ILKOM Jurnal Ilmiah Vol. 16, No. 2, August 2024, pp. 125-133
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article Open Access (CC–BY-SA)

Enhancing RESTful API Authentication with Cryptography in Student Information Systems

Sucipto [a,1,*]; Muhammad Najibulloh Muzaki [a,2]; Jamilah Karaman [b,3]; Yahya Zakur [c,4]
[a] *Universitas Nusantara PGRI Kediri, Indonesia*
[b] *Universitas Muhammadiyah Ponorogo, Indonesia*
[c] *University of Mazandaran, Iran*
[1] *sucipto@unpkediri.ac.id;* [2] *m.n.muzaki@gmail.com;* [3] *jamilah@umpo.ac.id;* [4] *yahyazakur92@gmail.com*
** Corresponding author*

Article history: Received May 22, 2024; Revised July 05, 2024; Accepted July 14, 2024; Available online August 20, 2024

Abstract

Data integration in this era is necessary for building a valid information system. Data in an information system must have a concept that interacts with other systems. With the development of information systems, data storage will increase. Big data must be channeled with a supporting information system connected to the data center information system. This research develops an API-integrated system with increased security in Basic Authentication with Cryptography. This research uses the Linear Sequential Model method with increased API security in Basic Authentication with Cryptographic hashes. test results using the CURL Library obtained appropriate data, and response time testing obtained an average result of 0.0611 per second. Acceptance testing obtained a percentage of results of 78%, which was included in the excellent functioning category. The research found that the Rest API can integrate and validate data between information systems.

Keywords: Basic Authentication; RESTful API; Student Basic Data; Web Service

Introduction

The rapid development of technology makes human work can be completed quickly. The role of technology and technology makes processing data and information easier. Almost all fields require the role of technology in its management, especially in the field of educational administration management. Some systems have been widely created but data integration is still very lacking so that information systems that utilize data integration in running their business. It cannot be denied if modern technology is considered much more practical compared to conventional technology. The management of data and information in the administration of education must be managed properly. Not only that, the management of basic educational data that contains the identity of students becomes an important factor in the management so that business processes can run smoothly and optimally [1].

Techniques to facilitate and speed up the process of distribution of existing information, we can use information technology that has developed today. In this study, researchers tried to distribute account access information on each information system the university environment can connect. The connection of accounts on each information system will reduce invalid data because the main data input process comes from one information system source. The technology used is a web service based on the PHP programming language JSON with the RESTful API method. The reason for using JSON web services in this study is that JSON is a simple data type format, does not require storage space, and does not require large resources in transferring data either using only a slow network connection, does not require large processor resources on Android-based smartphones to decode the JSON format [2]–[4]. The use of API Techniques is expected to overcome repeated registration problems in information systems. Various problems that often occur include, the first in terms of students and the second in terms of staff. In terms of students, as for the problems that occur repeatedly when students want to ask for data to the database for various applications contained in the university. In terms of Staff, when students request to access the database staff must verify the data whether the student is eligible to access other information systems. This is intended to minimize the misuse of databases in universities [5]–[7]. To get satisfactory results in the efforts to apply JSON data to support the RESTful API technique in Web Service applications as a medium in application development, therefore, of course, there has been a lot of research that examines this subject, below are some of them are as follows:

Research on RESTful and Web Service was conducted by [8] discusses the implementation of RESTful API to integrate information systems and save bandwidth usage. The results of the study created a web service to bridge different platforms. Another study that discusses RESTful API was conducted by [9] about the implementation of RESTful API to integrate between servers and improve the speed of access to information systems. The results of the study resulted in an API and were tested with the SUS method, which gave excellent results, reaching 80.88 [8], [9]. Other research on RESTful API was conducted by [10] presenting the implementation of RESTful API in the creation

http://dx.doi.org/10.33096/ilkom.v16i2.2255.125-133

of a master data planogram. The results of the study resulted in a web-based data master application that users can use to enter the data needed in the creation of a planogram. Another study on RESTful API was conducted by [11] discussing the implementation of RESTful API as a data Synchronization tool between servers. The results of the study can Synchronize between servers with API techniques. Basic authentication is a simple method that has several disadvantages. Yoshiaki Kasahara's research explains that the model is no longer used [12].

