Information Technology Studies Journal (ITECH)
Vol 2 (2) 2025 : 144-155

IMPLEMENTATION OF E-COMMERCE CERAMIC SHOP USING APPLICATION PROGRAMMING INTERFACE WITH JSON WEB TOKEN AUTHENTICATION

IMPLEMENTASI TOKO KERAMIK E-COMMERCE MENGGUNAKAN APPLICATION PROGRAMMING INTERFACE DENGAN AUTENTIKASI JSON WEB TOKEN

Mepa Kurniasih[1], Agung Alam Putra[2], Eka Anggara[3]
Universitas Budi Luhur[1,2,3]
*mepa.kurnia@gmail.com[1], 1911510152@student.budiluhur.ac.id[2],
1911510392@student.budiluhur.ac.id[3]

**Corresponding Author*

ABSTRACT
CV. Apuy Stone is a ceramic and granite distribution company located in Serang, Banten. One of the problems faced is the accumulation of goods in the warehouse due to a mismatch between customer demand and the amount of available stock. Often, products that customers are interested in are not available, while other products pile up and remain unsold for a long time. Based on these conditions, the purpose of this Internship Lecture is to design an information system-based solution that can help the company manage inventory more efficiently. The methods used include direct field observation, interviews with related parties, and analysis of sales transaction data for the past two years. The author built a web-based application using the Laravel framework and MySQL database that aims to process sales data and provide recommendations on inventory. This application is designed to identify products with a low turnover rate (slow-moving) and products with a high level of demand, and assist management in planning more targeted procurement of goods. The results of the system implementation show an increase in accuracy in decision-making related to stock management and a decrease in the number of items piling up unsold. In conclusion, the information system developed can be an effective tool in inventory control based on historical sales data. The benefits of compiling this KKP are directly felt by the company in the form of warehouse operational efficiency, as well as contributing to the development of information technology solutions in the world of small and medium businesses.
Keywords: JSON WEB Token, WEB-Based Application, Transaction History Data, Application Programming Interface.

ABSTRAK
CV. Apuy Stone merupakan perusahaan yang bergerak di bidang distribusi keramik dan granit yang berlokasi di Serang, Banten. Salah satu permasalahan yang dihadapi adalah penumpukan barang di gudang akibat ketidaksesuaian antara permintaan pelanggan dan jumlah stok yang tersedia. Seringkali produk yang diminati pelanggan tidak tersedia, sementara produk lain justru menumpuk dan tidak terjual dalam waktu lama. Berdasarkan kondisi tersebut, tujuan dari Kuliah Kerja Praktek ini adalah untuk merancang solusi berbasis sistem informasi yang dapat membantu perusahaan dalam mengelola persediaan barang secara lebih efisien. Metode yang digunakan meliputi observasi langsung di lapangan, wawancara dengan pihak terkait, serta analisis data transaksi penjualan selama dua tahun terakhir. Penulis membangun sebuah aplikasi berbasis web dengan menggunakan framework Laravel dan basis data MySQL yang bertujuan untuk mengolah data penjualan dan memberikan rekomendasi terhadap stok barang. Aplikasi ini dirancang untuk mengidentifikasi produk yang memiliki tingkat perputaran rendah (slow moving) dan produk dengan tingkat permintaan tinggi, serta membantu manajemen dalam merencanakan pengadaan barang yang lebih tepat sasaran. Hasil dari implementasi sistem menunjukkan peningkatan akurasi dalam pengambilan keputusan terkait pengelolaan stok serta menurunnya jumlah barang yang menumpuk tanpa terjual. Kesimpulannya, sistem informasi yang dibangun mampu menjadi alat bantu yang efektif dalam pengendalian persediaan berbasis data historis penjualan. Manfaat dari penyusunan KKP ini dirasakan langsung oleh perusahaan dalam bentuk efisiensi operasional gudang, serta memberikan kontribusi terhadap pengembangan solusi teknologi informasi dalam dunia usaha kecil menengah.

Kata Kunci: JSON WEB Token, Aplikasi Berbasis WEB, Data Histori Transaksi, Application Programming Interface.

1. INTRODUCTION

Ceramic tile inventory management is a crucial aspect of a store's operations, particularly in ensuring the availability of items needed by customers. Well-organized inventory makes it easier for consumers to find the products they're looking for. Furthermore, stores require recording tools, such as stationery, to capture purchase data and identify high-demand products. The use of web-based applications in architecture, particularly for managing ceramic tile inventory in home construction projects, provides significant added value. The collected information is not only useful for internal records but can also be analyzed to develop strategies for sustainable sales growth. Stock monitoring is crucial to prevent excess unsold inventory or shortages that can lead to customer dissatisfaction. Therefore, a system capable of automatically and accurately monitoring and analyzing inventory is essential.

