

Implementation of RESTful API Web Services Architecture in Takeaway Application Development

Imam Ahmad
Faculty of Engineering and Computer Science
 Universitas Teknokrat Indonesia
 Bandarlampung, Indonesia
 imamahmad@teknokrat.ac.id

Emi Suwarni
Faculty of Economics and Business
 Universitas Teknokrat Indonesia
 Bandarlampung, Indonesia
 emisuwarni@teknokrat.ac.id

Rohmat Indra Borman
Faculty of Engineering and Computer Science
 Universitas Teknokrat Indonesia
 Bandarlampung, Indonesia
 rohmat_indra@teknokrat.ac.id

Asmawati
Department of Development Economics
 Universitas Abulyatama
 Aceh Besar, Indonesia
 asmawati@abulyatama.ac.id

Farli Rossi
Faculty of Engineering and Computer Science
 Universitas Teknokrat Indonesia
 Bandarlampung, Indonesia
 farli@teknokrat.ac.id

Yessi Jusman
Department of Electrical Engineering, Faculty of Engineering
 Universitas Muhammadiyah
 Yogyakarta, Yogyakarta, Indonesia
 yjusman@umy.ac.id

Abstract— The Covid-19 that hit the world had an impact on the economy, especially in the trade sector, one of which was experienced by Small and Medium Enterprises (SMEs). Hanura Takeaway (Haway) is an SME engaged in the delivery of goods and food. To facilitate transactions for goods and food delivery services, it is necessary to develop applications that simplify the transaction process. In developing web services, it is necessary to exchange data that is accessed via standard internet protocols. Therefore, we need a web service in developing this application. Implementing a RESTful API web service will certainly facilitate the development of software applications outside the system or with different programming languages or platforms. This research will develop web service architecture using RESTful API in Takeaway application. To optimize the URI, several parameters are used, including filtering, sorting, selection and pagination. The Takeaway application consists of a website as a backend and an Android-based as a frontend. From the test results based on the function method using the Postman application, it shows that the REST API Server built on the server has been running well. In testing the response time using the Apache JMeter application, the application shows a good response time. Meanwhile, the comparison of responses and requests to SOAP and REST architectures shows that REST takes faster time.

Keywords—application development, RESTful API, web services

I. INTRODUCTION

Currently, Indonesia and even the world are being shocked by an infectious virus originating from the province of Wuhan, China. This virus is considered very dangerous because the transmission is so fast. Then, WHO determined and named the virus as Coronavirus Disease of 2019 or abbreviated as Covid-19 [1]. The Covid-19 virus has caused millions of deaths worldwide. So that it becomes a frightening specter for every human being to interact at close range. Indonesia has taken steps to deal with the Covid-19 pandemic, one of which is by issuing health protocols, including: wash hands, wear masks, keep distance, stay away from crowds and reduce mobility.

The Covid-19 pandemic that has occurred has had an unfavourable impact on the economy, especially in the trade sector [2]. One of the trade sectors affected by the Covid-19 pandemic is Small and Medium Enterprises (SMEs). The Covid-19 pandemic affects the income of SMEs, this is due to the difficulty of running a business, especially in marketing and selling products that are carried out with an offline system

[3]. The decline in sales during this pandemic period was felt by SMEs in Hanura Village in Teluk Pandan District, Pesawaran Regency, Lampung Province. For businesses in the culinary field, it is advisable to provide delivery or take away services in order to reduce crowds. This opens up opportunities for delivery or delivery services. However, in Hanura Village there are no online motorcycle taxis or start-up sector delivery of goods that reach the area. This is due to the distance between Hanura Village and the city centre.

The development of technology today is balanced with the emergence of applications that help in human work. The use of applications in helping human work is considered to be able to increase the speed and efficiency of energy, time and costs without reducing the quality of the work done [4][5]. The delivery system has been implemented by several start-ups including Gojek, Grab, Maxim, Shopee and others. The delivery application is an application that can help customers to order products and have them delivered to their destination without having to go to the location and meet the seller [6]. The use of online delivery applications is considered effective in helping buyers and sellers because it facilitates transactions and can be done by buyers wherever and whenever [7]. Businesses run using applications can increase marketing reach to get new customers [8]. Web applications are currently widely used in building information systems. Website development that is used for backend or web view on other platforms, for example Android, has faced many obstacles, including interoperability and integrating different systems. So it takes the implementation of a web architecture that can overcome this.

