CoreID JOURNAL

Vol. 3 No. 1, March 2025 (14-22)
ISSN: 2987-6990
DOI: https://doi.org/10.60005/coreid.v3i1.77

REST-API Implementation for Optimizing Features in the K-Mob Jabar Smart ASN Application

Nur Halizah[1], M. Rifqi Azkiya Jamaludin[2]
[1]Department of Informatics, UIN Sunan Gunung Djati Bandung, Indonesia
[2]Department of Information System, Pukyong National University, Busan, South Korea

Article Info

Article history:

Received January 20, 2025
Revised March 15, 2025
Accepted March 22, 2025

Keywords:

Aplication
K-Mob
Ionic Framework
API
REST-API

ABSTRACT

The Jabar Smart ASN (JSA) application is a staffing application used to access other West Java Province staffing applications, one of which is the K-Mob application. The JSA application cannot be fully used, because there are still some features in K-Mob that are not optimal in accordance with the expected functions. So that from these problems, the implementation of REST-API on the features of the K-Mob application in Jabar Smart ASN is carried out. This research successfully implemented REST-API on the Working Hours, Claims and Appeals, and Recapitulation features with a response code of 200. However, the Subordinate Presence feature gave a response code of 500, caused on the server side. Although there are problems in the REST-API implementation process, the problems have been identified and can be the basis for further improvement.

Corresponding Author:

Nur Halizah
Department of Informatics, UIN Sunan Gunung Djati Bandung,
Jl. A.H. Nasution No. 105 Cibiru, Kota Bandung, 40614, Indonesia.
Email: nurhalizahh81@gmail.com

1. INTRODUCTION

The development of technology has brought changes and convenience to every human activity. This great influence on the field of technology can be felt, one of which is by companies. The growing technology allows a company to increase work efficiency and productivity quickly, easily and comfortably, through the use of mobile applications [1][2]. However, as the application's capabilities and scope increase, it will become more difficult to maintain system compatibility, scalable development, and seamless data transmission [3][4].

The role of mobile applications provides convenience to every employee, such as the K-Mob application. The K-Mob application is an application to facilitate the implementation of state civil service administration. The application is a real time application for managing employee performance, such as employee work goals and work behavior adjusted to employee attendance data, budget absorption data, and available employee performance output data [5][6]. Apart from the K-Mob application, there is a new application which is a container of previous applications, namely the Jabar Smart ASN (JSA) application. This JSA application is a staffing application that is used to access other staffing applications. The JSA application consists of many West Java provincial staffing applications, one of which is the K-Mob application. This JSA application was developed by the Regional Staffing Agency of West Java Province (BKD JABAR) to facilitate the activities of ASN. As attached to Government Regulation Number 53 of 2010 Chapter Article 5, which reads “Carry out official duties entrusted to civil servants with full devotion, awareness, and responsibility” [7].

The JSA application is not yet fully usable, because there are still some features in K-Mob that need to be optimized in accordance with the expected functions [8]. Some of these features in the K-Mob application at JSA are the features of working hours, claims and appeals, recapitulation, and subordinate attendance. From this problem, it is necessary to apply Application Programming Interface (API) technology, namely REST-API. API is a link that allows applications to connect and share data. Meanwhile, REST is a communication architecture that is applied in application development [3][9].

API or Application Programming Interface is a set of functions, subroutines, communication protocols, or tools for creating software [5][7]. In the K-Mob application in JSA, the API applied is the REST API which is an existing architecture in the API where to perform a data transmission using HTTP which allows effective data exchange [10]. So, the purpose of applying the REST API to the K-Mob application features is to integrate and optimize the application with the back-end system in order to facilitate the data retrieval process and ensure that each feature runs according to its function.

2. THEORETICAL FOUNDATION

Theories and discussions related to the research conducted can be explained as follows:

2.1 Mobile Application

Mobile applications are mobile-based applications that can be used on mobile phones without the need to search for keywords in the browser. Mobile applications are applications that offer many conveniences that can be accessed by anyone, anytime and anywhere [11]. Mobile applications are applications made for special mobile-based platforms (mobile devices) with a specific purpose to help users according to the type of application provided [12].