Amidst increasingly competitive business conditions, businesses are required to be more astute in evaluating their product inventory to ensure market demand is consistently met. Continuous sales transactions, particularly in large-scale stores like supermarkets, generate invaluable data. This data can be used to study consumer purchasing patterns and formulate more effective marketing strategies in the future, with the hope of significantly increasing sales in the coming years.

CV. Apuy Stone is a company specializing in the supply of building materials, specifically ceramics. The company operates in Serang, Banten, and was founded in 2013. To date, CV. Apuy Stone relies on conventional marketing methods, including ordering via telephone and WhatsApp. This method is considered ineffective because it makes it difficult for the administration to respond quickly to customer requests, especially when there is a surge in orders.

To improve efficiency in transaction processes and customer service, an application system is needed that can facilitate online product ordering. This application is expected to provide customers with convenience in selecting products, making transactions, and tracking delivery. During its development, the Apuy Stone application was also designed to integrate with digital payment services such as Midtrans, as well as logistics services and shipping cost calculations through Raja Ongkir. This collaboration aims to simplify the payment process and simplify shipping cost calculations, thereby increasing convenience for both customers and administrators.

This research aims to implement a REST-based web service technology that can be accessed by various types of users, both through web and desktop applications. This system is designed to facilitate Apuy Stone customers in ordering products more conveniently, while also helping the company handle and manage incoming orders efficiently. This allows the ordering process to be faster and more effective.

The problem formulation in this study focuses on how the implementation of a sales web application can help overcome the problem of stockpiling of goods at CV. Apuy Stone, as well as how marketing strategies can be used to market ceramic products or brands that are experiencing stockpiling. This study specifically focuses on marketing strategies related to inventory management at the CV. Apuy Stone ceramic shop. The developed system is designed to assist shop owners in recording stock, transaction history, and product sales processes. The system architecture used in developing this application is the REST API as a communication medium between the frontend and backend, with PHP as the programming language in developing REST web services. The system testing process is carried out through two platforms, namely a web-based application and an Android mobile application.

2. LITERATURE REVIEW

2.1. Web Service

A web service is a software system designed to enable interoperability between applications running on various platforms. This technology enables the exchange of information and communication between systems over a network, typically using the HTTP protocol. Web services provide an interface described in a machine-readable format, allowing other applications to interact with them. One common form of communication is through SOAP (Simple Object Access Protocol) messages, typically sent using the HTTP protocol in XML format. This definition of a web service comes from the World Wide Web Consortium (W3C), the organization responsible for developing global web standards.

Although the SOAP standard is often used, web services are generally not limited to that protocol. Web services are applications accessible over the internet using standard protocols and XML as a data exchange format. Broadly speaking, web services are categorized into two main types: SOAP and REST.

2.1.1. SOAP (Simple Object Access Protocol)

SOAP is a formal communication protocol that uses XML as the basis for data exchange. This protocol defines a set of rules and standards used for sending messages, typically over HTTP or HTTPS. In its implementation, a SOAP-based web service sends requests to the server using a predefined XML structure. Similarly, the server's responses also have a specific XML format according to the SOAP standard.

2.1.2. REST (Representational State Transfer)

REST is an approach to web service development that adopts the concept of state transfer, or transition between states. This concept can be likened to when a browser accesses a web page; the server sends a representation of the page's state to the browser. When a user clicks a link or navigates, it is considered a transition from one state to another. REST operates on a similar principle, where interactions are conducted through HTTP links to perform various operations. Thus, each request sent to the server represents a change or transition in state. REST utilizes standard HTTP methods in its processes, such as GET to retrieve data, POST to send or add new data, and other methods such as PUT, DELETE, and so on for data manipulation purposes. HTTP methods like PUT and DELETE are also used in REST to update or delete data. The server's responses are generally in simple XML format, without the use of additional wrapping protocols. This makes the data easier to read and process (parse) on the client side. REST is generally better suited to resource-oriented web services. This means that REST is designed to provide various resources as the primary service, rather than as a set of functions or activities that process those resources. This approach emphasizes data manipulation and representation through URLs that represent specific resources. However, because REST lacks the same stringent standards as SOAP, its use is less than ideal in systems that require high interoperability between applications. In such contexts, formal standards are essential to allow disparate systems to communicate in a unified, structured language.

