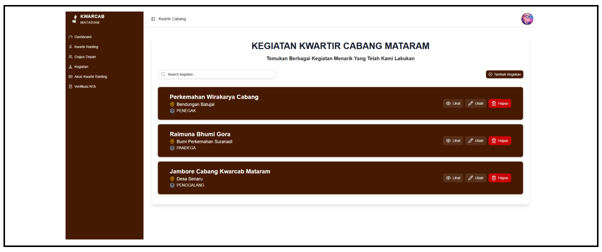
```
{
    "nama_agt": "Laila Abidatul",
    "nta": "23070102610422",
    "tgl_lahir": "2003-05-04",
    "alamat": "Perumahan BSA, Gunung Sari",
    "gender": "PEREMPUAN",
    "agama": "ISLAM",
    "no_telp": "081555102798",
    "status_agt": "AKTIF",
    "tahun_gabung": "2018",
    "jenjang_agt": "PENEGAK_BANTARA",
    "tgl_perubahan": "2025-01-12"
}
```

201 Created

```
{
    "message": "Member successfully added",
    "anggota": [
        {
            "id_anggota": "959dadc3-0f43-4e6c-b777-25cf02f9abfd",
            "nta": "2307.01.026.10422",
            "nama_agt": "Laila Abidatul",
            "tgl_lahir": "2003-05-04T00:00:00.000Z",
            "tahun_gabung": 2018,
            "gender": "PEREMPUAN",
            "agama": "ISLAM",
            "no_telp": "081555102798",
            "alamat": "Perumahan BSA, Gunung Sari",
            "status_agt": "AKTIF",
            "jenjang_agt": "PENEGAK_BANTARA",
            "gusdepKode": "01.431-01.432"
```

Fig. 5. Testing to Endpoint Member

Sprint 3 of Development. The development focuses on the activity management module. Endpoints under /kegiatan support full CRUD operations, enabling users to add, view, update, and delete activity records. Each activity entry requires a report document upload and the selection of at least one participant. These features ensure structured tracking of scout activities and support documentation and evaluation needs.



KEGIATAN KWARTIR CABANG MATARAM

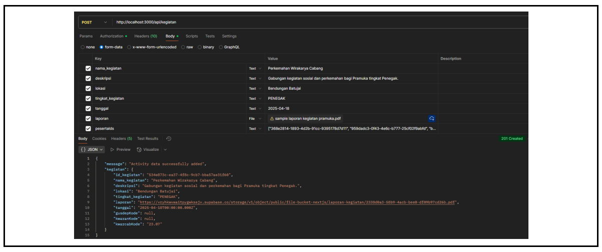
Temukan Berbagai Kegiatan Menarik Yang Telah Kami Lakukan

Perkemahan Wirakarya Cabang
Bendungan Batujai
PENEGAK

Raimuna Bhumi Gora
Bumi Perkemahan Suranadi
PANDEGA

Jambore Cabang Kwarcab Mataram
Desa Senaru
PENGGALANG

Fig. 6. Activity Page Integration Result



POST http://localhost:3000/api/kegiatan

	Key		Value	Description
✓	nama_kegiatan	Text	Perkemahan Wirakarya Cabang	
✓	deskripsi	Text	Gabungan kegiatan sosial dan perkemahan bagi Pramuka tingkat Penegak.	
✓	lokasi	Text	Bendungan Batujai	
✓	tingkat_kegiatan	Text	PENEGAK	
✓	tanggal	Text	2025-04-18	
✓	laporan	File	sample laporan kegiatan pramuka.pdf	
✓	pesertaIds	Text	["368e2814-1893-4d2b-91cc-9395178d7d11", "959dadc3-0f43-4e6c-b777-25cf02f9abfd", "b...	