2.2 Ionic Framework

Ionic is one of the frameworks that can be used in making mobile-based applications that are open-source. Ionic framework is a framework of a software that can be run on different mobile operating systems such as Android, IOS, and Windows Phone, and Ionic itself has a concept that can be run on different operating systems [13]. The Ionic framework is also one of the supporting tools for the Cordova framework which makes it easier for developers to develop their applications [14]. In addition, the Ionic framework uses HTML5, CSS, and JavaScript which can help in front-end development [15][16].

2.3 REST-API

REST-API or Representational State Transfer - Application Programming Interface is a form of software architecture that contains several rules in making web services, namely restrictions intended to limit programmers in accessing or performing a command to the database [14]. REST-API is a web-based communication architecture standard used for service development in applications, and HTTP (Hypertext Transfer Protocol) as a protocol for data communication [14][17]. The way REST-API works is done through HTTP links that allow state transfer. In addition, each API call made uses several commonly used methods such as GET, POST, PUT, and DELETE. The use of REST-API allows systems to be connected to each other, and facilitates the sending and receiving of data [18].

3. RESULTS AND DISCUSSION

After going through several processes of identifying problems that occur and carrying out the stages of problem solving that have been carried out, there are results that have been achieved in the implementation of REST-API on the features of the K-Mob application.

3.1 Impelementation of REST-API on orking Hours Feature

The working hours feature is a feature that functions to make submissions for changing working hours, double shifts, and off work. After the submission is made, the submission will be approved or rejected by the supervisor. The process of submitting a change starts by filling out the form, then the data that has been submitted will appear on the initial display. Here's a look at the working hours feature:

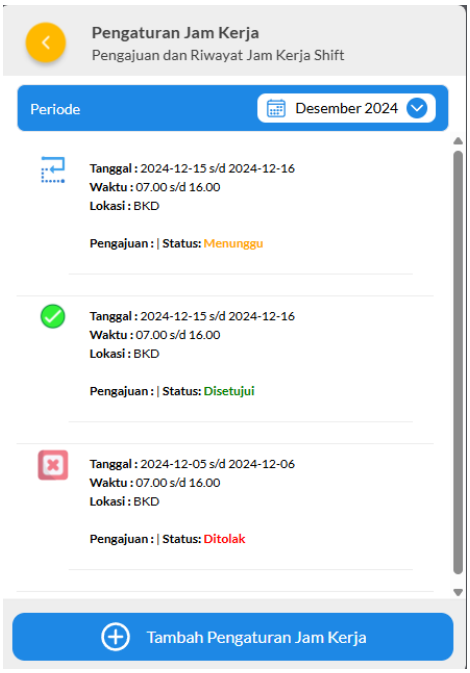


Figure 1. Working Hours Feature Initial View

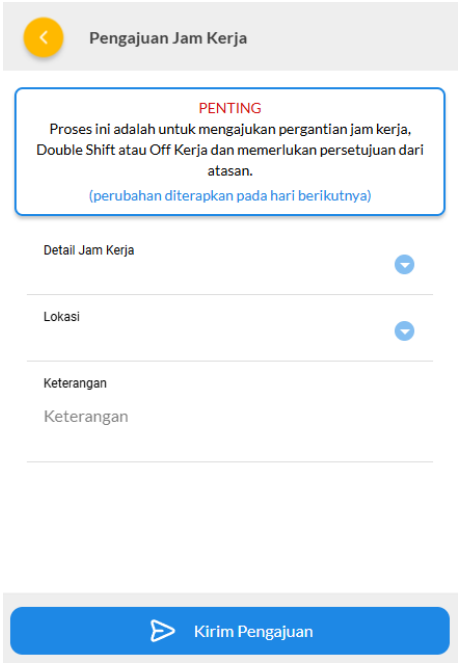


Figure 2. Working Hours Feature Form View

In Figure 1, the information displayed is dummy because the original data is privately owned by one of the ASN. Therefore, dummy information is used to replace the original data in the display section. However, API data calls are still made. In the initial display working hours feature, there are two REST-APIs that have successfully connected with a response code of 200. In addition, the formular display, sending new data that has successfully connected with a response code of 200. The following is a display of API data that has been successfully connected:

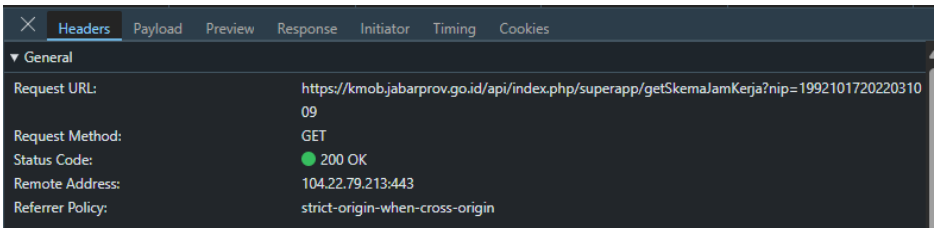


Figure 3. REST-API Display getJadwalKerja Successfully Connected

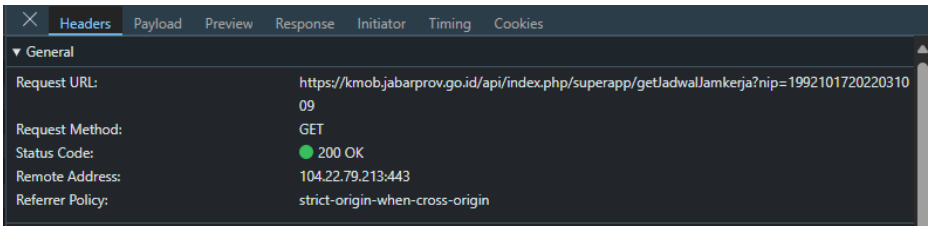


Figure 4. REST-API Display getSkemaJadwalKerja Successfully Connected

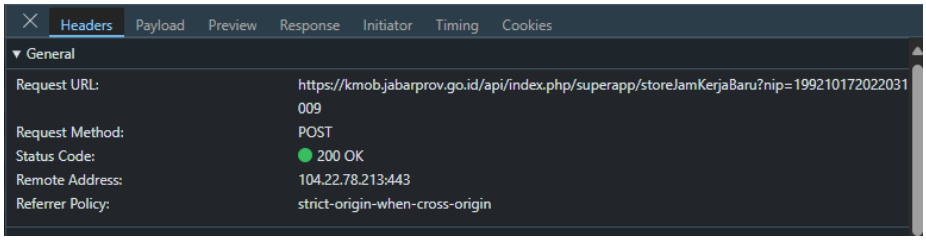


Figure 5. REST-API Display storeJadwalKerjaBaru Successfully Connected

3.2 Implementation of REST-API on Claims and Appleas Features

The claims and appeals feature is a feature that functions to submit claims and appeals. Problems that can be claimed and appealed by employees are, attendance in / return is not recorded, leave applications are not recorded, late due to K-Mob error, wrong working hours scheme that causes absences, recapitulation is not appropriate / wrong, and absent because it does not submit off. Just like in the working hours feature, the submission process by filling out the form in the claims and appeals feature by including evidence on the problems obtained. The following is a view of the claims and appeals feature:

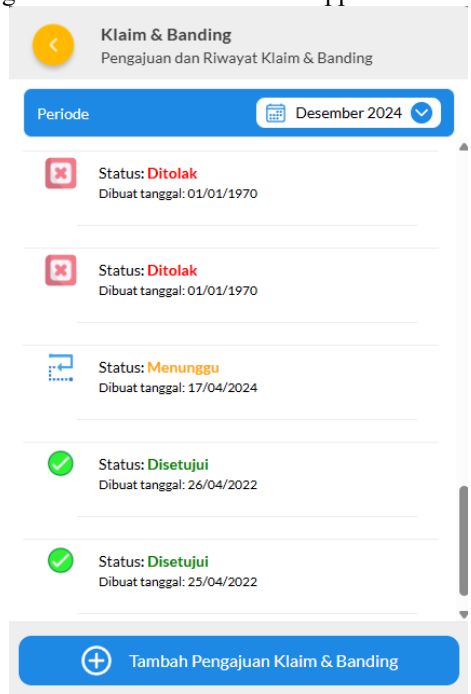


Figure 6. Claims and Appeals Features Initial View

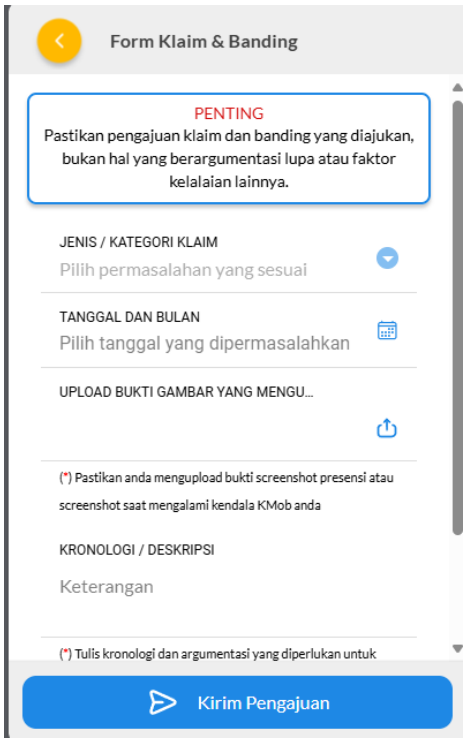


Figure 7. Claims Appeal Feature Form Display

In Figure 3. 6, the claims and appeals feature successfully calls API data with data displayed based on information contained in the database, and the API call is successfully connected with a 200 response code. The following is a display of API data that is successfully connected:

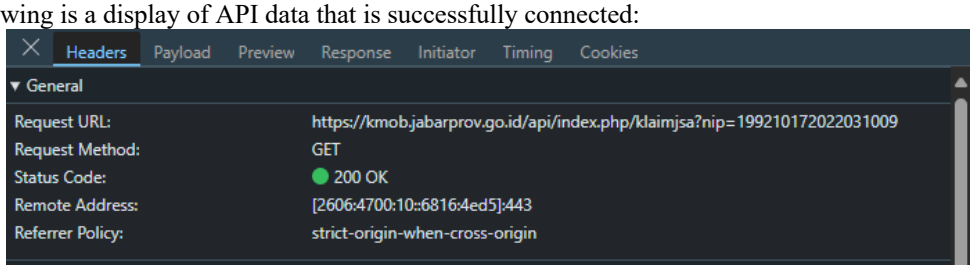


Figure 8. REST-API View of klaimjsa Successfully Connected

3.3 Recapitulation Feature Code Flow Adjustment

The recapitulation feature aims to be able to see the agenda that has been implemented every month. The agenda recap displayed is, External Service, Dynamic Working Arrangement (DWA), Permit, and Leave. In the recapitulation feature, the previous code had a static code flow. Where, the code flow in the recapitulation feature relies on data that is defined directly in the application, so it requires manual updates every time you want to make changes. Therefore, adjustments are made to dynamic code through API integration. In this adjustment, the code flow is changed to be more structured. The previous agenda list is replaced by taking data from the REST-API that has been provided.

```
const agendas = ref([
    {
        id: 1,
        type: "dinas",
        date: "2024-07-01",
        title: "Dinas Luar",
        time: "09:00",
        description: "Ke Jakarta",
    },
```

Figure 9 Code Display Before Improvement

```
data() {
    return {
        agendas: [
            {
                id: 1,
                type: "dinas",
                date: "2024-11-01",
                title: "Dinas Luar",
                time: "09:00",
                description: "Ke Jakarta",
            },
```

Figure 10. Code Display After Improvement

Figure 3.9 and Figure 3.10 are examples of the changes made. In Figure 3.9, dummy data is stored in a static const agenda, so data changes must be made manually in the code. While in Figure 3.10, dummy data is stored in agendas, which are defined directly in data(). However, the changes are adjusted so that the data can be dynamically retrieved from the API. So, the JSA application on K-Mob can retrieve data automatically.

