



Analysis of the implementation of Restful (Representational State Transfer) web services for data integration between information systems

I Nyoman Alit Arsana^{1*}, I Made Pustikayasa², Ni Wayan Purnawati³

^{1*,3}Information Technology, IAHN-TP Palangka Raya, Indonesia

²PGSD, IAHN-TP Palangka Raya, Indonesia

ARTICLE INFO

ABSTRACT

Article history:

Accepted Jul 29, 2023
Revised Aug 17, 2023
Accepted Aug 30, 2023

Keywords:

Information System;
JSON;
REST;
Slim Micro Framework;
Web Service.

The Library Information System and the SAT-IKA Information System (Sistem Infomasi Pangkalan Data Terintegrasi IAHNTP), which have their respective databases or are not integrated with each other, cannot be used together and can cause duplication and data inconsistency. Application of Web service technology enables interaction between heterogeneous software and can support data integration between information systems. The approach used is REST (Representational State Transfer) Web Service technology with JSON (Java Script Object Notation) format. Slim Micro Framework is used to code web applications and APIs (application programming interfaces) quickly and easily. The result of this research is data integration in each information system using a REST Web service with JSON data format so that data consistency is maintained and data redundancy is avoided.

This is an open access article under the [CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/) license.



Corresponding Author:

I Nyoman Alit Arsana,
Faculty of Dharma Duta and Brahma Widya/Information Technology,
Institut Agama Hindu Negeri Tampung Penyang Palangka Raya (IAHN-TP),
Jalan G.Obos X, Palangka Raya, Kalimantan Tengah, 73112, Indonesia.
Email: alitarsana@iahntp.ac.id

1. INTRODUCTION

Currently, the Palangka Raya State Hindu Institute of Religion (IAHN-TP) is developing an information system called SAT-IKA (Integrated Database Information System, IAHNTP). This information system provides document or report services for elements of the IAHN-TP campus leadership. This information system will be integrated with other information systems. One of the information systems that will be integrated is the library information system. The library information system uses SLiMS 8 (Senayan Library Management System) software and has its own database. The existence of differences in database and master data (basic data) between SAT-IKA and the Library Information System allows data redundancy and data duplication because they are stored in different locations and the same master data cannot be used together. The solution that can be used in this case is to perform data integration. Data integration is the process of combining heterogeneous data (two or more data sets) originating from several systems or sub-systems and connecting them in one data store (Arsana, 2020; Lenzerini, 2002; Valle, 2012). Data integration can be used to facilitate data sharing and analysis (Valle, 2012).

Data integration can be done by implementing a web service. Web services enable interaction between distributed heterogeneous software on different platforms and sharing of data resources via the Internet (Alshraideh & Katuk, 2019; Arsana, 2020; Baharuddin et al., 2022). Web service implementation provides convenience in software development without relying on databases, platforms, or programming languages (Choirudin & Adil, 2019; Nugroho & Primadewi, 2021). There are three web service methods that can be used in exchanging data and information and have been standardized by the web consortium, namely Extensible Markup Language Remote Procedure Call (XML-RPC), Simple Object Access Protocol (SOAP), and Representational State Transfer (REST) (Nugroho & Primadewi, 2021; Yogiswara et al., 2014).