Currently, many web services have been developed using the REST (Representational State Transfer) architecture. Where the architecture uses the existing functionality of the application layer protocol Hypertext Transfer Protocol (HTTP) [9]. The existence of REST has resulted in an increase in the use of websites that use this architecture when compared to the use of traditional web services with Simple Object Access Protocol (SOAP). Large companies, including Twitter, Amazon, etc., have used REST-like interfaces for their services with Application Programming Interfaces (API) documentation. REST implementation facilitates the development of software applications outside the system, because it offers the use of different programming languages or platforms [10].

REST which is an architecture for the application of web services in applying the concept of switching between states [11]. The state here can be described as a browser requesting a web page, on the server side it will send the current state of the web page to the browser. The REST API allows various systems to communicate and send/receive data in a very simple way [12]. Inside the RESTful API there is a REST client that can access data/resources on the REST server where each resource will be distinguished by a global ID or URIs (Universal Resource Identifiers). This makes the RESTful API very suitable to be applied to startup websites that are integrated with smartphones, one of which is the Takeaway application. Despite this trend, there are still no standards regarding RESTful web service development, especially on URI resource optimization URIs.

This research will develop web services architecture using RESTful API in Takeaway application. The Takeaway application consists of a website as a backend and an Android-based as a frontend. The RESTful API implementation is used to provide convenience in combining functions to develop application programs without relying on the same operating system, programming language, or database because this web service communicates using universal standard data formats, namely XML and JSON. For development standards in optimizing URI resources, this paper will use parameters including filtering, sorting, selection and pagination.

II. RELATED WORK

Currently, the development of web services using REST services has been widely used, and even provides available web Application Programming Interfaces (API) [10]. Several studies show that developers have switched from Simple Object Access Protocol (SOAP) to Representational State Transfer (REST) web services, as a means of website or application services [10][13]. REST is considered to provide easy access to data resources [11]. The REST architecture was introduced in 2000, by Thomas Fielding, and is based on the principles that underpin the World Wide Web [14]. Several studies have used the RESTful API in the web service architecture on the developed web [15][16][17][18]. In research [19], RESTful API can provide convenience in combining functions to develop application programs without relying on operating systems, programming languages, or databases with standard formats, namely XML and JSON. In this research, analyze and compare the use of Simple Object Access Protocol (SOAP) and REST. From the test data that for web services based on SOAP and REST API, it is shown that REST has a better performance than SOAP for checking requests and responses for web services.

In research [19], shows that two thirds of companies use REST services. The paper states that JSON is more general than XML and the most frequently used programming languages to implement REST services are Java, JavaScript, C#, Python, and Ruby. The REST interface relies exclusively on Uniform Resource Identifiers (URIs) to detect and interact with resources in the form of Hypertext Transfer Protocol (HTTP) [20]. A URI is a REST service that provides the location and name of a resource, which serves as a unique resource identifier. Predefined use of HTTP in describing the type of operation to be performed on the selected resource (for example, GET to retrieve, DELETE to delete). REST has become the de facto standard way of offering services on the Web [21][22]. However, REST is just an architectural style, provided without standard specifications. This implies that

some decisions have to be made by the developer when exposing his service, for example, which URI design scheme to use. In research [10], highlighting best practices for RESTful web service design. There are several practices that can be applied to optimize URIs, one of which is by using several parameters, including filtering, sorting, selection and pagination. The research was conducted to develop a RESTful API architecture by implementing development standards in optimizing URI resources, by applying filtering, sorting, selection and pagination parameters.

III. METHODOLOGY

In the development of the Takeaway application for SME Haway, an extreme programming (XP) system development approach is used. XP is a software development target for small to medium scale software and this method can also be used for developing systems with unclear requirements or very fast changes to requirements [23]. XP method is a method that is responsive to changes [24], this is shown in XP there are iterations that can be done repeatedly as needed. The stages in this research are presented in the chart in Figure 1 below.