201 Created

```
{
    "message": "Activity data successfully added",
    "kegiatan": {
        "id_kegiatan": "534e873c-ea37-4f8c-9cb7-bba57ae31f60",
        "nama_kegiatan": "Perkemahan Wirakarya Cabang",
        "deskripsi": "Gabungan kegiatan sosial dan perkemahan bagi Pramuka tingkat Penegak.",
        "lokasi": "Bendungan Batujai",
        "tingkat_kegiatan": "PENEGAK",
        "laporan": "https://vryhkwvwaitpygwkssjv.supabase.co/storage/v1/object/public/file-bucket-nextjs/laporan-kegiatan/2338d0a3-58b9-4acb-bee0-df09b97cd26b.pdf",
        "tanggal": "2025-04-18T00:00:00.000Z",
        "gusdepKode": null,
        "kwaranKode": null,
        "kwarcabKode": "23.07"
    }
}
```

Fig. 7. Testing to Endpoint Activity

Sprint 4 of Development. The development focuses on the dashboard and, following stakeholder requests, includes additional features for member history tracking. Endpoints under /anggota/:id/riwayat-jenjang and /anggota/:id /riwayat-kegiatan were implemented to manage rank and activity histories. Rank history records a member's rank progression, with the latest entry shown as the current rank. Activity history is auto generated from participation data, ensuring accuracy without duplication. Meanwhile, the /dashboard endpoints provide statistical visualizations of member distribution by gender, rank, year, and region, helping users interpret organizational data through intuitive charts.

Sprint 5 of Development. The development focuses on submission management through the /ajuan endpoints. Gugus depan users can create, edit, or delete their own submissions, while kwartir cabang users are authorized to review and verify them. Once a submission is approved and the NTA field is filled, the originally uploaded form is automatically deleted to reduce storage usage and prevent data redundancy. This role-based mechanism enforces hierarchical accountability and supports a secure, verifiable approval process.

Sprint 6 of Development. This sprint was initiated in response to feedback from academic reviewers. The reviewers recommended the addition of an account editing feature within the user management module, allowing users to update accounts they had previously created. Furthermore, a new contact attribute was added to both the anggota and pembina data models, based on the operational need for scout units to reach individuals when necessary. The submission verification flow under the /ajuan endpoint was also refined. When a kwartir cabang user approves a submission and provides the NTA value, the requester is automatically registered as a new member and their data becomes accessible through the /anggota endpoint. Gugus depan user then only needs to complete any remaining member information.

Daily Scrum. Daily scrum meetings were held to monitor REST API development progress and immediately resolve any technical issues that arose during the sprint. These 15-minute sessions helped maintain coordination and kept back-end development aligned with user needs.

Sprint Review. At the end of each sprint, a sprint review was conducted involving the Scrum team, product owner, and stakeholders. These collaborative sessions served to inspect the delivered REST API increments and refine the product backlog based on feedback. In sprint 3, stakeholders suggested the addition of a member history feature to better reflect individual progression. Following internal discussion, the Scrum team agreed to include this request in the next sprint. Similarly, sprint 6 was initiated in response to feedback from academic reviewers, which called for account editing capabilities, the addition of contact attributes to member and scout master data models, and refinement of the submission verification flow to automatically register approved requesters as members. Other sprints, namely sprints 1, 2, 4, and 5, were successfully completed as planned, with all targeted functionalities meeting the acceptance criteria and fulfilling user requirements.

Sprint Retrospective. At the end of each sprint, a retrospective was held to reflect on the team's performance, evaluate challenges, and identify areas for improvement. In sprint 1, challenges were encountered in implementing role-based access control. This was resolved by exploring the documentation of Next.js and NextAuth, along with configuring JWT token properties to manage role-specific authorization. Sprint 2 proceeded smoothly without significant technical issues. All planned features were implemented according to the product backlog and met stakeholder expectations. In sprint 3, the primary challenge involved handling file uploads for activity reports. This resolved by exploring the documentation of the storage service in use, which improved understanding of secure and efficient file handling in a server-side environment. Sprint 4 introduced changes to the database schema to support the newly added member history feature. Schema adjustments were performed using Prisma ORM, followed by structured database migrations. This process underscored the importance of adaptable schema design to support evolving functional needs. Sprint 5 was completed without major impediments, with all functionalities aligned with the product backlog and approved by stakeholders. Sprint 6 initiated in response to academic reviewer feedback, involved multiple structural enhancements. These included the implementation of account editing functionality within the user management module, the addition of a contact attribute to both member and scout master models, and refinement of the submission verification flow. The challenges were resolved by adding a handler function for account updates using the PATCH method, reconfiguring the Prisma ORM models, and refining the /ajuan endpoint to automatically add approved requesters as new members. Despite these obstacles, all backlog items were completed and successfully integrated into the front-end, meeting the stakeholder's expectations.