3.4 Implementation of REST-API in Recapitulation Features

The recapitulation feature is used to be a monthly recap for each ASN employee. Therefore, REST-API implementation is needed to retrieve data according to the information on the Outer Service agenda, Dynamic Working Arrangement (DWA), Permits, and Leave. The incoming agenda will be marked with each different color, according to the data called. And in the display of this recapitulation feature, the data that appears is dummy, because the data is personal to the owner of one of the ASN. The following is a display of features in the recapitulation:

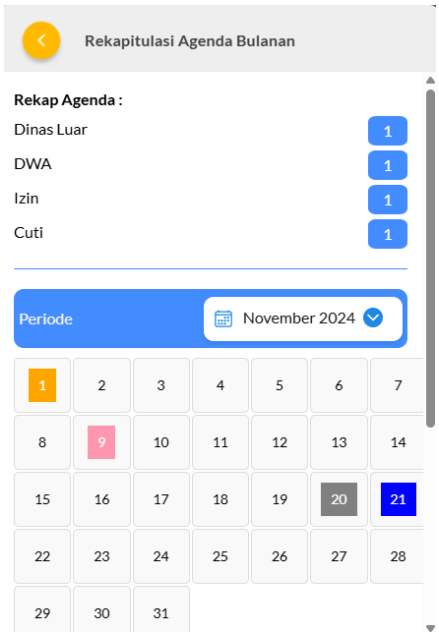


Figure 11. Recapitulation Feature Display

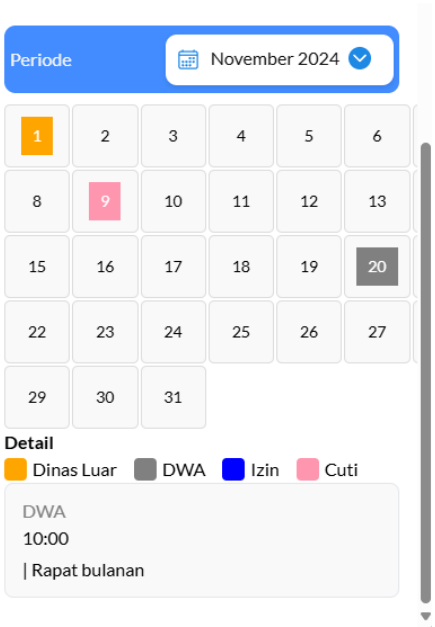


Figure 12. Detailed Information Display in the Recapitulation Feature

The recapitulation feature successfully called the connected API data with a response code of 200. The following is a display of the API data that was successfully connected:

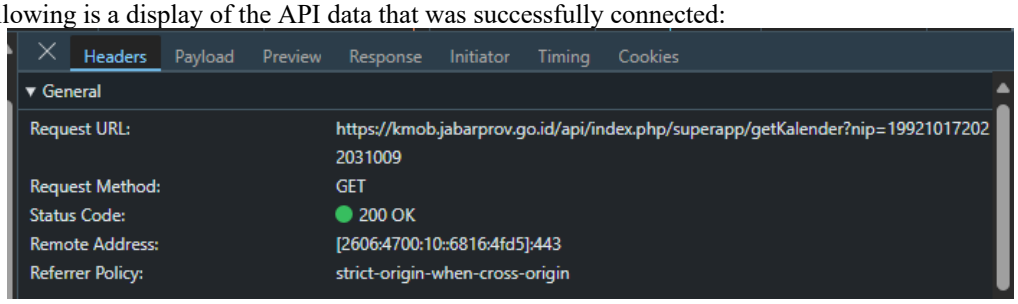


Figure 13. REST-API Display getKalender Successfully Connected

3.5 Implementation of REST-API on Subordinate Attendance Feature

The subordinate attendance feature is one of the features used by superiors or leaders in conducting attendance for subordinates directly. This attendance can be done by superiors, if their subordinates do not bring a cellphone or cannot do attendance as usual. The following is a display of the subordinate attendance feature:

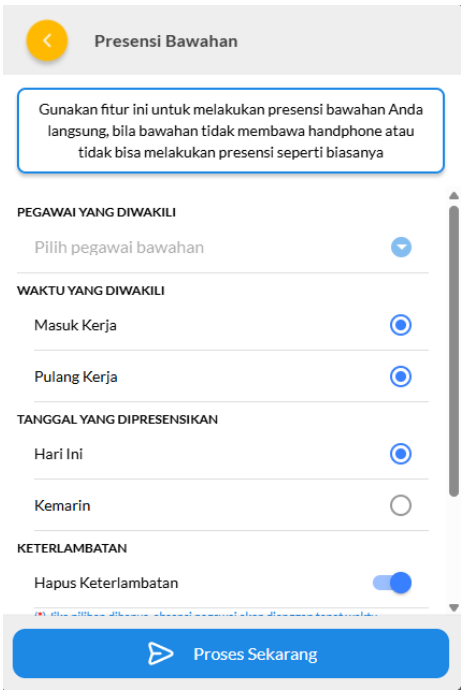
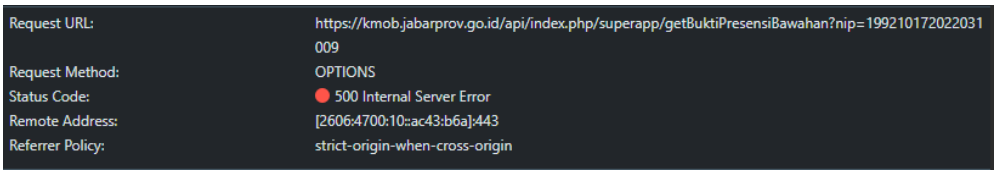


Figure 14. Subordinate Attendance Feature Display

In subordinate attendance, REST-API has been implemented to integrate data with the server. However, the API call gives a response code of 500, as in the following image.



Fiture 15. REST-API Display getBuktiPresensiBawahan Not Connected

The implementation of REST-API on this subordinate attendance feature shows an error. To determine the cause of the error, identification is carried out in the data call process and checking the available API documents. All data sent to the API is in accordance with the format received in the API documentation. So it can be identified that the error in retrieving the API data comes from the server side.

3.6 Fixing Errors in Several Features in the K-Mob Application

In the K-Mob application in the Jabar Smart ASN (JSA) Application, there are still errors that often appear. These errors are caused by several commas and semicolons that are forgotten to be implemented in several features. In addition, sometimes there are several libraries that are forgotten to be installed in the code, which causes the code to error and not work.

First of all, in fixing the error, identification is carried out by looking at the application screen or console, where the error is located. Then repairs are made to several features. After the error has been successfully eliminated, the next step is to test the features, whether they are in accordance with their functions, and check the API data that has been called.

4. CONCLUSION

From the implementation of the Internship that has been carried out at the Regional Civil Service Agency (BKD), it can be concluded that the implementation of Reat-API on the K-MOB Jabar Smart ASN (JSA) application has been successfully integrated according to project needs. In the Working Hours, Claim Appeals, and Recapitulation features, it has successfully called API data with a response code of 200. However, the Subordinate Presence feature gives a response code of 500 Interval Server Error. Although there are problems in the REST-API implementation process, the problems have been identified as obstacles from the server side.

REFERENCES

[1] L. S. Saputra and H. F. Hardiyatmoko, “Implementasi Rest Api Pada Sistem Pendataan Kapal

Menggunakan Laravel 9 Dan Nuxtjs," *Implementasi Rest Api Pada Sist. Pendataan Kapal Menggunakan Laravel 9 Dan Nuxtjs*, vol. 2, no. 1, pp. 10–21, 2024.

[2] N. I. Putri, Y. Saputra, S. Nurhayati, and D. Dzarwah, "Sistem Pendukung Keputusan Pemilihan Program Studi Calon Mahasiswa Menggunakan Weighted Product (WP)," vol. 10, no. 2, pp. 322–327, 2023.

[3] R. K. Safitri and H. P. Putro, "Implementasi REST API untuk Komunikasi Antara ReactJS dan NodeJS (Studi Kasus : Modul Manajemen User Solusi247)," *Automata*, vol. 2, no. 1, pp. 1–5, 2021.

[4] A. Pudoli, Y. Yulianawati, and I. Suwandi, "Implementasi Web Service Restful Dengan Autentikasi JSon Web Token Berbasis Web dan Android," *Acad. J. Comput. Sci. Res.*, vol. 5, no. 2, p. 95, 2023, doi: 10.38101/ajcsr.v5i2.9410.

