Jurnal Agriment 10(1):27-40, 2025

doi.org/10.51967/jurnalagriment.v10i1.3308

Development of a Microservices-Based Geographic Information System for Mapping Flora and Fauna in Kutai National Park

Rasni Alex[1]*, Nur Maulida Sari[2], Aulia Khoirunnita[3], Indah Fitri Astuti[4], Dedy Cahyadi[5], Awang Kridalaksana[6], Ida Maratul Khamidah[7]

[1,3,4,5,6]Informatic, Mulawarman University, Indonesia
[2]Forest Product Processing, State Polytechnic Agricultural of Samarinda, Indonesia
[7]Software Engineering Technology, State Polytechnic Agricultural of Samarinda, Indonesia
***Corresponding email: rasnialex@ft.unmul.ac.id**

Submitted: 2025-05-06; Accepted: 2025-06-30; Published: 2025-06-30

ABSTRACT

Kutai National Park serves as a vital habitat for diverse flora and fauna, yet the park faces challenges in achieving comprehensive and efficient mapping of its biodiversity. In alignment with the 2019–2029 Long-Term Management Plan, which prioritizes completing a robust mapping system, this study proposes the development of a Geographic Information System (GIS) based on a microservices architecture. The system aims to streamline data management and support area conservation efforts. Utilizing the Extreme Programming methodology, the project leverages React JS for the web interface, React Native for mobile applications, and MongoDB for database management. System testing incorporates Postman for microservices, Selenium for web applications, and Black-Box testing for mobile apps. Feasibility evaluations use Equivalence Partitioning and the Mean Opinion Score (MOS) method, reflecting high acceptance among staff (71% Strongly Agree, 29% Agree) and visitors (47.9% Strongly Agree, 43% Agree, 3.8% Neutral). The results demonstrate the system's potential to enhance biodiversity management while fostering stakeholder engagement in conservation initiatives.

Keywords: Biodiversity Mapping, Geographic Information System, Microservices Architecture, Conservation Efforts.

INTRODUCTION

Indonesia is a tropical country, where there are only two seasons: the dry season and the rainy season. Due to this, forests have become essential areas to have. The advantage of having forests is their significant impact in reducing air pollution, making the air healthier for the surrounding living beings to breathe (Darmayantie, 2020; Habiburrahman, 2020). Kutai National Park is one of the Forest Parks in Indonesia and the only one located in East Kalimantan.

Kutai National Park is rich in natural resources, including flora such as mahogany (Swietenia mahagoni), kesambi (Scleria oleosa), klokos (Syzygium javanica), and others. Meanwhile, the fauna includes deer (Cervus timorensis), macaques (Macaca fascicularis), lutungs (Trachypithecus auratus), and others.

Because Kutai National Park hosts a variety of flora and fauna, it has significant potential as center for education and research on wildlife and vegetation in Kalimantan. Based on the researcher’s interview with Kutai National Park officials, the flora and fauna mapping system has not yet fully utilized technology (Chang,2019; Pahl & Jamshidi, 2020), making the information difficult for the public to access and rarely updated. Consequently, the utilization of flora and fauna as an educational and research center is still not optimal. However, Kutai National Park acknowledges this issue and, in its Long-Term Development Plan for 2019-2029, has included an activity plan aimed at increasing visitor numbers and local revenue for

Kutai National Park (Richards, 2017). One of the plans is to increase the diversity of flora and fauna in the area, with the success indicator being the compilation of data on the types, numbers, and diversity of flora and fauna in Kutai National Park. To support this goal, this study aims to develop a Geographic Information System (GIS) for mapping flora and fauna in Kutai National Park using a microservices architecture (Lewis & Flower, 2021; Lokapitasari et al, 2020; Chandra et al, 2020). A GIS is an information system that processes and stores data in the form of geographic information of an area (Budi & Bachtiar, 2018). The application of GIS in mapping has been widely conducted and proven effective for mapping areas, making it suitable for mapping flora and fauna in Kutai National Park, which currently lacks a GIS-based mapping system (Suryotrisongko, 2017). So far, information systems have been developed using a monolithic architecture, where all components are in one place, making future development difficult (Darmayantie, 2020). Monolithic architecture is inflexible because it continues to rely on previously used technologies (Park, 2018). As a result, system owners find it challenging to find developers who truly understand the technology in use (Albakri & Al-Emran, 2017). The solution is to adopt a microservices architecture, which has recently gained popularity (Richards, 2017; Bastiaansen, 2021). Microservices are small applications that can be implemented independently, scaled independently, tested independently, and have a single responsibility (Kraak, 2018). The system to be developed for Kutai National Park is a GIS with a microservices architecture, expected to provide information about the flora and fauna in the area (Silverman, 2019; Chen , 2020; Johnson, 2020;). The use of microservices architecture is also expected to facilitate the scaling of the GIS application if future improvements are needed (Nissen, 2015; Hart, 2017). It is hoped that this system will have a positive impact on the management of Kutai National Park and simplify research efforts for academics studying the flora and fauna in Kutai National Park.

METHODOLOGY

Tools and materials

The tools and materials used in the research are implemented in the form of software, hardware, data, and information required during the research process (Singh, 2021).