Based on the description and several research references above, this research proposes improvements to previous research by utilizing cryptography in RESTful API authentication. This research uses the Linear Sequential Model method with increased API security in Basic Authentication with Cryptographic hashes. This research contribution to improving the security of RESTful APIs with cryptographic hash algorithms

Method

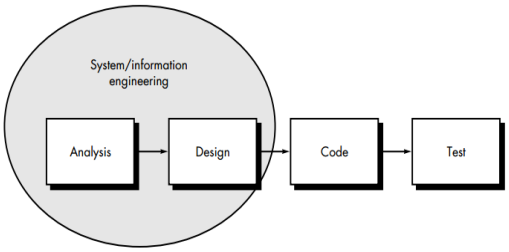


Figure 1. Linear Sequential Model

The study used the Linear Sequential Model (LSM). This method demonstrates a systematic, sequential approach to software development that begins at the system level and develops through analysis, design, coding, and testing [13]. The use of LSM is a suitable model for developing information systems [14]. Information system development involves adding feature modules and increasing security as carried out in this research, namely increasing authentication security on RESTful APIs with hash cryptographic algorithms [15], [16]. **Figure 1** illustrates a linear sequential model for software engineering. System information engineering and modeling become part of a business system. The work begins with establishing requirements for all system elements and then allocating some of these requirements to the software. Software systems must interact with other components such as hardware, people, and databases. Systems engineering and analysis include collecting requirements at the system level with a small number of top-level designs and analyses. Information engineering includes the collection of requirements at the strategic business level and the business area level.

1) Analysis

This process stage requires intensive requirements gathering and focuses specifically on software. To find out the needs, identify the following API needs used by the tracer study system. The data used in this research is tracer study system log data, which consists of 1575 records of 240 KiB. This data is processed to become RESTful API data.

2) Design

This process stage is carried out by software design. This is done with a RESTful API flow diagram. There are two flow schemes: the RESTful API Architecture Design in **Figure 2** and the Basic Authentication Enhancement flow design in **Figure 3**.

3) Code

This process stage is carried out by creating code. The design that has been made is translated into a programming language. The programming language used is PHP & uses the MariaDB database. At this stage, the RESTful API coding uses the Native PHP model with Basic Authentication security and additional security.

4) Testing

The Testing stage focuses on the logical internals of the software, ensuring that all statements have been tested, and on functional externals; That is, conducting tests to uncover errors and ensure that the specified input will produce actual results that match the required results. Testing using the Response Time Performance Comparison [17]–[19] and acceptance testing method [20]–[22]. The RT formula is in Equation 1:

$$RT = T_{end} - T_{start} \quad (1)$$

Where: "End Time" is when a system or entity completes a task or responds to a request. "Start Time" is when a request or task is first received or started.

Results and Discussion

The design in **Figure 2** describes the architecture of the web service to be built. The goal is to connect between SIAKAD server as the main server with the tracer study server as the client. The Main Server has a function that is as the main data center, one of which is student data. The main data will be connected through APIs with other systems in various information systems, one of which is in this study, namely the tracer study system. The RESTful API Design Solution selected to solve data communication problems is to use web services [23]. The web service to be built using the REST (Representational State Transfer) architecture with the protocol used is HTTP (Hypertext Transfer Protocol). The communication flow that occurs is that the client will send data or send data requests over HTTP Requests, then the server responds to the client using HTTP Response. RESTful API uses a basic authentication security system. RESTful API security through authentication in this RESTful API makes applications safer because API users can only access APIs with authentication accounts registered on the API system [24].