Five Limitations of REST Architecture:

1. **Client Server**

A type of computer network where the server serves client requests. This allows multiple users to have access to the same database at the same time, allowing the database to store a lot of information.

2. **Resource Identification**

REST uses a Uniform Resource Identifier (URI) to identify each available resource. With this approach, the URI serves as a unique address, allowing each resource to be accessed through a clear and easily recognizable identification scheme.

3. **Uniform Interface**

One of the hallmarks of REST is the use of a uniform interface for interacting with resources. This is typically implemented through the HTTP protocol with standard methods such as GET, POST, PUT, and DELETE, each of which has its own semantic meaning in data management.

4. **Self-Describing Messages**

Each request and response in REST must contain enough information to explain how the message is processed. REST supports various data representation formats such as JSON, XML, RDF, and others, which gives developers flexibility in choosing the format according to their needs, unlike SOAP, which is limited to XML.

2.2. API (Application Programming Interface)

An API is a software interface consisting of a set of commands or instructions, typically stored in a library, and serves as a link between software programs so they can interact with each other. An API describes how a program can communicate and utilize the functions of another program without requiring detailed knowledge of how the process works. As an illustration, an API can be likened to a contractor building a house. The homeowner simply states the work to be completed, while the contractor is responsible for the technical implementation methods and processes. In this analogy, the house is the software being developed, while the contractor represents the API that carries out specific tasks in the software development process, without the user needing to know the detailed working mechanisms.

2.3. JSON

JSON is a lightweight, easy-to-understand data exchange format that is easy to write and automatically generate. It is derived from the syntax of the JavaScript programming language, but its use is not limited to JavaScript. JSON is a text format that is language-independent, as its structure follows a writing style familiar to many C-family languages such as C, C++, C#, Java, JavaScript, Perl, and Python. Due to its simplicity and flexibility, JSON is an ideal choice for data exchange between systems, particularly in the development of web-based applications and APIs. JSON has several advantages over XML, including:

2.3.1. Writing Format.

When representing complex, hierarchical data structures, JSON has a simpler and more structured writing format. This makes it easier for developers to read and write data, making data processing and integration more efficient and faster.

2.3.2. Size.

JSON has the advantage of data size efficiency. For equivalent information, the JSON format typically requires fewer characters than XML. This can have a positive impact on data exchange speed, especially in slow network conditions or when used in data-intensive web applications. While the difference may not be noticeable for small data volumes, this efficiency benefit becomes significant when handling large amounts of data or in systems with limited bandwidth. However, if both JSON and XML are compressed before transmission, the size difference between the two becomes less significant.

2.4.3. Browser Parsing.

The processes or small elements that make up an XML or JSON document. For example, when data is available in JSON format, it must first be parsed before it can be read,

accessed, and manipulated by the system. Client-side (browser) parsing, also known as browser-side parsing, tends to be simpler when using JSON than XML. In JSON, this process can be performed using the eval() or JSON.parse() functions in JavaScript. XML documents, on the other hand, are typically processed through the XMLHttpRequest object, which involves additional steps in retrieving and interpreting the data structure.

3. RESULTS AND DISCUSSIONS

3.1. Problems

3.1.1. Problem Analysis

Apuy Stone Banten provides a variety of ceramic products with varying levels of demand. Some products are highly sought after, while others are less so. A problem that arises in the field is the buildup of stock for certain products due to a lack of optimal inventory data management. This is because stock information is not updated regularly, resulting in product availability not matching actual demand. Furthermore, the current ordering system is unable to support cross-regional transactions, such as between cities and provinces, which require integration with delivery services. Furthermore, customers come from diverse backgrounds and use different types of banks for payment. To improve service efficiency and facilitate product ordering, an application system is needed that can accommodate customer needs quickly, accurately, and flexibly. This system is expected to manage inventory in real time, support online ordering across regions, and facilitate the payment process.

3.1.2. Problem Solving Strategy

Based on the previously identified problems, a system is needed that can act as an intermediary between various cross-platform applications without depending on the programming language used by each application. In addition, the system must also be able to maintain the integrity and security of the data exchanged within it. To realize this solution, the development of a web-based e-commerce system is the right alternative, where this platform can be used as a link between digital marketing strategies and web-based technology services. In its implementation, the service architecture used refers to the Atomic Transaction concept, which aims to maintain the consistency and integrity of the overall data process. This is to ensure data security in RESTful Web Services. With the implementation of this system, it is hoped that an application will be created that can not only run optimally and cross-platform, but also guarantee the security of the data stored and processed within it.