Research tools

The following are the tools used in the process of developing the flora and fauna mapping system at Kutai National Park:

1. ROG GL-552VW Laptop,
2. Windows 10 Pro Operating System,
3. Visual Studio,
4. ReactJS,
5. React Native,
6. MongoDB,
7. NodeJS,
8. ExpressJS,
9. Xiaomi Redmi Note 10.

Research materials

The materials used in this research to develop the flora and fauna mapping system were sourced from books and existing research journals (Habiburrahman, 2020; Kato & Liu, 2020). In addition, there are flora and fauna data that serve as initial sample data for the application, obtained through direct observation at Kutai National Park on June 27, 2021. This data serves as the primary material for this research, including.

doi.org/10.51967/jurnalagriment.v10i1.3308

Table 1. Observation Result

No	Name	Location
1	Pacing Tawar	-8.498812, 116.293577
2	Bamboo	-8.515478, 116.281717
3	Breadfruit	-8.499690 116.293513
4	Banana Milk	-8.499589,116.293392
5	Walet Linci	-8.4495863, 116.293755
6	Kra monkey	-8.499589, 116.293392

Table 1 shows, the observation results obtained from direct field studies conducted at Kutai National Park. It lists six different flora and fauna species, along with their respective geographic coordinates recorded during the observation. These data points include plant species such as Pacing Tawar, Bamboo, Breadfruit, Banana Milk, and Walet Linci, as well as a fauna species identified as Kra monkey. The precise latitude and longitude values provided in the table ensure accurate geolocation for each observation, which supports further analysis and mapping processes for this research.

Research flow

Figure 1 illustrates the research method flow diagram, depicting a sequential process that includes literature study, data analysis and planning, system requirements identification, design, application creation, testing, and documentation. The diagram also highlights feedback loops for addressing new needs, bugs, and relationships to ensure iterative development and improvement (Morris, 2020; Martinez & Fadely, 2019).

Application design

The application design is created using UML to represent the activities, data, interacting components, and other elements. The diagrams include the use case diagram, service decomposition, and class diagram (Choi & Lee, 2018).

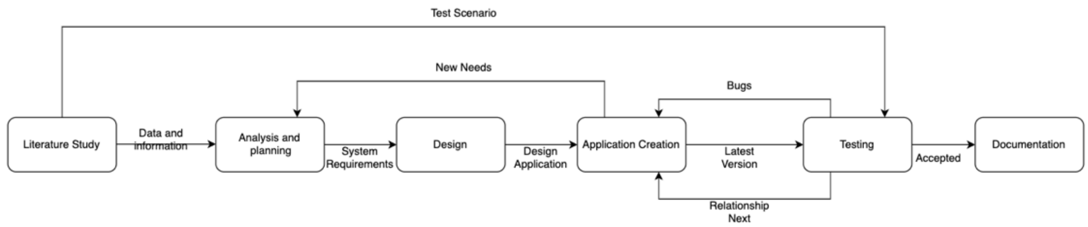


Figure 1. Research Method Flow Diagram

Figure 2 shows, the use case diagram designed to model the interactions between users and the application. It depicts two primary actors, User and Admin, who perform various tasks related to flora and fauna data management. The User can view flora and fauna details, their locations, and edit user data, while the admin has additional permissions such as managing flora and fauna data, logging out, and viewing profiles. Furthermore, the Admin can add, edit, and delete data for both flora and fauna, manage maps, and input coordinates. This diagram illustrates the system's functionality and highlights the roles and responsibilities of each actor, ensuring clarity in application design and workflow.

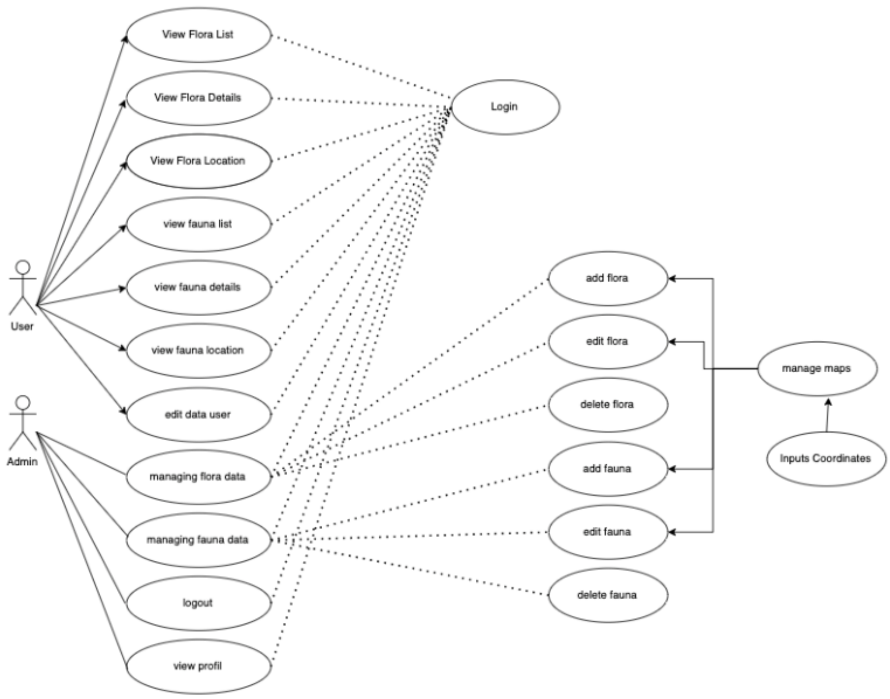


Figure 2. Use Case

Services Decomposition

Figure 3 shows, the service decomposition of the application, illustrating the interaction between different components and services. It highlights three primary client interfaces: the User Mobile App (React Native), User Web App (React JS), and Admin Web App (React JS). These interfaces communicate with the backend through REST APIs, which act as intermediaries between the clients and services (Zhang & Liu, 2018).