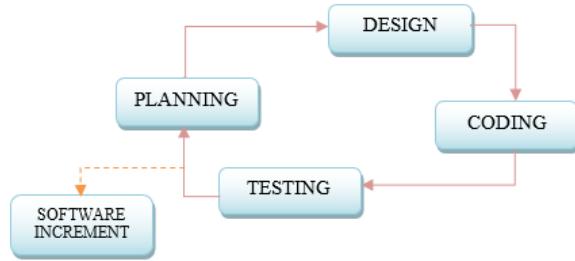


Fig. 1. Research Stages

A. Planning

The planning stage begins with understanding the business context of the application, defining the output, existing features in the application, the function of the application created, and the application development flow [25]. It can be said that this stage determines the overall functionality that will be developed in the system [26].

Planning activities begin by listening to customers through interviews, it aims to collect information about the needs of the system to be built so that the business context of the system to be built can be known. The results of listening to customers are then compiled in the form of a functional requirements analysis. Functional requirements are needed to find out what processes can be carried out by the system, as well as who can use the system that was built [27].

Takeaway application development in the form of a web application for the backend and an Android application as a frontend that can be accessed via a smartphone that can be used to order food or goods by customers and the driver will take the goods from the seller and then deliver it to the consumer. All transactions and reporting will be managed by the system. There are three users who use this Takeaway application, namely the admin manager, driver and the general public. Here are the functional requirements of the Takeaway application that will be built:

- 1) MSE Haway admin managing orders, promotions and data collection.
- 2) Drivers can receive orders, view income, view delivery folders.

- 3) The public as general users can place orders, view the delivery map and view delivery costs.

B. Design

The next step is to design the system. This Takeaway application is built on a website and Android, connecting users via the internet, then for order management and promotion arrangements through a website managed by the admin. The data is stored in a cloud database on the server. Rest API (Remote and Configuration) technology makes it easy to access the database through several media interfaces and programming languages.

REST can be described as when a user accesses a web address, the browser will request a web page and then the web server will send the state of the web page to the browser. This method is oriented towards information resources and has an output in the form of XML or JSON. Figure 2 below is a web service architecture design using a RESTful API that will be developed.

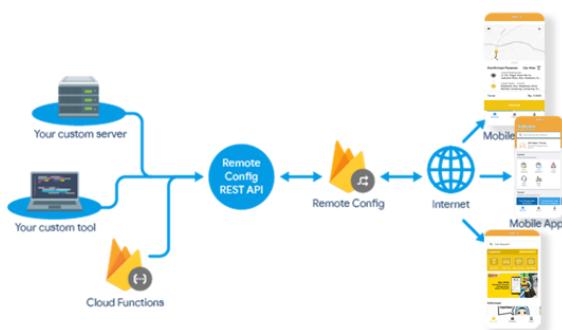


Fig. 2. Web Service Architecture Using Developed RESTful API

From Figure 2, the client will make an HTTP request to the web server to request data and it will be handled by the API (Application Programming Interface). Then the API confirms the request from the client, and forwards the request to the web server. Next, the server on the web gets the request from the client and provides the intended data to the API and then passes it to the client in JSON (JavaScript Object Nation) format.

C. Coding

The system design implemented in this paper uses the PHP programming language which is a multiplatform programming language using RESTful web service technology, so PHP can be run on all computers and smartphones and data distribution will be easier by using Restful web service. The database used in this study is MySQL which is also an open source database that can be run on various types of computers and smartphones so that the application of this system is not limited to certain device specifications.

D. Testing

In the testing phase, three approaches were carried out, namely testing the function of the RESTful API method and testing the response time and comparing responses and requests from the use of SOAP and REST. Testing RESTful API methods using the Postman app. This test will test the GET function, if the data is displayed on the Postman application in JSON format then the REST API Server built on the server has been successful. To test the response time using the Apache JMeter application with a total sample access of

50 users. Meanwhile, to test the response time and compare responses and requests from the use of SOAP and REST by accessing the backend built with SOAP and REST.

IV. RESULTS AND DISCUSSION

Web service has two methods that are oriented towards information resources and information services, namely Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). Many companies have used the SOAP method, but currently the REST method is the most dominant used for web service development because it is more effective in development and use. SOAP response format is only in XML form, in contrast to REST which can provide responses in the form of XML, JavaScript Object Notation (JSON) or other text formats. Building services with web services describes the business process flow of each section. This service is built based on the required parameters of each part directly with other application software using XML-based messaging via internet protocol. Figure 3 below is the system architecture that was built.