IMPLEMENTATION AND PROGRAM TRIAL

3.2. Screen Display

Below is a screen display on the ceramic shop webservice E-Commerce application.

3.2.1. Customer Screen Display

1. **Login Page Screen Display**

Figure 3.1 displays the initial interface when a user (buyer) accesses the Ceramic Shop web application. On this page, the user is asked to complete the authentication process by filling in two fields: email and password. After the data is entered, the system will send the information to the REST API for credential validation. If the entered data matches that registered in the database, the user will be directed to the application's main page. Conversely, if there is an error in the email or password, the system will display an error message. This process aims to ensure access security and ensure that only registered users can log in to the system.

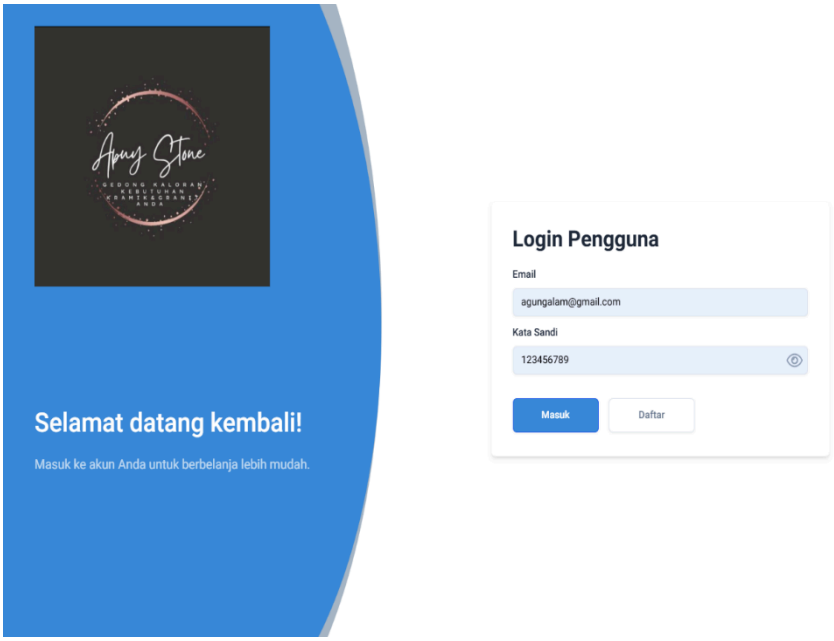

Figure 3.1. User Login Page Screen Display

2. List Page Screen Display

Figure 3.2 shows the registration page interface used by prospective buyers to create a new account in the Toko Keramik web application. On this page, users are required to fill in several personal data, such as full name, email address, password, and other information required for the registration process. Once all data is filled in correctly, the information will be sent to the REST API for processing and storage in the database. If the registration process is successful, the user will receive a success notification and be directed to the login page to enter the system. This page serves as the initial step so users can access the full application features, such as product ordering, delivery tracking, and transaction history.

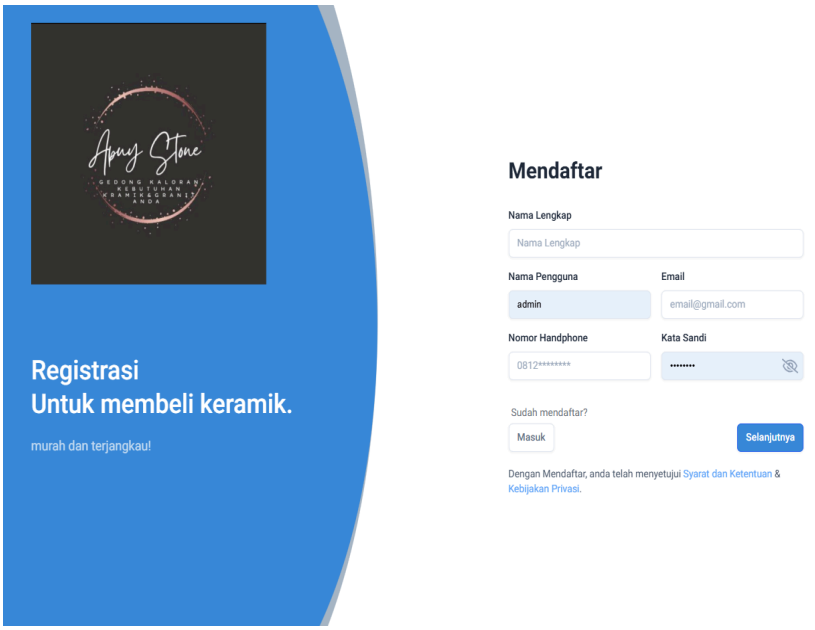

Figure 3.2. List Page View

3. Product Information and Cart Screen Display

In Figure 3.3 below is a display of the product information page screen when the buyer views the product information.