The services are divided into three categories—User Service (Express JS), Organisme Service (React JS), and Maps Service (Node JS) (Thomas & Ma, 2019). Each service connects to its corresponding MongoDB database, namely User DB, Organisme DB, and Peta DB, to store and manage data (Yang & Zeng, 2020). Additionally, the diagram shows integration with Firebase Storage for image handling, ensuring scalability and efficient storage management (Kwon & Kim, 2019). This design demonstrates a modular architecture that supports flexibility and ease of maintenance.

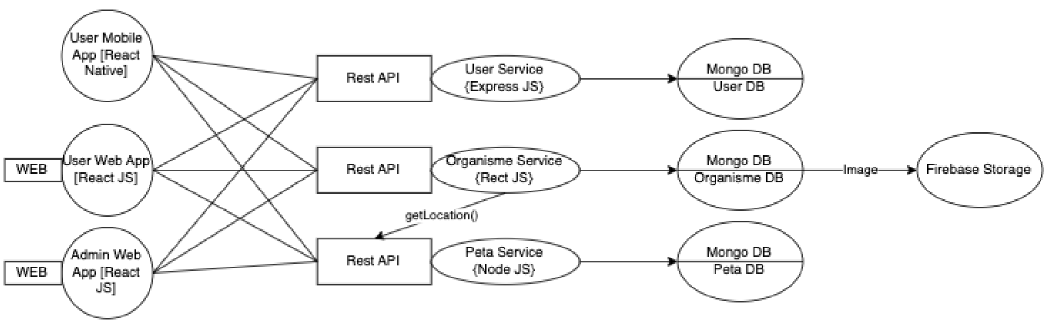


Figure 3. Services Decomposition

doi.org/10.51967/jurnalagriment.v10i1.3308

Class Diagram

Figure 4 shows, the class diagram of a system that implements a Microservices Architecture Model to manage users, organisms, and maps. Each service is modular and focuses on specific functionality. The Login_Controller handles authentication by interacting with the Login_M service, which manages user credentials. The User_Controller manages user-related operations, such as adding, editing, and deleting user data, stored in the User_M service. Interfaces like Page_User and Registration_Page facilitate user interactions. The Organisme_Controller manages organism data through Organisme_M, which handles attributes like name, description, and status, with Page_Fauna providing the interface. Similarly, the Map_Controller manages geospatial data via Map_M, including attributes like coordinates and flora IDs, with Page_Map as its interface. This architecture ensures scalability and flexibility by decoupling services and enabling independent development and deployment.