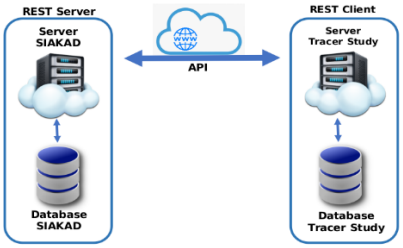


Figure 2. RESTful API Architectural Design

The RESTful API system in this study is known as the SYNC API information system. The main function of SYNC API is that users can perform user authentication access management to customize the RESTful API. In this API system, users can access the available modules to support easier and safer API access, including API usage tutorials, Administrator Contact Modules, User Modules, and System Access Log Modules. Users can access RESTful API functions in several methods, namely get, post, put and delete on the module provided by the selected API [25]. The concept in **Figure 2** shows the implementation model of integration between SIAKAD information systems and Tracer Study information systems [24]. The API design flow in **Figure 2** is described in **Figure 3** which is the flowchart of the system between SIAKAD information systems with tracer study. Sync API information system product design can manage access to data between information systems and validate access to each information system. Access validation is done when an information system performs an access request on an API. During the data request process, the SYNC API will Synchronize data and validate so that the data on the tracer study system and SIAKAD will be valid. RESTful API security uses basic authentication with additional security

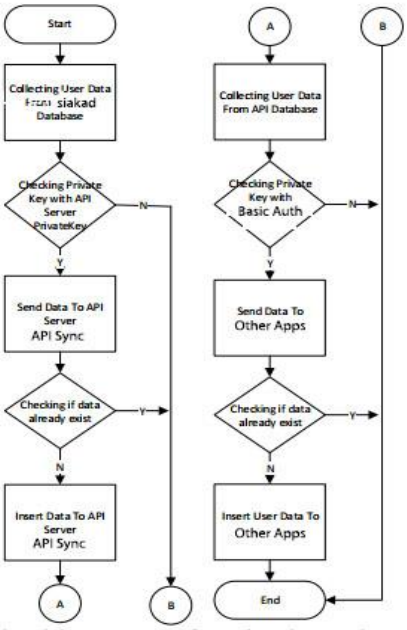


Figure 3. Sync API Flowchart Diagram

The flow in **Figure 3** shows the details of the RESTful API design in **Figure 2**. The Sync API system performs data retrieval on SIAKAD servers using SIAKAD RESTful API. Data retrieval in SIAKAD API through API authentication process. If key authentication is successful, the data can enter the Sync API database through the data validation process. There is another feature that is a Synchronization of account data and passwords if there is a password difference

between SYNC API and SIAKAD. The improvement offered is by storing the RESTful API authentication account in the storage database and encrypting the account password with a hash function algorithm, namely MD5.

Added security to Basic Authentication on the RESTful API system with a Digest Authentication security approach [26]–[28]. Additional security is carried out in Basic Authentication including protocols used using Secure Sockets Layer, the use of multiuser to be able to access the API by storing it in the system log, and the main thing is to add hash cryptocurrency security to the authentication password of the RESTful API account [29], [30].

Table 1. Result of Functional Analysis

Actor	Description
SIAKAD	Provide student account Collect validation input tracer study
Tracer Study	Collect student account Provide validation input tracer study
API Sync	Collect student account Collect validation input tracer study Distribute student account Distribute validation input tracer study

RESTful API in the form of Basic Authentication is widely used in various information system developments. This is based on convenience and the fact that information systems, especially web-based services, compete with similar information systems. RESTful API frameworks can work in the same way as typical web applications [31]. Implementation of basic authentication security with additional system security used for the needs of several student information systems in **Table 1** is produced regarding the features needed in other API systems [32]. Based on the results of functional analysis, there are three information system actors in this study, namely SIAKAD, Tracer Study, and API SYNC. SIAKAD is a system that provides basic data for all systems in this research object. SIAKAD and Tracer Study need each other data, meaning that the tracer study system requires data from SIAKAN and SIAKAD also takes data from tracer study used to validate students and alumni who have filled in data on the tracer study system. In addition, Tracer Study has a role as a provider of student and alumni validation data to fill out the study tracer form. Business processes covered by SIAKAD systems and tracer study systems are bridged by the Sync API. The Sync API provides both information systems to keep the data distributed between systems valid and secure [33], [34]. **Table 1** shows the results of functional analysis, showing the activity of each actor in the system.