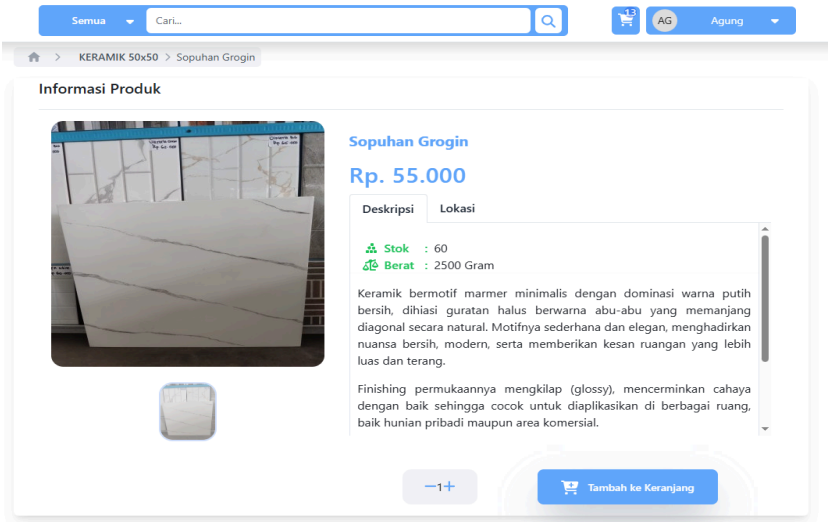

Figure 3.3. Product Information Screen Display

4. Cart Page Screen Display

In the following figure 4.5, the display of the shopping cart page that customers can do when making a purchase will go to the shopping cart first, and how many activities do the product insertion in the shopping cart then you can check out the type of product and category selected to do the checkout.

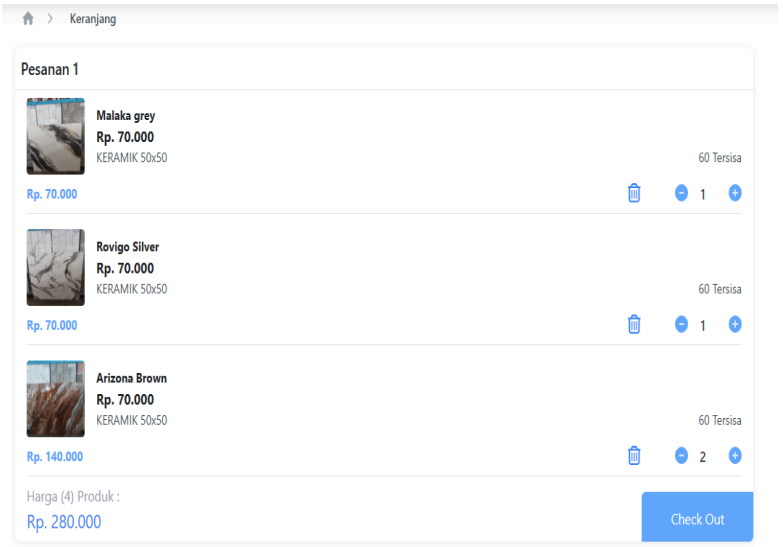

Figure 3.4. Cart Page Screen Display

5. Checkout Page View

In the following figure 4.6, the Checkout page screen is displayed. The function is to display the Shipping Address, the ceramic products ordered and the courier selection for the user. The courier selection is taken from the third party API from Raja Ongkir.