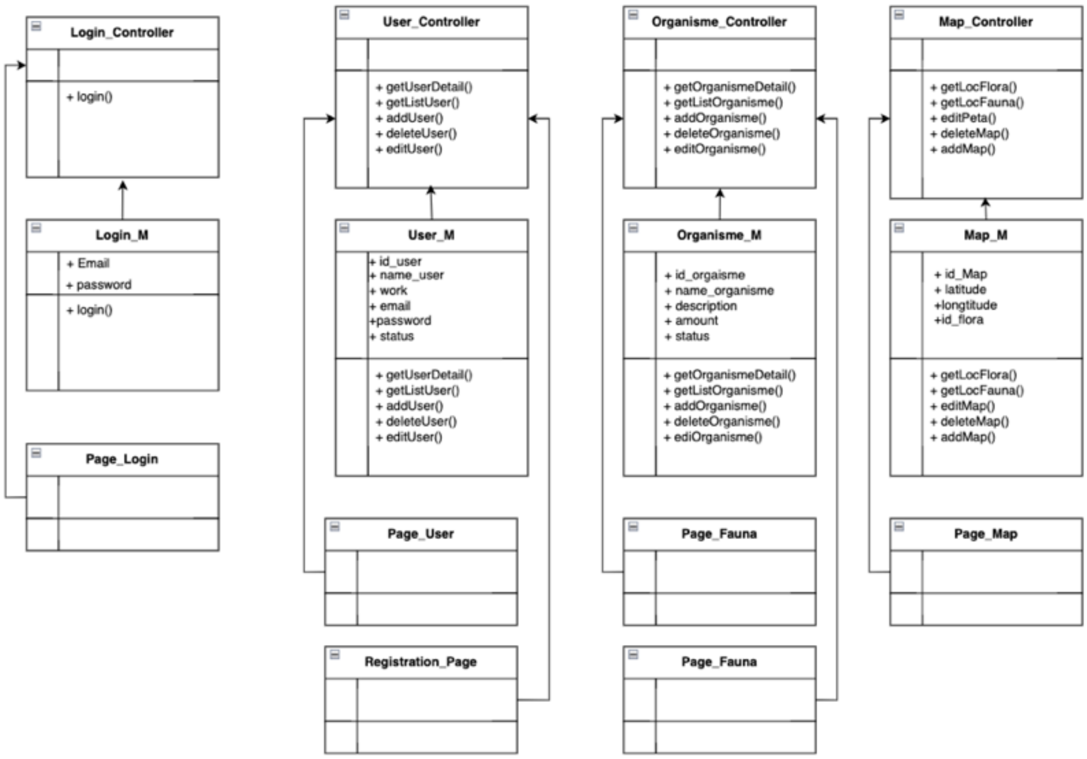


Figure 4. Class Diagram

RESULT AND DISCUSSION

Application interface

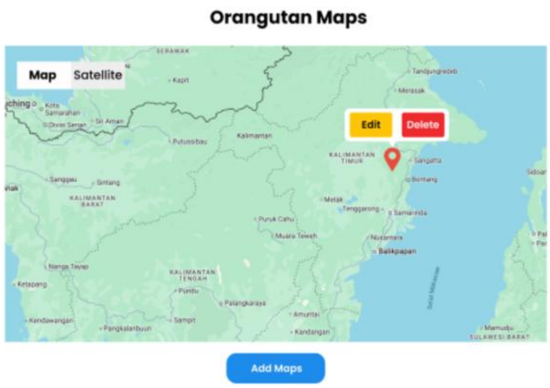
This stage presents the results of the user interface implementation applied in the system development. The user interface consists of three types: web users, mobile users, and web admins

Web application interface

Figure 5 shows the web application interface for the Kutai National Park system, featuring three main sections. Figure 5 (a) shows, the homepage displays the latest fauna and flora in a card layout with images and "View Details" buttons for more information. Figure 5 (b) shows, the fauna details page provides comprehensive descriptions of specific fauna, such as orangutans, including habitat and classification details, with a "View Classification" button for further exploration. Figure 5 (c) shows, the map interface allows users to view fauna locations on a map or satellite view, with functionalities to edit, delete, or add new map data. This interface integrates visual, informational, and interactive elements to enhance usability.

(a)

(b)



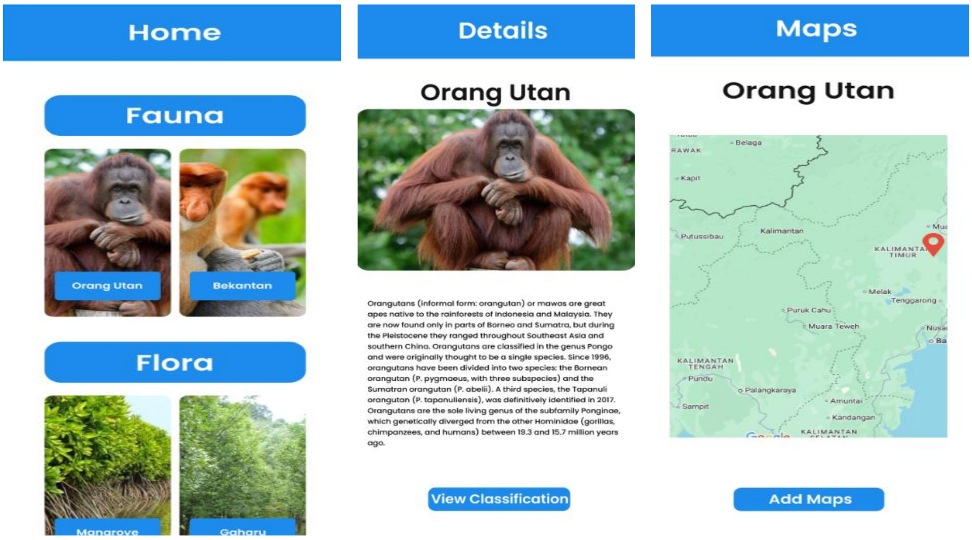
(c)

Figure 5. Web interface (a) Homepage interface, Fauna details (b) Fauna details, Map details (c) Map details

Mobile application interface

Figure 6 shows, the user interface of a mobile-based application designed to explore flora and fauna, consisting of three main sections. Figure 6 (a) shows, the Homepage provides categories for "Fauna" and "Flora," each with visual thumbnails for easy navigation. Figure 6 (b) shows, the Details Page displays detailed information about a selected species, such as the "Orang Utan" including an image, a description, and a "View Classification" button for further exploration. Figure 6 (c) shows, the Maps Page presents a map view highlighting the habitat or distribution of the selected species, with an "Add Maps" button for integrating additional mapping features. This interface offers an intuitive and visually engaging way to access biodiversity information.

doi.org/10.51967/jurnalagriment.v10i1.3308



(a) (b) (c)

Figure 6. Mobile interface (a) Homepage, Flora detail (b) Flora detail, Map details (c) Map details

Services and application

Figure 7 shows, the architecture of the Geographic Information System (GIS) application, which is divided into three main components: organism services, map services, and user services. The system utilizes a central JWT for API Gateway that connects the Mobile Apps and Web Apps to backend services (Vargas & Sanchez, 2020). Each service—Organism Service, Map Service, and User Service—is powered by a REST APIs built with frameworks like Express.js or Node.js, connected to MongoDB databases for data storage. The Organism Service manages organism data and integrates with a cloud storage service for image handling, while the Map Service connects to Google Maps APIs for geographic visualization. The User Service handles user-related data. This modular architecture ensures efficient and scalable GIS application functionality.