[5] C. Chandra1 *et al.*, "Perancangan dan Implementasi RESTful API untuk Aplikasi Mobile Pembelajaran Flora dan Fauna pada Google Cloud Platform," vol. 4, no. 1, pp. 58–69, 2024, doi: 10.54259/satesi.v4i1.2850.

[6] M. Irfan, P. Alkautsar, A. R. Atmadja, and Wildan Budiawan Zulfikar, "Diagnosis of Asthma Disease and The Levels using Forward Chaining and Certainty Factor," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 5, pp. 761–767, 2022, doi: 10.29207/resti.v6i5.4123.

[7] B. A. Yudanto *et al.*, "Implementasi REST API Menggunakan Bahasa GO," vol. 06, no. 02, pp. 384–393, 2025.

[8] A. R. Atmadja, A. Rahmawati, C. N. Alam, P. Dauni, and Y. Saputra, "Sentiment Analysis on Tourism Place using Naive Bayes," *Proceeding 2023 17th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2023*, pp. 1–6, 2023, doi: 10.1109/TSSA59948.2023.10366891.

[9] Fatchur Shofyan and Rizky Tahara Shita, "Implementasi Web Service Restful API dengan Autentikasi Personal Access Tokens dan Algoritma AES 256," *J. Ticom Technol. Inf. Commun.*, vol. 12, no. 3, pp. 108–114, 2024, doi: 10.70309/ticom.v12i3.130.

[10] S. B. Farchani, B. A. Kusuma, and N. Hermanto, "Implementasi rest api dalam pengembangan backend inventory peminjaman," vol. 10, no. 2, pp. 1404–1413, 2025.

[11] Rangga Gelar Guntara and V. Azkarin, "Implementasi dan Pengujian REST API Sistem Reservasi Ruang Rapat dengan Metode Black Box Testing," *J. Minfo Polgan*, vol. 12, no. 1, pp. 1229–1238, 2023, doi: 10.33395/jmp.v12i1.12691.

[12] M. A. Novianto and S. Munir, "Analisis dan Implementasi Restful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi," *J. Inform. Terpadu*, vol. 8, no. 1, pp. 47–61, 2022, doi: 10.54914/jit.v8i1.409.

[13] S. Widyaningtyas and T. Wahyono, "Implementasi Rest Api Menggunakan Retrofit Pada Aplikasi Monitoring Grooming Berbasis Android," *IT-Explore J. Penerapan Teknol. Inf. dan Komun.*, vol. 3, no. 2, pp. 147–161, 2024, doi: 10.24246/itexplore.v3i2.2024.pp147-161.

[14] N. Ransi *et al.*, "Implementasi Web Service Menggunakan Rest Api Untuk Integrasi Data Tata Naskah Dinas Pada Aplikasi OFC Universitas Halu Oleo," *Just TI (Jurnal Sains Terap. Teknol. Informasi)*, vol. 14, no. 2, pp. 80–88, 2022, [Online]. Available: http://e-journal.polnes.ac.id/index.php/justi/article/view/1547

[15] A. H. THohari, F. Hidayat, M. Fani, and N. Nelmiawati, "Rancang Bangun Sistem Pengelolaan Barang Milik Negara Berbasis QR Code," *J. Integr.*, vol. 14, no. 1, pp. 35–46, 2022, doi: 10.30871/ji.v14i1.3975.

[16] M. F. Ramadhan, "Pengembangan Rest Api Sistem Uiiadmisi dengan Microservices," 2023.

[17] A. Gufron, F. N. Putra, and N. Haryuni, "Rancang Bangun Aplikasi Formulasi Pakan Ternak Berbasis WEB Menggunakan Framework CodeIgniter 3," *J. Autom. Comput. Inf. Syst.*, vol. 1, no. 2, pp. 46–54, 2021, doi: 10.47134/jacis.v1i2.12.

[18] B. Fajri, A. Arwan, and D. Pramono, "Sistem Manajemen Toko Perhiasan Perak menggunakan Progressive Web App (Studi Kasus: Toko Borneo Silver Banjarmasin)," *J. Pengemb. Teknol. ...*, vol. 5, no. 10, pp. 4612–4620, 2021, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/10021/4469