Table 2. The Created Endpoints

Service	Endpoints	Method	Function
SIAKAD	/get_mahasiswa	GET	Show student account
	/get_mhs_id	GET	Show specific student account
Tracer Study	/tracer_del_id/{npm}	DELETE	Delete selected tracer study validation
	/patch_account	PATCH	Change student account
	/get_tracer	GET	Show tracer study validation
	/tracer	POST	Insert validation tracer study
	/get_tracer_id/{npm}	GET	Show specific tracer study validation

Table 2 shows the functions that work on the system. Web services are built using REST architecture with basic authentication security with additional security. The REST architecture consists of five HTTP methods. All methods in rest api including POST, GET, PATCH, DELETE, and PUT must go through enhanced basic security. POST is used to create new resources. GET is used to retrieve specific resources. PATCH is used to update specific resources. DELETE is used to delete a particular resource and PUT is used to replace all of a particular resource. Each endpoint is registered with the API Gateway using the API SYNC. The system is built using the RESTful API additional security HTTPs protocol and cryptographic hashes on the RESTful API account. The response provided from each request is sent in JSON form. The database used by all services is MariaDB. Then, the system is equipped with a security layer to protect data and communication processes between users and the system. Password protected using md5 algorithm. It meets three security criteria that are believed to be sufficient to protect the system. Basic authentication is used when a user wants to access an endpoint in the system. Secure API will ensure that data will only be accessed by certain accounts by comparing account claims. Next, the implementation of the API system is in **Figure 4**.

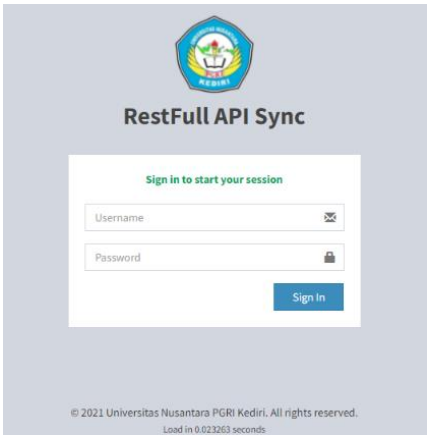


Figure 4. Sync API Access Login Page

In **Figure 4** displays an overview of the Sync API access home page. Steps to access the Sync API must be entered into the system to find out the RESTful API authentication account and how to use the RESTful API. Account access to the system is provided by the administrator.

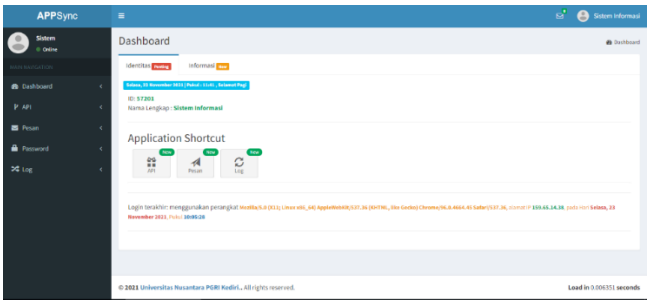


Figure 5. Sync API Dashboard Page

After the user logins, the initial dashboard will appear. In **Figure 5**, users can have access to several RESTful API system shortcuts. This shortcut can access the Sync API menu, send messages, and log access Sync API. RESTful API authentication is described in **Figure 6** and **Figure 7**.

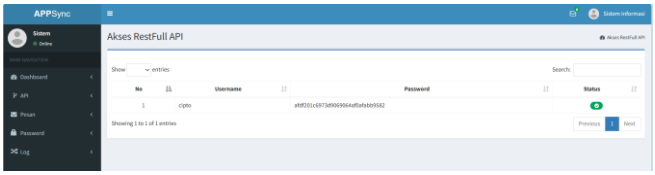


Figure 6.

Sync API

Account

In **Figure 6** is a listing page of accounts that can be used to access the Sync API. **Figure 6** can be accessed by clicking the Sync API menu. There is a username and password for access to the Sync API with Basic Authentication security. The green check status indicates the RESTful API account can be used.

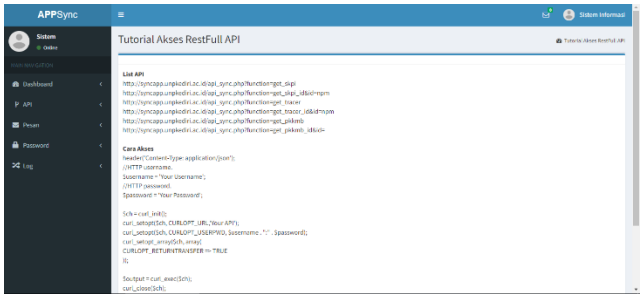


Figure 7. Sync API Access Tutorial Page

In **Figure 7** is a list of available Sync API along with API access procedures. **Figure 7** can be accessed by clicking the API menu then the API Tutorial. Users can use the API in multiple programming platforms. Among them are PHP, Android, iOS, phyton, and so on. In **Figure 8** is a trial of RESTful API access with Postman. Researchers conducted a trial on one of the RESTful APIs with Basic Authentication securities. The test results were successful by displaying data on the API.