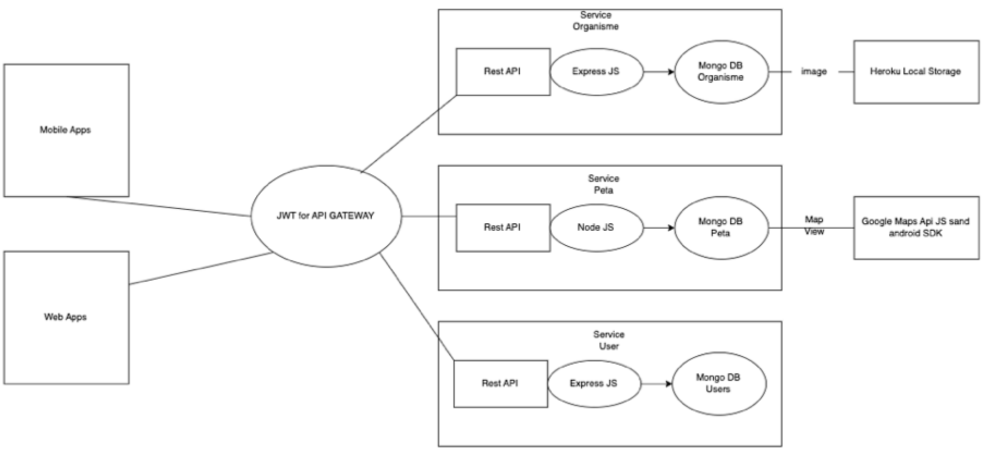


Figure 7. Services and application

Micro services Testing

At this stage, testing is conducted on the microservices that have been developed. The microservices are divided into three services: organism service, map service, and user service. The testing scheme is based on the endpoints of each service.

Table 2 shows, the results of testing for the endpoints associated with three microservices: the user service, map service, and organism service. Each endpoint was tested using different HTTP methods, including POST, GET, DELETE, and PUT, to ensure proper functionality. The status column indicates that all endpoints responded with "OK," signifying successful execution of the requests. This testing verifies the reliability and correctness of the microservices by confirming that each endpoint operates as expected under its respective HTTP method.

Table 2. Testing Services

End Point	***Method***	***Status***
https://nationalkutaiservices.herokuapp.com/user/login	POST	OK
https://nationalkutaiservices.herokuapp.com/user/me	GET	OK
https://nationalkutaiservices.herokuapp.com/peta	GET	OK
https://nationalkutaiservices.herokuapp.com/peta/ 24j4g5432kk6b3364ik	DELETE	OK
https://nationalkutaiservices.herokuapp.com/peta	POST	OK
https://nationalkutaiservices.herokuapp.com/organisme	GET	OK
https://nationalkutaiservices.herokuapp.com/organisme/ 7bj4ad7f9f9vf868ff432h	PUT	OK

The implementation of the question layout is a Popup view that raises questions and there are 2 options can be seen in figure 8, and questions will arise when picking up a box at each level. This question can be from material books or information on each type of animal. The player must answer the questions that arise by selecting multiple answers.

Test result Selenium IDE

The testing was conducted using Selenium IDE based on the recorded results from previous actions. This testing is performed according to the Use Case Diagram that has been created to assess the success rate of each defined feature. Below is the table of testing results using the Selenium IDE tool that has been carried out. Table 3 shows, the results of feature testing conducted using Selenium IDE to validate the functionality of the application's features. Each feature was tested twice, and all tests were successful, as indicated by the absence of failed attempts in the "Failed" column. This comprehensive testing covered critical operations such as login, register, adding and editing flora and maps, deleting entries, viewing profiles, and logout. The results confirm the reliability and proper implementation of the tested features, with a 100% success rate for all scenarios.

doi.org/10.51967/jurnalagriment.v10i1.3308

Table 3. Testing Selenium IDE

Features Tested	Number Testing	Successful	Failed
Login	2	2	0
Register	2	2	0
Add Flora	2	2	0
Add Map	2	2	0
Delete Map	2	2	0
Edit Map	2	2	0
Edit Flora	2	2	0
Delete Flora	2	2	0
View Profile	2	2	0
Logout	2	2	0

Black Box Equivalence Partitioning Testing on the Mobile Application

The Black-Box method focuses more on testing the program's functionality. The Black-Box technique used in this testing is Equivalence Partitioning (EP), which divides inputs into several groups based on their functionality. Below are the results of the Equivalence Partitioning (EP) testing.

Table 4 shows, the results of Black-Box Equivalence Partitioning (EP) testing for the login functionality of the mobile application. The test scenarios include logging in with valid credentials and attempting to log in with invalid credentials. The system successfully fulfilled expectations in both cases: it logged in correctly when valid credentials were provided and displayed an appropriate error message when incorrect credentials were used. All results align with the expected outcomes, indicating proper handling of login scenarios.

Table 4. Test result from form login

Testing Description	Expected Result	Result Testing
Filling email with “admin@gmail.com” and fill password with “admin123”	System successful login and leading to view home	Succeed
Filing username and password that not registered by database	System failed login with displaying pop up with information username / incorrect password	Succeed

Table 5 shows, the results of Black-Box Equivalence Partitioning (EP) testing for the registration functionality. The tests evaluated various scenarios, including complete registration, partial registration, and filling in only specific fields like email. In each case, the system's behavior matched the expected outcomes, such as successfully completing registration when all required fields were filled or providing appropriate error messages for incomplete forms. This demonstrates that the registration process is robust and handles various input conditions effectively.