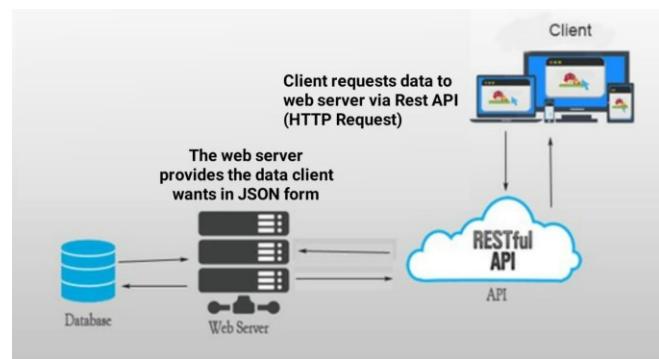


Fig. 3. RESTful API RESTful API Architecture Developed

From Figure 3 above, the client makes an HTTP request to the web server to request data and it will be handled by the API. Then the API confirms the request from the client, and forwards the request to the web server. Next, the web server receives a request from the client, then provides the intended data to the API and forwards it to the client in JSON (JavaScript Object Nation) format. Messages sent via JSON will be returned by the REST request and response elements. In accessing REST resources, we also use URLs where there are methods used, with defaults GET, PUT, DELETE and POST. The following is an example of using the GET, PUT, DELETE and POST methods.

GET, the method used to group HTTP in retrieving or reading data. POST, method for HTTP group to create new item/resource. This group of methods usually does not return output which is sometimes called a procedure. PUT, in its use methods for HTTP groups to update existing items/resources. While DELETE, the method used for this HTTP verb group is to delete an existing item/resource. In data processing on the backend, it is usually in the form of input, change, delete, and view data processes. The API structure for features such as managing data can be exemplified in table 1.

TABLE I. API STRUCTURE FOR DATA PROCESSING

Component	Data Administration	Data View
Method	Inputs: POST / DELETE Change: POST / PUT Delete: DELETE / POST	GET
Header Params	Authorization: Bearer <token>	Authorization: Bearer <token>

Component	Data Administration	Data View
URL params	Required: id_bs = [integer], id_bs = 1 Optional: <pre>{ Data : { Name : [string] }, { Data : { Name : "Budi" } } }</pre>	Required: id_bs = [integer], id_bs = 1 Optional: <pre>{ Data : { Name : [string] }, { Data : { Name : "Budi" } } }</pre>
Success Response	Code : 200 Status : true Message : "success data displayed" Result : {id=2}	Code : 200 Status : true Message : "success data displayed" Result : { <pre> Data : { Id : 2 Store_Name: "Donat Hanura" } }</pre>
Error Response	Code : 404 Status : false Message : "Unauthorized" Result : null	Code : 404 Status : false Message : "Unauthorized" Result : null

In the input data processing process, change and delete generally use the POST, PUT, and DELETE API methods although in fact you can also use the GET method. However, when using the GET method for the administrative process, there are weaknesses, namely the data parameters that are sent cannot use params data, but must use url params. The difference between using url params and data params is that if you use url params, the parameter names and data sent will be visible in the URL, whereas if you use params data the parameter names and data sent will not be visible in the URL. The weakness when using the GET method for the data administration process is that the data can be viewed at the URL and it is not safe. On the other hand, the process of viewing data generally uses the GET API method. To decide whether to use the GET or POST method, it is necessary to know in advance whether there will be changes to the data on the server or not. If, there will be data changes then use POST, but if there will be no data changes then use GET.

In a REST developer, it is important to set standardization in its development, especially in URIs. URIs can be further classified as trackers, names, or both. URIs, in addition to identifying a resource, provide a way to find a resource by describing its main access mechanism. For URI optimization, the resource can be expanded with several parameters, including:

1) Filtering

It is used to filter good resource information its attributes as well as a custom query language can be used. To choose one of the variants in the filtering depends on the strength of the expression needed to filter the information.

```
GET /profiles?filter=
(competencies=java%20and%20
```

certificates=MCSE_Solutions_Expert)

The code describes how a specific user group can be retrieved with a specific query.