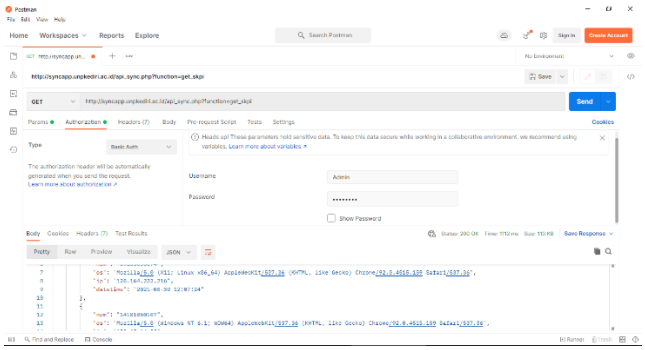


Figure 8. POSTMAN Access View

The last stage is testing. Testing was carried out using the response time method. The test was carried out 10 times with three test scenarios and use 1575 records. The first scenario is to access the RESTful API with basic authentication without additional security with 1 iteration, the second test is to access the RESTful API with basic authentication with additional security with 1 iteration, and the third test is to access the RESTful API with basic authentication with additional security with 60 iterations. The results can be seen in **Figure 9**.

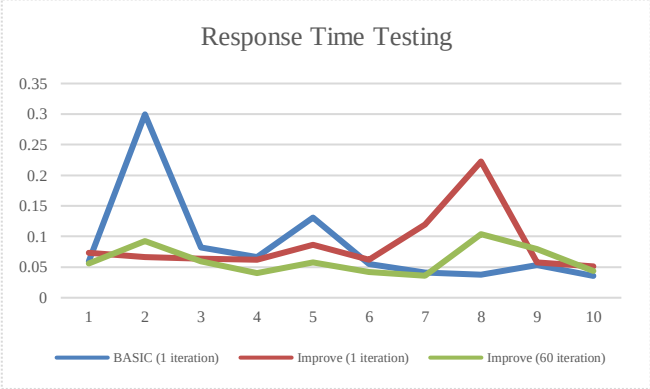


Figure 9. Response Time Testing

Results on average responsiveness were less than 0.0611 seconds. Further testing using methods using acceptance testing [35]. This test is done with a questionnaire method based on usability testing with variables of effectiveness, efficiency, and satisfaction [36]. Acceptance testing is done by distributing application tests to several respondents of prospective RESTful API users. The number of test data was 30 respondents. The test results of the Sync API information system prototype showed that all the function variables tested could run properly. The results of the function using the Black-Box testing method follow the communication and planning stages [37], [38]. System usability testing uses usability testing methods for levels of effectiveness, efficiency, and satisfaction. The results of the usability test are shown in **Figure 10**.

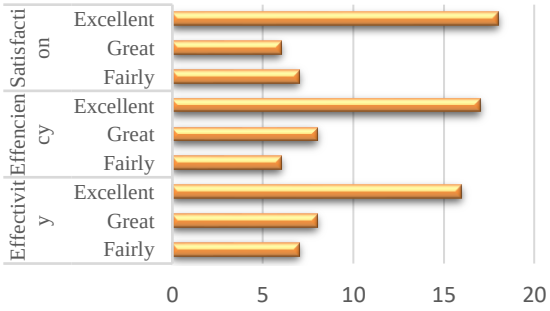


Figure 20. RESTful API Testing

The test used 30 respondents with 3 levels, namely effectiveness, efficiency, and satisfaction. In **Figure 9**, the test results showed that the results of RESTful API usage of 7 respondents were dissatisfied with the API, 8 respondents were satisfied with the API and 17 respondents were very satisfied with the API. The average percentage of testing effectiveness, efficiency, and satisfaction are that 22% of respondents have less response and 78% of respondents are satisfied with the RESTful API that has.