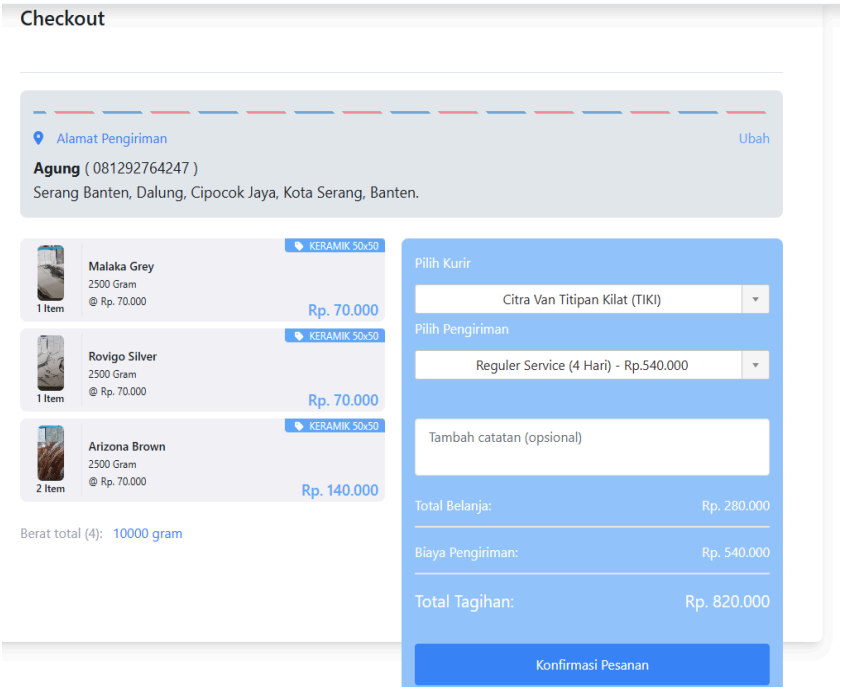

Figure 3.5. Checkout Page Screen Display

3.3. Web Service Testing

The display below shows the results of testing using Postman on all services in the web service.

1. Request Post Login Display

Screen view of Request POST login to register Email and Password for Admin.

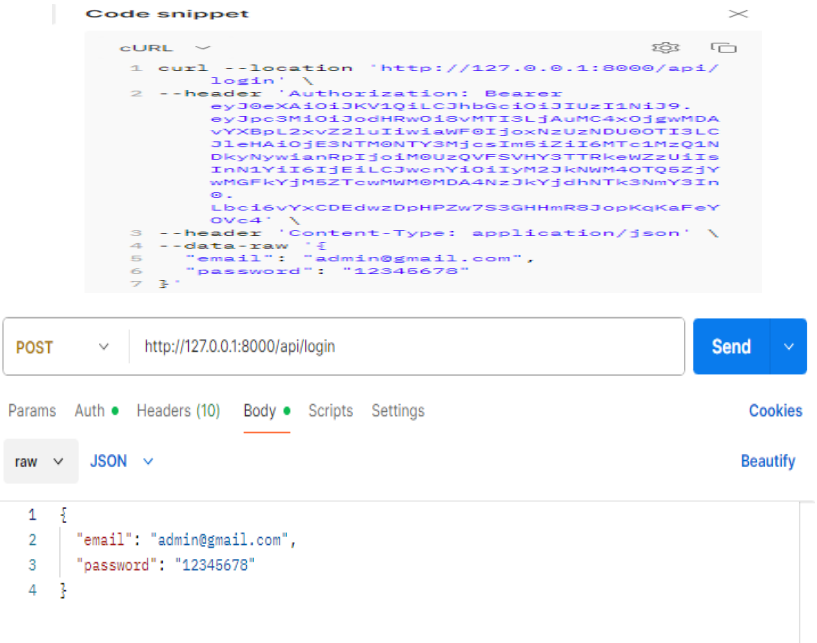

Figure 3.6. Post Request Login Form Display

2. The admin Login POST Response display displays the jwt token.

Screen view of POST Login Response Contains status, jwt token from login table.

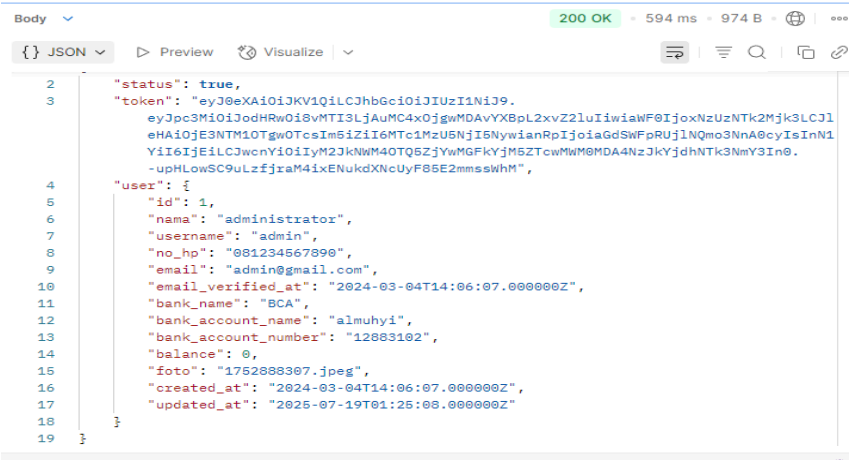

Figure 3.7. JWT Admin Login

Figure 3.8. Admin Login Response Form Display

3. Product Admin GET Request Display

Screen display of the product gets requested by the Admin to display product data.