Table 5. Test Result Form Register

Test Description	Result that Expected	Result Testing
Filling in all registration from completely with such as Name, Job, Email, and Password	I succeeded registered to in database and leading to display homepage	Succeed
Filling in the email column on the form	The system failed login with	Succeed
Registration not complete	Displaying pop up with explanation	Succeed

MOS Testing

After conducting the MOS (Mean Opinion Score) testing, which is a method for measuring the performance and quality of the developed system through questionnaires, the responses received were calculated to draw conclusions about the system's marketability. Two types of questionnaires were distributed: one for the staff of Balai Kutai National Park, consisting of 3 administrators, and the other for 40 tourists, who were university students from various institutions in the Samarinda City area. The staff questionnaires were distributed in physical form, while the tourist questionnaires were distributed through a Google form. Below is a list of questions for both types of questionnaires:

Admin use

Figure 8 shows, the distribution of responses from the three staff members at Kutai National Park across eight questions provided in the questionnaire. The responses are categorized into Strongly Agree, Agree, Neutral, Disagree, and Strongly Disagree. The chart indicates that most respondents selected "Strongly Agree," followed by "Agree," demonstrating a generally positive evaluation.

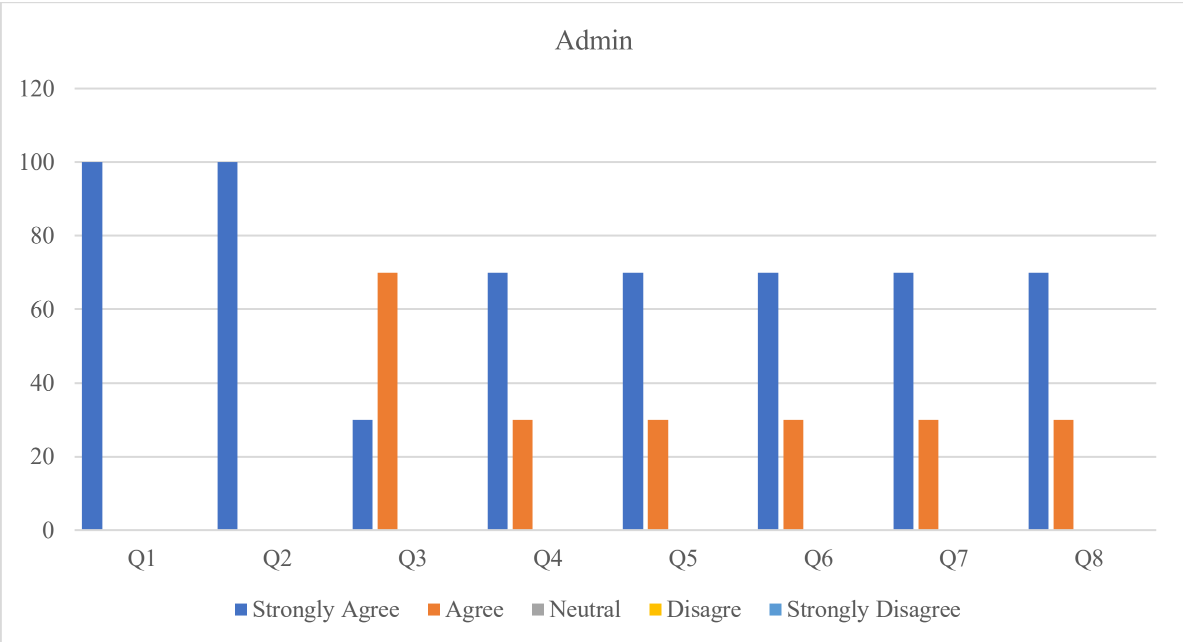


Figure 8. Admin Questionnaire Percentage Chart

doi.org/10.51967/jurnalagriment.v10i1.3308

Respondents were asked to fill out the questionnaire using the following answer choices:

Q = Question

The results of the questionnaire testing conducted with the 3 staff/admin members of Kutai National Park showed that the average responses were Strongly Agree: 70.3% and Agree: 20.6%

General use

Figure 9 shows, the questionnaire results from 40 users across eight questions, categorized into Strongly Agree, Agree, Neutral, Disagree, and Strongly Disagree. The chart illustrates that most users selected "Strongly Agree" and "Agree," indicating a positive reception overall.