The significance of Response Time (RT) lies in its ability to quantify the speed at which the system reacts to user requests or input, as highlighted by Ahmad and Hasan in their 2021 study. Rapid real-time security measures can enhance security by minimizing the chances for attackers to exploit vulnerabilities through time-lapse attacks or denial-of-service attempts [39]. Within the realm of fast Usability RT, it enhances the user experience by guaranteeing that the system's interaction seems seamless and prompt. This is crucial for assuring user pleasure and productivity. The results obtained in this research have several advantages compared to previous studies, where previous studies only focused on developing easy access to RESTful APIs and did not highlight specific security issues in authentication [37], [40]. The RT testing capability is also better than other models proposed in previous research, namely with results below 0.0611 seconds [41]. However, this research has limitations, and there is still a need for improvement in terms of capabilities in integration between devices, such as testing on Internet of Things (IoT) integration [41].

Conclusion

Based on the research results, it can be concluded that Api Web Service with a RESTful API architecture with increased security of basic authentication hash algorithms has succeeded in achieving its aims and objectives where it can be used to integrate various information systems with valid data results. Response time testing results with additional improvements with an average responsiveness of less than 0.0611 seconds. The test results using the Acceptance Test method obtained a result percentage of 78%, which was included in the good functioning category. The limitation of this research is that it has not tested the increase in RESTful API integration with other devices such as IoT. Future research will focus on improving the development of cryptographic algorithms by testing between IoT devices to integrate information systems with IoT devices.

Acknowledgement

The author would like to thank the Institute for Research and Community Service at Universitas Nusantara PGRI Kediri which has provided financial support for this research through the Basic Research Scheme.

References

[1] P. L. L. Belluano, "Penerapan Sistem Replikasi Dan Integrasi Basis Data Terdistribusi Pada Pangkalan Data Pendidikan Tinggi (PDPT)," *ILKOM Jurnal Ilmiah*, vol. 9, no. 1, pp. 42–48, 2017.

[2] S. Sucipto, E. K. Dewi, N. C. Resti, and I. H. Santi, "Improving The Performance of Alumni Achievement Assessment by Integrating Website-Based Tracer Study Information Systems and Telegram API," *Teknik*, vol. 41, no. 1, pp. 72–77, 2020, doi: 10.14710/teknik.v41i1.25307.

[3] M. A. Baazizi, D. Colazzo, G. Ghelli, C. Sartiani, and S. Scherzinger, "Negation-closure for JSON Schema," *Theor Comput Sci*, vol. 955, p. 113823, Apr. 2023, doi: 10.1016/J.TCS.2023.113823.

[4] T. Whairit, B. Phadermrod, and V. Attasena, "JINDEX: JSON and index search system for plant germplasm database," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101701, Sep. 2023, doi: 10.1016/J.JKSUCI.2023.101701.

[5] E. Chavarriaga, F. Jurado, and F. D. Rodríguez, "An approach to build JSON-based Domain Specific Languages solutions for web applications," *J Comput Lang*, vol. 75, p. 101203, Jun. 2023, doi: 10.1016/J.COLA.2023.101203.

[6] M. Pirani, A. Cucchiarelli, and L. Spalazzi, "Paradigms for database-centric application interfaces," *Procedia Comput Sci*, vol. 217, pp. 835–845, 2023, doi: 10.1016/j.procs.2022.12.280.

[7] L. Diop, C. T. Diop, A. Giacometti, and A. Soulet, "Pattern on demand in transactional distributed databases," *Inf Syst*, vol. 104, p. 101908, 2022, doi: 10.1016/j.is.2021.101908.

[8] A. B. Warsito, A. Ananda, and D. Triyanjaya, "Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service," *Technomedia Journal*, vol. 2, no. 1, pp. 26–36, 2017, doi: 10.33050/tmj.v2i1.313.

[9] B. W. Putra, A. Saputra, R. Sanjaya, and D. Kurniawan, "Implementasi Framework CodeIgniter dan Restful API pada Sistem Informasi Manajemen Tugas Akhir 1st," *Prosiding Annual Research Seminar*, vol. 5, no. 1, p. 309, 2019.