D. *REST API Testing*

REST API testing is performed at the end of each sprint, after all related endpoints have been fully developed. The sprint continues only if the tests meet the requirements. The following are the test results for the REST API endpoints developed to support the Mataram Scout Information System. Each test scenario in TABLE IV corresponds to an endpoint identified by its code in TABLE III.

TABLE IV. Black-Box Testing Result

Code	Scenario	Result	Status
E001	Valid request	201 Created, message "User created successfully"	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
	Register with wrong username format	400 Bad Request, message "Username cannot contain spaces"	
	Register with wrong password format	400 Bad Request, message "Password must be at least 8 characters long, contain uppercase, lowercase letters, and numbers."	
	Register with duplicate username	409 Conflict, message "Username already taken"	
E002	Valid request	201 Created, message "User created successfully"	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
	Create user account with wrong username format	400 Bad Request, message "Username cannot contain spaces"	
	Create user account with wrong code format	400 Bad Request, message "Code cannot contain spaces"	
	Create user account with wrong password format	400 Bad Request, message "Password must be at least 8 characters long, contain uppercase, lowercase letters, and numbers."	
	Create user account with duplicate username	409 Conflict, message "Username already taken"	
	Create user account with duplicate code	409 Conflict, message "Code already taken"	
	Unauthorized user tries to create account	403 Forbidden, message "Unauthorized "	
E003	Retrieve accounts	200 OK, list of created accounts returned	Pass
E004	Delete existing account	200 OK, message "User deleted successfully"	Pass
E005	Valid request	201 Created, message "User created successfully"	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
	Create user account with wrong username format	400 Bad Request, message "Username cannot contain spaces"	
	Create user account with wrong code format	400 Bad Request, message "Code cannot contain spaces"	
	Create user account with wrong password format	400 Bad Request, message "Password must be at least 8 characters long, contain uppercase, lowercase letters, and numbers."	
	Create user account with duplicate username	409 Conflict, message "Username already taken"	
	Create user account with duplicate code/name	409 Conflict, message "Code or name already taken"	
	Unauthorized user tries to create account	403 Forbidden, message "Unauthorized"	
E006	Retrieve accounts	200 OK, list of created accounts returned	Pass
E007	Delete existing account	200 OK, message "User deleted successfully"	Pass
E008	Retrieve members	200 OK, list of members returned	Pass
E009	Valid request	201 Created, message "Member added successfully"	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
	Add member with duplicate NTA	409 Conflict, message "NTA already registered"	
E010	Valid request	200 OK, message "Member updated successfully"	Pass
	Edit member with duplicate NTA	409 Conflict, message "NTA already registered"	
E011	Delete existing member	200 OK, message "Member deleted successfully"	Pass
E012	Retrieve scout masters	200 OK, list of scout masters returned	Pass
E013	Valid request	201 Created, message "Scout master added successfully"	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
	Add scout master with duplicate NTA	409 Conflict, message "NTA already registered"	
E014	Valid request	200 OK, message "Scout master updated successfully"	Pass
	Edit scout master with duplicate NTA	409 Conflict, message "NTA already registered"	