2) Sorting

In sorting information, it is necessary to correct the attribute list which is then separated by commas and "sort" for URI parameters followed by the "+" symbol which refers to the prefix for ascending order and the "-" sign for descending order. This is so that sorting can occur.

```
GET /profiles?sort=-order,+rating
```

The code above explains that information can be ordered by using certain attributes.

3) Selection

In selecting information in the form of attributes, it reduces the size of the transmission over the network by responding to the requested information. In selection you can use a comma-separated list of attributes and a term field as URI parameters.

```
GET /profiles?fields=order_id, location
```

The code above explains how the required information can be selected prior to delivery.

4) Pagination

It takes the process of dividing and numbering the pages in the document, so that information on references to the next and previous pages exists, as well as to the first and last pages. As URI parameters, offset and limit are recommended, where the former identifies the virtual page and the latter describes the amount of information on the virtual page. Default values for offset and limit cannot be provided because they depend on the information to be sent to the client.

```
GET /profiles?offset=0&limit=15
```

The above code describes the request with pagination in the resource profile

This Takeaway application is built on the website and Android, connecting users via the internet, then for order management and promotion settings through a website managed by the admin. Admin can manage all their needs on the backend website. To start the application, the general public first opens the application and registers. After registering, users can search for products or food to be ordered. Figure 4 below is a view of the application that has been built and has been used.

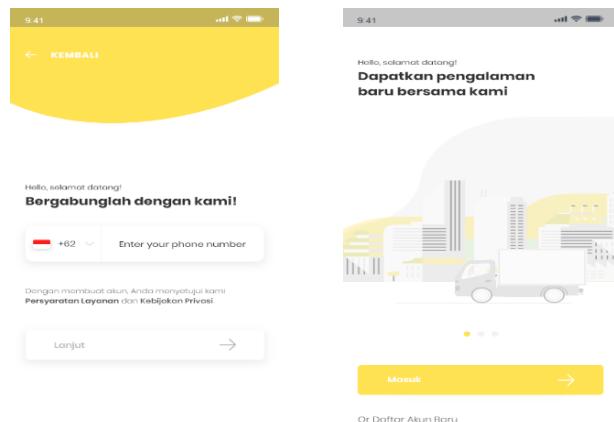


Fig. 4. Home Menu Display and User Registration

When the general public orders goods, the system will look for the driver closest to the location. After the driver receives the order, the driver will see the location for picking up the goods and the fees to be paid to the driver, as shown in Figure 5 below.

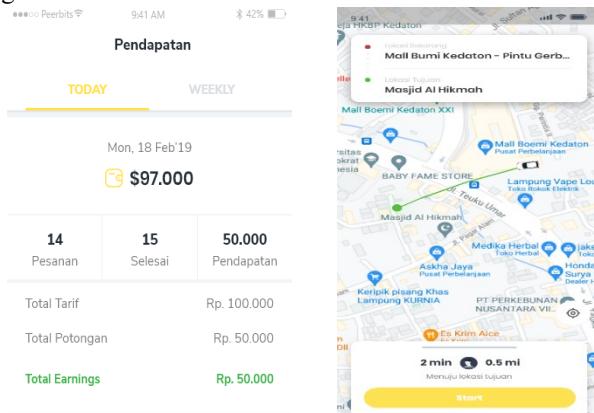


Fig. 5. Driver Menu Display View Map and Revenue

The system developed uses a server with a Virtual Private Network (VPN). Servers built using cloud servers through the website <https://www.exabytes.co.id/>. The server used has specifications: 20Gb SSD Disk Space, 1Gb Memory, 1TB Data Transfer, 100Mbps Bandwidth. In the testing phase, three approaches are used, namely testing the function of the RESTful API method, testing the response time, and comparing responses and requests from the use of SOAP and REST. The following tests are performed:

1) Testing the functionality of the RESTful API method

Tested RESTful API functionality using the Postman App. Postman can support multiple parameters, including GET, POST, PUT, DELETE. The parameter used in this research is GET. Testing is done by entering the GET parameter to display some data. When the data is displayed in the Postman application in JSON format, the REST API Server built on the server has succeeded. Then Postman will display the results in the form of response time. If seen in Figure 7, the GET function can function properly and the response time obtained for displaying data is 137 ms.