[10] E. Susanti, "Implementasi RESTful API dalam Pembuatan Master Data Planogram Menggunakan Framework Flask (Studi Kasus: PT Sumber Alfaria Trijaya, Tbk)," *Techno.Com*, vol. 19, no. 3, pp. 295–307, 2020, doi: 10.33633/tc.v19i3.3468.

[11] F. W. Handono, H. Nurdin, F. B. Siahaan, H. Sugiarto, and I. Chaidir, "Jurnal Mantik Mysql Database Syncronization Using Restful Webservice Api PT . Minori," *Jurnal Mantik*, vol. 5, no. 36, pp. 855–859, 2021.

[12] Y. Kasahara, "End of Basic Authentication and Migration to Modern Authentication for Exchange Online," *Proceedings ACM SIGUCCS User Services Conference*, pp. 32–35, 2023, doi: 10.1145/3539811.3579560.

[13] Roger S. Pressman, *Software Quality Engineering*, 2010th ed., vol. 9781118592. Wiley, 2014. doi: 10.1002/9781118830208.

[14] M. Bianchi, G. Marzi, and M. Guerini, "Agile, Stage-Gate and their combination: Exploring how they relate to performance in software development," *J Bus Res*, vol. 110, pp. 538–553, Mar. 2020, doi: 10.1016/J.JBUSRES.2018.05.003.

[15] S. Pradhan and V. Nanniyur, "Large scale quality transformation in hybrid development organizations – A case study," *Journal of Systems and Software*, vol. 171, p. 110836, Jan. 2021, doi: 10.1016/J.JSS.2020.110836.

[16] Sucipto *et al.*, "Hidden Treasures of Kediri's Medicinal Plants: A Collaborative Effort to Map and Validate Authentic Information Using Innovative QR Code Security and Cryptography," *IOP Conf Ser Earth Environ Sci*, vol. 1242, no. 1, p. 012036, Sep. 2023, doi: 10.1088/1755-1315/1242/1/012036.

[17] R. V. Rochim, A. Rahmatulloh, R. R. El-Akbar, and R. Rizal, "Performance Comparison of Response Time Native, Mobile and Progressive Web Application Technology," *Innovation in Research of Informatics (INNOVATICS)*, vol. 5, no. 1, pp. 36–43, May 2023, doi: 10.37058/INNOVATICS.V5I1.7045.

[18] S. Jalil, S. Rafi, T. D. Latoza, K. Moran, and W. Lam, "ChatGPT and Software Testing Education: Promises & Perils," *Proceedings - 2023 IEEE 16th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2023*, pp. 430–437, 2023, doi: 10.1109/ICSTW58534.2023.00078.

[19] K. Xue, V. Yaneva, C. Runyon, and P. Baldwin, "Predicting the Difficulty and Response Time of Multiple Choice Questions Using Transfer Learning," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 193–197, 2020, doi: 10.18653/V1/2020.BEA-1.20.

[20] M. Zheng and D. Bender, "Evaluating outcomes of computer-based classroom testing: Student acceptance and impact on learning and exam performance," *Med Teach*, vol. 41, no. 1, pp. 75–82, Jan. 2019, doi: 10.1080/0142159X.2018.1441984.

[21] C. H. Wang, W. J. Lo, and M. J. J. Wang, "Usability evaluation of augmented reality-based maintenance instruction system," *Human Factors and Ergonomics in Manufacturing & Service Industries*, vol. 32, no. 3, pp. 239–255, May 2022, doi: 10.1002/HFM.20942.

[22] R. Silva, A. C. Lima, E. Andrade, A. I. Martins, and N. P. Rocha, "Heuristic Evaluation of the Usability of a Mechanical Ventilator Interface through a Simulator," *Procedia Comput Sci*, vol. 219, pp. 1232–1239, 2023, doi: 10.1016/j.procs.2023.01.406.

[23] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, "Restful API Architecture Based on Laravel Framework," *J Phys Conf Ser*, vol. 910, no. 1, 2017, doi: 10.1088/1742-6596/910/1/012016.