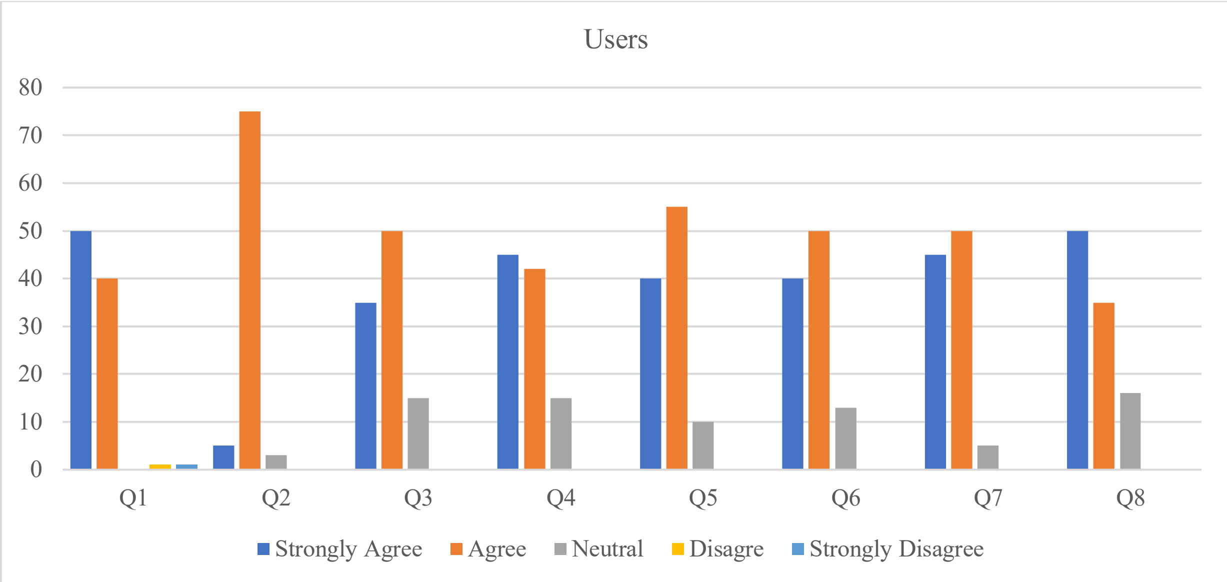


Figure 9. Questionnaire Percentage Chart

Respondents were asked to fill out the questionnaire using the following answer choices:

Q = Question

The results of the questionnaire testing conducted with 40 users showed the following average responses:

Strongly Agree: 47.9%, Agree: 43% and Neutral: 3.8%

Implementation

Table 6 shows, illustrates the benefits of implementing microservices, along with their corresponding implementation status and descriptions. The table highlights scalability, adaptability, deploy ability, and maintainability as implemented features, while noting the absence of team independence due to minimal human resource allocation for deployment processes.

Table 6. Implementation Benefit

Benefit	Implementation / No	Description
Service scalable in independent. (Scalable)	Implementation	Only one object that can be configured in each service it. So, for development can be further using specific service according to what is needed in RPJP.
Architecture microservices create teams to become independent	No	In the process resource system human very minimum is only writers, so that deployment is hindered
Enables trial and technology adoption new with easy (Adaptable)	Implementation	Due to the service that to create using Node JS and Express JS and for the display web using React JS and mobile using React Native.
Services can be applied independent (Deployable)	Implementation	Because each service is deployed on its server so that each can stand alone. There is the use of service especially service organism can be used for the data collection of flora and fauna according to RPJP.
Service small and maintainable (Maintainable)	Implementation	Because each service managing object that are small so that easy maintainable and can be managed.

CONCLUSION

The Geographic Information System (GIS) for Kutai National Park, for mapping the location of flora and fauna, was built by implementing the extreme programming method. This GIS is created using two user interfaces: a website interface for administrators and a mobile application for general users. The website application uses React JS, and the mobile application uses React Native.Microservices are designed using NodeJS and ExpressJS, resulting in 3 services, namely organism service, map service, and user service. This system is built with website and mobile interfaces using microservices architecture, which facilitates sustainable development because the same services created earlier can be used for other features that will be developed.

The Geographic Information System using microservices divides the service based on the needs defined in the use case. The stages involve identifying system requirements, then identifying services, and identifying API services and collaboration. As a result of this implementation, three services were obtained, namely user service, organism service, and map service. The questionnaire results from testing the application on users revealed the following outcomes: for admins, the percentage of those selecting strongly agree was 70.3%, and agree was 20.6%; for regular users, the percentage of those selecting strongly agree was 47.9%, agree was 43%, and neutral was 3.8%.

DAFTAR PUSTAKA

A. Darmayantie, F. Teknologi, and J. Barat, “Microservice Architecture in Case Study of Information Systems,” *J. Tech. Inf. Dev.*, vol. 25, no. 1, pp. 50–63, 2020.

Albakri, M. H., and H. Al-Emran. “Integration of GIS and IoT in Disaster Management.” Journal of Environmental Informatics, vol. 41, no. 2, pp. 76–83, 2018. DOI: 10.1016/j.envsci.2017.12.001.

Bastiaansen, R. “Advanced GIS for Biodiversity Conservation.” Geoinformatics Journal, vol. 52, pp. 22–27, 2021. DOI: 10.1016/j.geoinf.2021.06.003.

doi.org/10.51967/jurnalagriment.v10i1.3308

Budi, C. S., and A. M. Bachtiar. "Microservices Architecture Implementation in Comrades Backend." Proceedings of the University of Computer Indonesia, pp. 1–6, Bandung, Indonesia, 2018.

Chang, K.-T. "Geographic Information System." International Encyclopedia of Geography, pp. 1–10, 2019. DOI: 10.1002/9781118786352.

Chen, J. "Geospatial Technologies for Wildlife Management." Springer, 2020. ISBN: 978-3-030-40223-4.