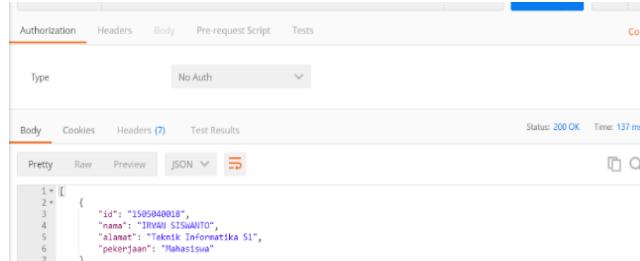


Fig. 6. GET Method Testing with Postman

2) Test the response time

Response time testing is done using the Apache JMeter application, this application is a desktop application that works on the Java VM which functions to measure system performance and load-test applications for the client server. Testing is done by entering a sample of 50 access users. This means that there are 50 users accessing the app at the same time. The following are the results of the response time testing with 50 users using the Apache JMeter application.

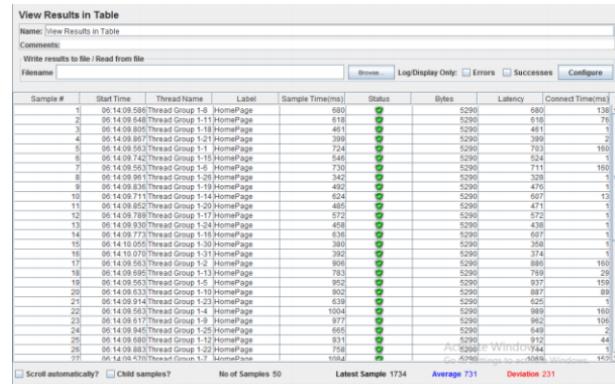


Fig. 7. Testing Response Time with Apache JMeter

Based on the results obtained, it is concluded that in measuring HTTP Requests with a sample of 50 users successfully made requests in 1 time, none of them failed. When making a request, the average response time obtained is 731 milliseconds or about 0.7 seconds.

3) Comparison of responses and requests from the use of SOAP and REST

To compare the use of SOAP and REST, access to the backend built with SOAP and REST will be carried out. The access used consists of POST, GET and PUT. Testing is done by measuring the response and request from each architecture by accessing the terminal, then the time given to serve the response and request from the client to the server is calculated. Figure 9, shows that when the client requests a request to the server using the terminal.

```

root@gilvy: /home/gilvy/todo-api
File Edit View Terminal Help
* Running on http://127.0.0.1:50800/ (Press CTRL+C to quit)
* Restarting with stat
127.0.0.1 [08/Jun/2019 19:48:57] "GET /todo/api/v1.0/tasks HTTP/1.1" 200 -
  Over and changing to '/home/gilvy/todo-api/app.py', reloading
* Restarting with stat
* Detected change in '/home/gilvy/todo-api/app.py', reloading
  Reloading worker...
127.0.0.1 [08/Jun/2019 19:49:01] "GET /todo/api/v1.0/tasks HTTP/1.1" 200 -
  Detected change in '/home/gilvy/todo-api/app.py', reloading
  [Intercepted KeyboardInterrupt, likely raised during interpreter shutdown]:
Exception in thread [main] (most recent call last):
  File "/usr/lib/python2.6/threading.py", line 532, in __bootstrap_inner
    File "/usr/lib/python2.6/threading.py", line 484, in run
      File "/usr/lib/python2.6/threading.py", line 408, in __call__
        File "/home/gilvy/todo-api/flask/lib/python2.6/site-packages/werkzeug/serving.py", line 400, in __call__
          File "/usr/lib/python2.6/socketserver.py", line 428, in server_close
            File "/usr/lib/python2.6/socketserver.py", line 108, in close
              TypeError: 'NoneType' object is not callable
  Restarts with stat
127.0.0.1 [08/Jun/2019 19:51:31] "GET /todo/api/v1.0/tasks HTTP/1.1" 200 -
127.0.0.1 [08/Jun/2019 19:53:01] "GET /todo/api/v1.0/tasks HTTP/1.1" 200 -
  
```

Fig. 8. GET Response for REST Server

This paper uses a sample of 10 responses and requests with access used consisting of POST, GET and PUT architectures on SOAP and REST. Table II below are the results of the comparison tests carried out.