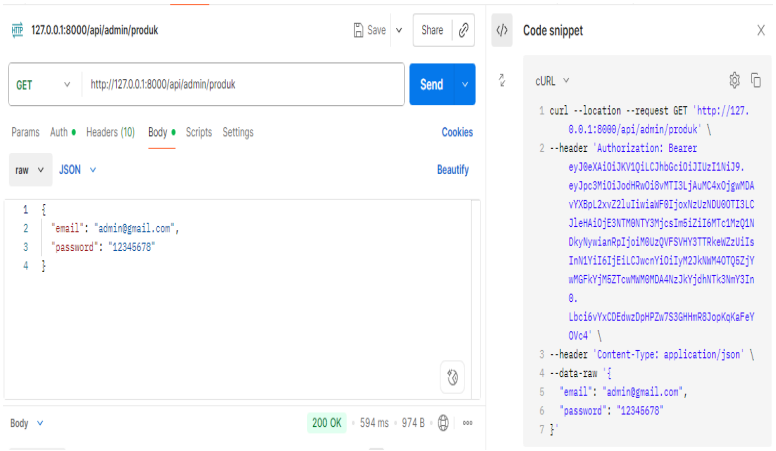

Figure 3.9. Admin Product Request Form Display

4. Display of the Get Admin Product Response Form.

Screen view of the GET response from the product admin which contains True data for the product list and table.

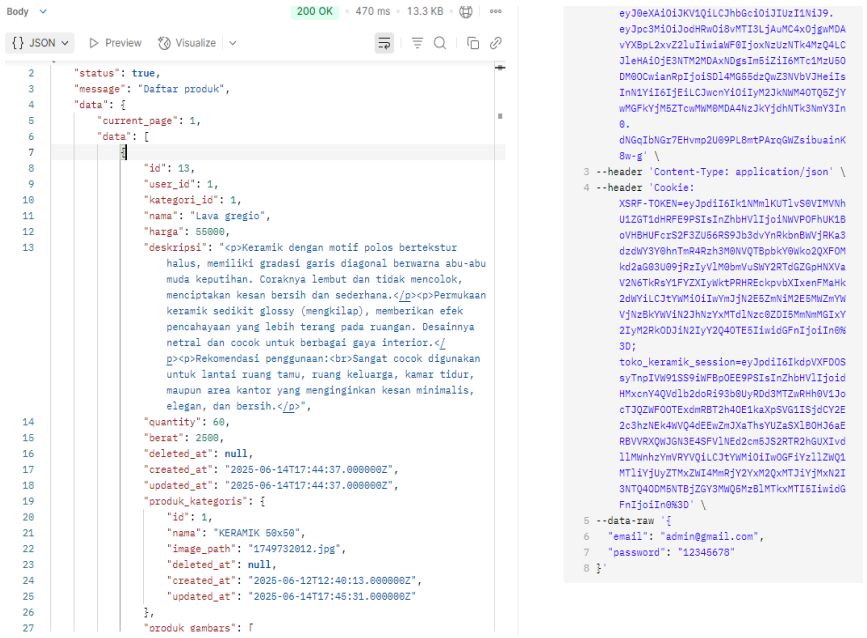

Figure 3.10 Display of the GET Response Form for the admin product

4. Display of the Admin Category Product Request Form

GET request form screen display to display admin category products.