[24] H. Mezni, "Web service adaptation: A decade's overview," *Comput Sci Rev*, vol. 48, p. 100535, May 2023, doi: 10.1016/J.COSREV.2023.100535.

[25] A. D. K. B. W. Putra, A. Saputra, M. R. Sanjaya, "Enabling Collaboration of CodeIgniter Framework and RESTful API for Utilize Web Mobile Interface Implemented on Final Project Management System," vol. 172, no. Siconian 2019, 2020, doi: 10.2991/aisr.k.200424.080.

[26] T. Telang, "MicroProfile JSON Web Tokens and Jakarta Security," *Beginning Cloud Native Development with MicroProfile, Jakarta EE, and Kubernetes*, pp. 183–212, 2023, doi: 10.1007/978-1-4842-8832-0_8.

[27] M. Baker, "Authentication and Authorization," *Secure Web Application Development*, pp. 287–349, 2022, doi: 10.1007/978-1-4842-8596-1_10.

[28] A. R. Khan, L. Khalid, A. A. Finance, and S. Arabia, "A Brief Review on Cloud Computing Authentication Frameworks," *Engineering, Technology & Applied Science Research*, vol. 13, no. 1, pp. 9997–10004, Feb. 2023, doi: 10.48084/ETASR.5479.

[29] H. Subramanian and P. Raj, *Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs*. Packt Publishing Ltd, 2019.

[30] K. H. Huynh and J. Kerssens, "Security Evaluation on Amazon Web Services' REST API Authentication Protocol Signature Version 4," *rp.os3.nl*.

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

[32] J. A. Suthendra and M. A. I. Pakereng, "Implementation of Microservices Architecture on E-Commerce Web Service," *ComTech: Computer, Mathematics and Engineering Applications*, vol. 11, no. 2, pp. 89–95, 2020, doi: 10.21512/comtech.v11i2.6453.

[33] A. Arcuri, "RESTful API automated test case generation," *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security, QRS 2017*, no. 1, pp. 9–20, 2017, doi: 10.1109/QRS.2017.11.

[34] M. A. Yaghoub-Zadeh-Fard and B. Benatallah, "API2CAN: a dataset & service for canonical utterance generation for REST APIs," *BMC Res Notes*, vol. 14, no. 1, pp. 1–3, 2021, doi: 10.1186/s13104-021-05593-w.

[35] S. Parsa, M. Zakeri-Nasrabadi, and B. Turhan, "Testability-driven development: An improvement to the TDD efficiency," *Comput Stand Interfaces*, vol. 91, p. 103877, Jan. 2025, doi: 10.1016/J.CSI.2024.103877.

[36] S. Sucipto and J. Karaman, "Integration of Legalization Information System Web-Based using Shipping API and Telegram API," *JUITA: Jurnal Informatika*, vol. 8, no. 2, p. 131, 2020, doi: 10.30595/juita.v8i2.7104.

[37] D. Felicio, J. Simao, and N. Datia, "RapiTest: Continuous Black-Box Testing of RESTful Web APIs," *Procedia Comput Sci*, vol. 219, pp. 537–545, Jan. 2023, doi: 10.1016/J.PROCS.2023.01.322.

[38] O. Banias, D. Florea, R. Gyalai, and D. I. Curiac, "Automated specification-based testing of REST APIs," *Sensors*, vol. 21, no. 16, 2021, doi: 10.3390/s21165375.

[39] N. Ahmad and S. M. R. Hasan, "A new ASIC implementation of an advanced encryption standard (AES) crypto-hardware accelerator," *Microelectronics J*, vol. 117, p. 105255, Nov. 2021, doi: 10.1016/J.MEJO.2021.105255.

[40] A. Borrego, M. Bermudo, F. Sola, D. Ayala, I. Hernández, and D. Ruiz, "Silence — A web framework for an agile generation of RESTful APIs," *SoftwareX*, vol. 20, p. 101260, Dec. 2022, doi: 10.1016/J.SOFTX.2022.101260.

[41] F. Palma, T. Olsson, A. Wingkvist, and J. Gonzalez-Huerta, "Assessing the linguistic quality of REST APIs for IoT applications," *Journal of Systems and Software*, vol. 191, p. 111369, Sep. 2022, doi: 10.1016/J.JSS.2022.111369.