TABLE II. COMPARISON OF RESPONSE AND REQUEST IN SOAP AND REST

Case Sample	Response And Request	
	SOAP	REST
1	2209	146
2	1809	154
3	1421	267
4	1749	128
5	2311	224
6	1852	265
7	1723	433
8	1982	198
9	2134	566
10	1876	432

From the data obtained in Table II, it shows that for web services that use SOAP and REST, it can be seen that REST has a better performance than SOAP, because it requires faster request and response time for web services.

V. CONCLUSION

The application of the RESTful API architecture in developing Takeaway applications for SME Haway can make it easier to combine functions to develop application programs without relying on the same operating system, programming language, or database because this web service communicates using a standard data format using JSON. To optimize the URI, several parameters are used, including filtering, sorting, selection and pagination. Based on the test results using Postman shows the function of each method can run with a good response time. In addition, response time testing with 50 samples of access tests carried out using the Apache JMeter application in a relatively good response time. Meanwhile, for testing the comparison of responses and requests from SOAP and REST architectures, it is seen that REST produces faster times. However, in the development of the Haway application, further testing has not been carried out regarding the user acceptance variable for the application. Future research may consider this to come up with useful applications.

ACKNOWLEDGMENT

Thanks are especially addressed to the Ministry of Education and Culture, RISTEK BRIN, which provided grant funds so that it can be carried out properly. Thanks are also conveyed to the Universitas Teknokrat Indonesia and Universitas Muhammadiyah Yogyakarta as well as those who have assisted in the implementation of the service.