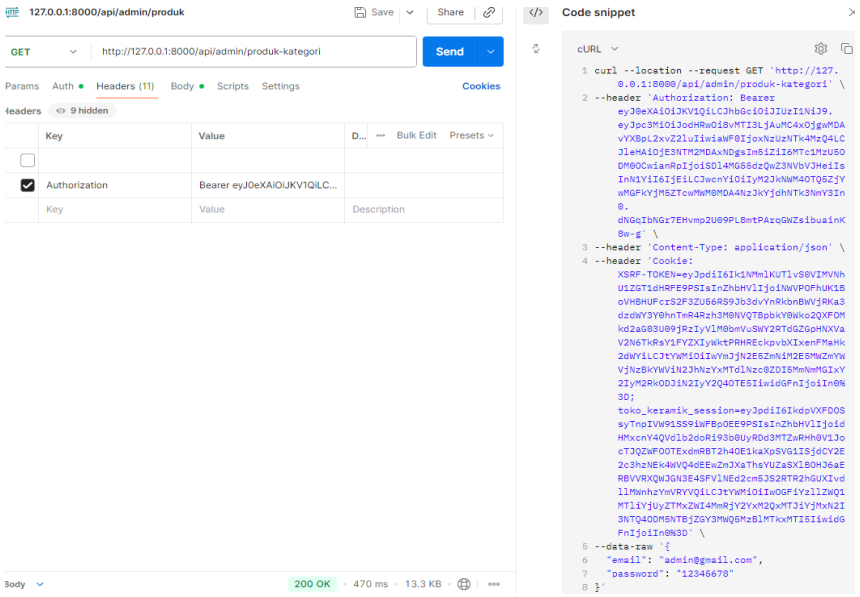

Figure 3.11. Display of the GET Request Form for Admin Category Products

Advantages and disadvantages of the Program

After analyzing the application test results, several advantages and disadvantages of this application can be found, namely as follows:

Advantages of the Program

1. User Friendly Interface
   The application is designed with a simple and easy-to-understand display, so that users, especially buyers, can quickly understand how to use the application.
2. Integration of Various Payment Methods
   The system has been integrated with Midtrans, allowing users to choose from a variety of payment methods, including bank transfers, e-wallets, and credit cards. This enhances the transaction process.
3. Shopping Cart Feature
   The app provides a shopping cart feature that allows users to add products to their shopping list before checking out. This feature makes it easier for users to manage and evaluate the products they plan to purchase.

Disadvantages of the Program

1. The drawback of the automatic feature of the shipping receipt number system which is located at the order checkout
2. Mobile application can only get data, cannot post from mobile

5. CONCLUSION

Based on the design, development, a series of trials, and program analysis of this application, it can be concluded that the web application bridged by a web service is capable of connecting buyers with the ceramic shop CV. Apuy Stone quickly and securely. Furthermore, the REST method in creating web services has been proven to be usable by various applications even using different programming languages.

The suggestions that can be given so that this application can run better in the future include: further development is expected to add more functions provided by the web service so that it can reduce the load on the client side, developers are advised to create applications on other platforms such as Android to make it easier for users to access, and the time to make requests and responses from REST is expected to run faster with better server hardware support.

6. REFERENCE

Ardhana, V. Y. P., Kusuma, R. D., & Firmansyah, D. (2023). Implementasi RESTful API pada Laravel dan simulator IoT Wokwi untuk pengukuran suhu dan kelembaban. Jurnal Teknologi dan Rekayasa, 14(1), 45–52.

Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures (Doctoral dissertation, University of California, Irvine).

Izhar, A. (2023). Pengembangan REST API dengan menggunakan Express JS untuk mencari mentor pribadi. Jurnal Informatika Terpadu, 9(1), 33–40.

Keith, J., & Harrington, W. (2017). API design patterns. O'Reilly Media.

Massé, M. (2011). REST API design rulebook. O'Reilly Media.

Pratama, F. R., Santoso, N., & Fanani, L. (2020). Pengembangan aplikasi e-commerce menggunakan payment gateway Midtrans. Jurnal Rekayasa dan Sistem Komputer, 8(3), 98–106.

Putra, D. A., & Sasmita, G. M. A. (2020). E-commerce marketplace petshop menggunakan integrasi Rajaongkir API dan iPaymu payment gateway API. Jurnal Teknologi Informasi, 12(2), 77–85.

Raharti, R., Irwan, M., & Nasution, P. (2024). Analisis dampak e-commerce terhadap pengaruh penjualan UMKM. Jurnal Riset Manajemen.

Safitri, R. K., & Putro, H. P. (2021). Implementasi REST API untuk komunikasi antara ReactJS dan NodeJS. Jurnal Teknologi dan Sistem Informasi, 5(2), 115–122.

Subramanian, V. (2019). Pro RESTful APIs: Design, build and integrate with REST, JSON, XML and JAX-RS. Apress.

Zammetti, F. A. (2020). Practical API design: Confessions of a Java framework architect. Apress.