Choi, S., and J. Lee. "Integrating GIS with Cloud Computing for Environmental Research." Computing in Science and Engineering, vol. 19, no. 6, pp. 34–44, 2018. DOI: 10.1109/MCSE.2018.2873545.

Darmayantie, A., F. Teknologi, and J. Barat. "Microservice Architecture in Case Study of Information Systems." Journal of Technology and Information Development, vol. 25, no. 1, pp. 50–63, 2020.

H. Suryotrisongko, "Microservice Architecture for Information System Resilience," *Sisfo*, vol. 06, no. 02, pp. 231–246, 2017, doi: 10.24089/j.sisfo.2017.01.006.

Habiburrahman. "Design and Development of Tourism Mapping System for Nuraksa Grand Forest Park." Thesis, Gadjah Mada University, Yogyakarta, Indonesia, 2020.

Hart, G. GIS Solutions for Environmental Management. Redlands, CA: ESRI Press, 2017. ISBN: 978-1-58948-568-0.

Johnson, H. "Enhancing GIS Mapping with AI." Journal of Spatial Analytics, vol. 11, no. 3, pp. 34–40, 2020. DOI: 10.1016/j.jspatial.2020.02.004.

Kato, T., and S. Liu. "Big Data and Its Applications in Geographic Information Systems." Journal of Big Data Research, vol. 4, no. 2, pp. 82–93, 2020. DOI: 10.1016/j.jbdar.2020.04.002.

Kraak, M. J. "Cartography in Times of GIS and Big Data." International Journal of Cartography, vol. 4, no. 1, pp. 1–12, 2018. DOI: 10.1080/23729333.2018.1442339.

Kwon, S., and J. Kim. "Optimizing GIS Data Models for Cloud Computing." International Journal of Cloud Computing, vol. 12, no. 3, pp. 108–119, 2019. DOI: 10.1016/j.cloud.2019.05.007.

Lewis, J., and M. Fowler. "Microservices: A Definition of This New Architectural Term." Accessed Apr. 10, 2021. https://martinfowler.com/articles/microservices.html. (Accessed: Apr. 10, 2021).

Martinez, R., and S. Fadely. "Geospatial Data Management for Large Scale Environmental Monitoring." Environmental Management Journal, vol. 44, pp. 12–18, 2019. DOI: 10.1016/j.envman.2019.05.012.

Morris, D. "The Role of Microservices in Modern IT Architecture." Journal of Computing, vol. 22, pp. 201–212, 2020. DOI: 10.1109/JC.2020.091923.

Nissen, T. A. "A Review of Geographic Information Systems in Conservation Planning." Environmental Science and Policy, vol. 58, pp. 61–72, 2018. DOI: 10.1016/j.envsci.2015.02.004.

P. L. Lokapitasari Belluano, P. Purnawansyah, B. L. E. Panggabean, and H. Herman, "Student Creativity Program Information System Based on Web Service and Microservice," *Ilkom J.*, vol. 12, no. 1, pp. 8–16, 2020, doi: 10.33096/ilkom.v12i1.492.8-16.

Pahl, C., and P. Jamshidi. "Proceedings of the 10th International Conference on Cloud Computing and Services Science." CLOSER Proceedings, vol. 1, pp. 137–146, 2020. DOI: 10.5220/0008651401370146.

Park, H. "GIS Applications in Smart Urban Planning." Journal of Urban Management, vol. 7, no. 4, pp. 203–210, 2018. DOI: 10.1016/j.jum.2018.11.003.

Richards, M. "Microservices: The Evolution of Building Software." IEEE Cloud Computing, vol. 4, no. 5, pp. 6–8, 2017. DOI: 10.1109/MCC.2017.3641442.

Silverman, B. W. “Mapping Flora and Fauna Using Advanced Remote Sensing.” Journal of Remote Ecology, vol. 2, pp. 57–62, 2019. DOI: 10.1016/j.jreco.2019.03.007.

Singh, R. K. “Applications of Microservices in Spatial Data Infrastructure.” IEEE Access, vol. 7, pp. 179–189, 2019. DOI: 10.1109/ACCESS.2019.2893509.

Thomas, D., and C. Ma. “Microservice Solutions for Modern GIS Platforms.” IEEE Geoscience and Remote Sensing Magazine, vol. 7, no. 2, pp. 17–24, 2019. DOI: 10.1109/MGRS.2019.2943833.

Vargas, C., and D. T. Sanchez. “Microservices Implementation for Geospatial Data Analysis.” Journal of Geospatial Information Science, vol. 16, no. 2, pp. 11–23, 2020. DOI: 10.1016/j.jgis.2020.01.009.

Y. Chandra, T. Putra, T. Adi, P. Sidi, and J. E. Samodra, “Implementation of Microservice Architecture in Islamic Education Web Applications,” *J. Tech. Inf. Dev.*, vol. 1, no. 1, pp. 88–97, Nov. 2020.

Yang, Z., and L. Zeng. “Microservices Architecture for Urban GIS Applications.” Smart Cities Journal, vol. 1, no. 1, pp. 22–30, 2020. DOI: 10.1016/j.smartcity.2020.03.008.

Zhang, H., and X. Liu. “Cloud-Based GIS Applications in Environmental Protection.” Environmental Informatics Journal, vol. 36, pp. 52–63, 2019. DOI: 10.1016/j.enviroinfo.2018.12.004.