E015	Delete existing scout master	200 OK, message "Scout master deleted successfully"	Pass
E016	Retrieve activities	200 OK, list of activities returned	Pass
E017	Retrieve detail activity	200 OK, detailed activity returned	Pass
E018	Valid request	201 Created, message “Activity added successfully”	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
	Add activity with too long description	400 Bad Request, message "The description is too long (max 300 words)"	
	Attempt to upload a non-PDF activity report file	400 Bad Request, message "Only PDF files are allowed"	
	Attempt to upload a file larger than 2MB	400 Bad Request, message "File size must be less than 2MB"	
	Add activty without selecting participant	400 Bad Request, message "At least one participant must be selected"	
E019	Valid request	200 OK, message “Activity updated successfully”	Pass
	Edit activity with too long description	400 Bad Request, message "The description is too long (max 300 words)"	
	Attempt to upload a non-PDF activity report file	400 Bad Request, message "Only PDF files are allowed"	
	Attempt to upload a file larger than 2MB	400 Bad Request, message "File size must be less than 2MB"	
	Edit activty without selecting participant	400 Bad Request, message "At least one participant must be selected"	
E020	Delete existing activity	200 OK, message "Activity deleted successfully"	Pass
E021	Retrieve member’s activity history	200 OK, member’s activity history returned	Pass
E022	Retrieve member’s rank history	200 OK, member’s rank history returned	Pass
E023	Valid request	201 Created, message “Rank history added successfully”	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
E024	Delete existing rank history	200 OK, message "Rank history deleted successfully"	Pass
E025-E032	Retrieve dashboard data based on parameters	200 OK, dashboard data returned	Pass
E033	Retrieve submissions	200 OK, list of submissions returned	Pass
E034	Valid request	201 Created, message “Submission added successfully”	Pass
	Missing required fields	400 Bad Request, message "All fields are required"	
	Attempt to upload a non-PDF submission file	400 Bad Request, message "Only PDF files are allowed"	
	Attempt to upload a file larger than 2MB	400 Bad Request, message "File size must be less than 2MB"	
E035	Valid request	200 OK, message “Submission updated successfully”	Pass
	Attempt to update submission without providing NTA value when the status is approved	400 Bad Request, message "Field 'NTA' is required if the status is 'DITERIMA'"	
E036	Valid request	200 OK, message “Submission updated successfully”	Pass
	Edit submission with too long note	400 Bad Request, message "The note is too long (max 300 words)"	
	Missing NTA value when the status is approved	400 Bad Request, message "Field 'NTA' is required"	
	Edit submission with wrong NTA format	400 Bad Request, message "NTA must be 14–16 digit numbers"	
	Edit submission with duplicate NTA	409 Conflict, message "NTA already registered"	
E037	Delete existing submission	200 OK, message “Submission deleted successfully"	Pass

As shown in TABLE IV, the tests were conducted to ensure each REST API endpoint correctly handles the expected inputs and returns appropriate responses, confirming the system functions as expected for both valid and invalid data scenarios. A User Acceptance Testing (UAT) session was held through a meeting with representatives of the Mataram Scout Movement to demonstrate the Scout Information System and gather feedback via a survey. The survey involved 23 respondents from various organizational levels (gugus depan, kwartir ranting, kwartir cabang). The system received an average feasibility rating of 81.74%, classified as "Good," indicating that the respondents perceive the system as efficient and suitable for improving data management compared to the previous manual workflows using Google Forms and Excel [14]. A more detailed explanation of the UAT process is provided in a separate study authored by another member of the development team, who was responsible for the front-end implementation. This positive evaluation highlights the system’s potential to address inefficiencies in data management within Kwartir Cabang Mataram.

V. Conclusion and Suggestion

This research developed a REST API for the Mataram Scout Information System using the Scrum method. A total of 25 endpoints were implemented, which follow

REST principles such as resource-oriented URL naming (e.g., /anggota, /pembina, /kegiatan, /ajuan, /dashboard), standardized HTTP methods (GET, POST, PATCH, DELETE), and stateless communication using JWT-based authentication via NextAuth. Consistent request-response formats and modular endpoint design supported clear separation of concerns and smooth front-end integration. These aspects indicate that the REST API was successfully developed making it scalable and maintainable.

The development process followed all stages of the Scrum method. The initial product backlog was based on detailed system requirements and was continuously updated in response to stakeholder feedback. Sprint planning ensured that each sprint had clear goals, realistic task estimates, and a focused scope. Sprint reviews allowed stakeholders to evaluate completed work, and their feedback directly led to additional features such as rank and activity history, which were not included in the initial backlog. Sprint retrospectives helped identify workflow improvements, which were applied in the following sprints. All planned backlog items were completed as scheduled, and no major delays occurred during development. This shows that Scrum was implemented and played a key role in organizing the development process.

The system's reliability was verified through black-box testing, which confirmed the correct functioning of the REST API endpoints across various valid and invalid input scenarios. This was further supported by a User Acceptance Testing (UAT) session involving 23 respondents from different scout organizational levels, resulting in a feasibility score of 81.74%, categorized as "Good". These results indicate that the system meets both technical and practical requirements, particularly by supporting real-time management of member data and reporting of activity data. For future research, developing a mobile application may enhance user experience and system flexibility in managing scouting data. Suggested features include offline data access, push notifications for activity updates, and mobile-friendly interfaces.