REFERENCES

- [1] J. M. Kirigia, R. Nabi, and D. Karimi, "The Fiscal Value of Human Lives Lost from Coronavirus Disease (COVID-19) in China," *BMC Res. Notes*, vol. 13, no. 198, pp. 1–5, 2020.
- [2] A. Anggarani and W. P. Tyas, "Dampak Pandemi Covid 19 Terhadap Sektor Ekonomi Berimbang Pada Tingginya Angka Perceraian di Wilayah Kabupaten Bogor," *J. Abdimas*, vol. 7, no. 7, p. 3, 2021.
- [3] J. C. Hoekstra and P. S. H. Leeftlang, "Marketing in the era of COVID-19," *Ital. J. Mark.*, vol. 2020, no. 4, pp. 249–260, 2020.
- [4] A. D. Saputra and R. I. Borman, "Sistem Informasi Pelayanan Jasa Foto Berbasis Android (Studi Kasus: Ace Photography Way Kanan)," *J. Teknol. dan Sist. Inf.*, vol. 1, no. 2, pp. 87–94, 2020.
- [5] P. Prasetyawan, I. Ahmad, R. I. Borman, A. Ardiansyah, Y. A. Pahlevi, and D. E. Kurniawan, "Classification of the Period Undergraduate Study Using Back-propagation Neural Network," in *Proceedings of the 2018 International Conference on Applied Engineering, ICASE 2018*, 2018.
- [6] S. Sudewi and K. Khodijah, "Analisis Sistem Informasi Delivery Berbasis Web Mobile Pada Warung Makan Setia Kawan Pringsewu Dalam Upaya Peningkatan Omset Penjualan," *J. Aktual STIE Trisna Negara*, vol. 16, no. 2, pp. 127–135, 2018.
- [7] S. N. Rakhmah, R. Reza, and K. Novel, "Aplikasi Delivery Order Makanan dan Minuman Berbasis Web Pada Restoran Mang Kabayan," *J. Tek.*, vol. 11, no. 2, pp. 1109–1116, 2019.
- [8] R. I. Borman, Y. P. Putra, Y. Fernando, D. E. Kurniawan, P. Prasetyawan, and I. Ahmad, "Designing an Android-based Space Travel Application Trough Virtual Reality for Teaching Media," in *Proceedings of the 2018 International Conference on Applied Engineering, ICASE*, 2018.
- [9] P. Giessler, M. Gebhart, D. Sarancin, R. Steinegger, and S. Abeck, "Best Practices for the Design of RESTful Web Services," in *ICSEA 2015 : The Tenth International Conference on Software Engineering Advances Best*, 2015, pp. 392–397.
- [10] A. Neumann, N. Laranjeiro, and J. Bernardino, "An Analysis of Public REST Web Service APIs," *IEEE Trans. Serv. Comput.*, vol. 14, no. 4, pp. 957–970, 2021.
- [11] X. J. Hong, H. Sik Yang, and Y. H. Kim, "Performance Analysis of RESTful API and RabbitMQ for Microservice Web Application," in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 2018, pp. 257–259.
- [12] M. S. Alam, U. D. Atmojo, J. O. Blech, and J. L. M. Lastra, "A REST and HTTP-based Service Architecture for Industrial Facilities," in *Proceedings - 2020 IEEE Conference on Industrial Cyberphysical Systems, ICPS 2020*, 2020, pp. 398–401.
- [13] S. M. Sohan, F. Maurer, C. Anslow, and M. P. Robillard, "A study of the effectiveness of usage examples in REST API documentation," in *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 2017, vol. 2017-Octob, pp. 53–61.
- [14] R. T. Fielding and K. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," 2000.
- [15] S. Nugroho and A. Primadewi, "Penerapan Web Service untuk Integrasi Data Simperpus dan SIAK," *J. Komitika (Komputasi dan Inform.)*, vol. 4, no. 2, pp. 71–81, 2021.
- [16] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIX J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019.
- [17] E. Christianto and J. Santosa, "Perancangan Arsitektur RESTful pada Integrasi Data Refrensi Buku di Perpustakaan Daerah dan Perpustakaan Universitas (Studi Kasus: Kota Salatiga)," *Pustablibia J. Libr. Inf. Sci.*, vol. 2, no. 1, pp. 1–24, 2018.
- [18] A. Belkhir, M. Abdellatif, R. Tighilt, N. Moha, Y.-G. Guéhéneuc, and É. Beaudry, "An Observational Study on the State of REST API Uses in Android Mobile Applications," in *IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 2019, pp. 66–75.
- [19] B. P. Manuaba and E. Rudiastin, "API REST Web Service and Backend System of Lecturer's Assessment Information System on Politeknik Negeri Bali API REST Web service and backend system Of Lecturer's Assessment Information System on Politeknik Negeri Bali," in *The 2nd International Joint Conference on Science and Technology (IJCST)*, 2017.
- [20] B. Costa, P. F. Pires, F. C. Delicato, and P. Merson, "Evaluating a Representational State Transfer (REST) Architecture What is the impact of REST in my architecture?," in *Conference: Software Architecture (WICSA)*, 2014.
- [21] A. Gamez-diaz, P. Fernandez, and A. Ruiz-cortes, "An analysis of RESTful APIs offerings in the industry," in *Proc. Serv.-Oriented Comput.*, 2017, vol. 1867.
- [22] F. Palma, J. Gonzalez-huerta, M. Founi, N. Moha, G. Tremblay, and Y.-G. Gueheneuc, "Semantic Analysis of RESTful APIs for the Detection of Linguistic Patterns and Antipatterns," *Int. J. Coop. Inf. Syst.*, vol. 26, no. 2, pp. 1–37, 2017.
- [23] A. Shrivastava, I. Jaggi, N. Katoch, D. Gupta, and S. Gupta, "A Systematic Review on Extreme Programming," *J. Phys. Conf. Ser.*, vol. 1969, no. 1, 2021.
- [24] R. I. Borman, A. T. Priandika, and A. R. Edison, "Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan," *JUSTIN (Jurnal Sist. dan Teknol. Informasi)*, vol. 8, no. 3, pp. 272–277, 2020.
- [25] S. K. Purushothaman, N. Kashyap, D. Ha, S. Agarwal, S. Iqbal, and V. Behere, "Unified Approach towards Automation of any Desktop Web, Mobile Web, Android, iOS, REST and SOAP API Use Cases," in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology, ICCSDET 2018*, 2018.
- [26] P. Purmasari, "Development of Information Systems Using Extreme Programming Method," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 3, no. 2, pp. 95–99, 2018.
- [27] I. Ahmad, R. I. Borman, J. Fakhrurozi, and G. G. Caksana, "Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android," *J. Invotel Polbeng - Seri Inform.*, vol. 5, no. 2, pp. 297–307, 2020.