References

[1] Renilda Manis Da Gomez, Elviana Pona Rato, Donatus Datoq, Marwan Marwan, Wihelmus Sede, and Fitrah Hamzah, "Membangun Karakter Bangsa Melalui Kepramukaan," *Jurnal Ilmiah Dan Karya Mahasiswa*, vol. 3, no. 1, pp. 182–187, Feb. 2025, doi: 10.54066/jikma.v3i1.3015.

[2] L. Lutfiasin, "Sejarah Pembentukan Gerakan Pramuka Dan Pengaruhnya Dalam Dunia Pendidikan Islam," *Thawalib | Jurnal Kependidikan Islam*, vol. 2, no. 1, pp. 37–52, Apr. 2021, doi: 10.54150/thawalib.v2i1.19.

[3] *Undang-Undang Republik Indonesia Nomor 12 Tahun 2010 Tentang Gerakan Pramuka*. [Online]. Available: https://www.pramuka.or.id/files/document/UU-Gerakan-Pramuka.pdf

[4] S. U. Meshram, "Evolution of Modern Web Services: REST API with its Architecture and Design," *International Journal of Research in Engineering, Science and Management*, vol. 4, no. 7, pp. 83–86, 2021.

[5] A. C. Sassa, I. Alves De Almeida, T. Nakagomi, F. Pereira, and M. Silva De Oliveira, "Scrum: A Systematic Literature Review," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 14, no. 4, pp. 173–181, 2023, doi: 10.14569/IJACSA.2023.0140420.

[6] R. Noveandini, M. S. Wulandari, and A. Hakim, "Penerapan Metode Scrum pada Rancang Bangun Sistem Informasi Penjualan Toko Sepatu Rabbani Shoes," *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, vol. 7, no. 1, pp. 192–199, Jul. 2023.

[7] T. Wibowo and S. Salim, "Pengembangan Dan Implementasi Back End Website Sistem Laporan Keuangan Di Smk Multistudi High School Menggunakan Kerangka Kerja Agile Scrum," *NaCosPro: National Conference For Community Service Project*, vol. 4, no. 1, pp. 880–887, Aug. 2022, [Online]. Available: http://journal.uib.ac.id/index.php/nacospro

[8] S. B. Farchani, A. Kusuma, and N. Hermanto, "Implementasi REST API Dalam Pengembangan Back-End Inventory Peminjaman," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 10, no. 2, pp. 1404–1413, Jun. 2025, doi: 10.29100/jipi.v10i2.6249.

[9] R. Muhammad, G. Dennaya, and A. Zubaidi, "Pengembangan Back-End Pada Aplikasi Presensi dan Website Pendataan UMKM NTB Mall," *JBegaTI*, vol. 5, no. 2, pp. 170–181, Sep. 2024, [Online]. Available: http://begawe.unram.ac.id/index.php/JBTI/

[10] W. Ahmad Widodo, I. Ratna Indra Astutik, and N. Ariyanti, "Penerapan Framework NextJS Dan Prisma ORM Untuk Perluasan Marketplace Newborn Photography," *JOISIE (Journal Of Information Systems And Informatics Engineering)*, vol. 8, no. 2, Dec. 2024, doi: 10.35145/joisie.v8i2.4704.

[11] S. L. Mufreni *et al.*, "Sistemasi: Jurnal Sistem Informasi Development of Web-App Using Agile Scrum Method at PT. Stechoq Robotika Indonesia," *Sistemasi: Jurnal Sistem Informasi*, vol. 14, no. 2, pp. 917–931, 2025, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[12] I. R. D. Muhammad and I. V. Paputungan, "Development of Backend Server Based on REST API Architecture in E-Wallet Transfer System," *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 3, no. 2, pp. 79–87, Jan. 2024, doi: 10.20885/snati.v3.i2.35.

[13] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, and Q. A. Giansyah, "Pengujian Black Box dan White Box Sistem Informasi Parkir Berbasis Web," *Jurnal Testing dan Implementasi Sistem Informasi*, vol. 1, no. 1, pp. 1–16, 2023.

[14] Wulandari, Nofiyani, and H. Hasugian, "User Acceptance Testing (Uat) Pada Electronic Data Preprocessing Guna Mengetahui Kualitas Sistem," *Jmik (Jurnal Mahasiswa Ilmu Komputer)*, vol. 4, no. 1, pp. 20–27, Mar. 2023.