In this research, the data integration process used the Representational State Transfer (REST) web service method with the JSON (Java Script Object Notation) data format and was built using the Slim micro framework. Slim micro framework is a PHP micro framework using the MVC (Model View Controller) model, which can help code web applications and APIs (Application Programming Interface) quickly and easily (Lockhart et al., 2020; Sunardi & Suharjito, 2019). The Slim framework focuses on the basic needs of a web, namely receiving an HTTP request, sending it to the appropriate code snippet, and returning an HTTP response (Khoirunnisa & Kurniawan, 2019; Lockhart et al., 2020). RESTful web services technology uses HTTP verbs as a basis for managing data. HTTP verb methods used in REST include Get, Post, Put, Delete, and so on (Herdiyatomoko, 2022). The HTTP verbs method is used to perform CRUD operations on the database, namely Create, Read, Update, and Delete. REST does not require parsing XML (eXtensible Markup Language) and does not require message headers to communicate with service providers (Jatrain et al., 2021). This is why REST web service architecture is more widely used because it uses less bandwidth (Pautasso, 2009; M. G. L. Putra & Putera, 2019). The JSON data format is used because it is lighter in data exchange, easy to translate and produce by computers, and easy to read and write (Droettboom, 2020; Rahmatulloh et al., 2019). JSON is a text format that does not depend on any programming language because it uses the language styles commonly used by C family programmers, including C, C++, C#, Java, JavaScript, Perl, Python, and others (www.json.org, n.d.). JSON is ideal for use as a data exchange language because it has a language style that is commonly used by programmers. JSON has two structural models. The first model is referred to as an object. One object contains a collection of name/value pairs (key-value pairs) that are not sorted (non-sorted) and start with a "{" sign and end with a "}" sign. Each name is followed by a ":" and name/value pairs are separated by a ",". The second structure is Array. Array is a collection of a number of consecutive values. Array starts with "[" and ends with "]", every two array elements are separated by "," (www.json.org, n.d.; Zhou, 2022). The term "array" and the term "object" originate from JavaScript (Wickham, 2018).

Many studies related to RESTful Web Service for data integration have been carried out, including one by Rizal and Rahmatulloh entitled "RESTful Web Service for the Integration of Academic and Library Systems at Perjuangan University" (Rahmatulloh et al., 2019), which successfully implemented REST Web Service using the JSON data format to integrate academic and library information systems. This research is based on the existence of various applications in each work unit of the Perjuangan University campus. This causes the process of inputting loan data to be repeated because students who have been recorded as returning loan books to the library information system are not directly recorded as returning loan books to the academic system. Furthermore, research was conducted by Nugroho and Primadewi with the title "Application of Web Services for Simperpus and SIAK Data Integration" (Nugroho & Primadewi, 2021). This research resulted in an integrated system between library information systems and academic information systems. The purpose of data integration between the two information systems is because each has a database, causing duplication and data

inconsistency. The next previous research conducted by Fitriani et al (Fitriani et al., 2022). was entitled Implementation of Restful API in Efforts to Synchronize Data in Web-Based Library Automation Systems Using RMS and White Box Trial Methods. The purpose of this research is to overcome data redundancy problems that occur in the STTPLN library automation system, namely the Senayan and Cendana systems. This research has successfully implemented a web service for migrating and merging data from the Senayan and Cendana databases to the Akasia database. REST web services are used to test handling migration and data aggregation with large amounts of data.

This research can contribute in helping the integration of data between SAT-IKA and the Library Information System, so that data consistency is maintained, reducing data redundancy, data duplication because it is stored in different locations, and the same master data can be used together.

2. RESEARCH METHOD

This research uses Data Collection And Analysis Techniques Method. The data in this research were collected using several methods, including literature review and Observation. A literature review was conducted to collect and find literature and library sources that support this research. The literature review is in the form of books, articles, and research journals related to RESTful web services with the JSON data format. The collected literature and library can support and provide adequate information for completing this research. Observation, Observations are carried out by making direct observations in the field in order to obtain the condition of the object under study. Through observation, the writer gains a better understanding of the subject and object being studied. Observations were made on the database of each system under study. This observation was made to find out the platform and table structure of each system so that data integration can be carried out.

Software Development Life Cycle (SDLC) method SDLC is a sequential framework used to identify all activities needed in software engineering, where the completion of one set of activities causes the start of the next activity (Hanani & Supriyono, 2019; Iqbal & Arifianto, 2019). In general, SDLC (figure 1) is divided into several stages (Y. W. S. Putra & Adhim, 2022; Wahyudi, 2018): (1) Planning: this stage aims to identify and prioritize what information systems will be developed; (2) Analysis: this stage aims to determine system or software requirement specifications; (3) Design: at this stage, the design of the interface, database relations, and general description of the system are carried out; (4) Implementation is the coding stage using a certain programming language. (5) Testing: testing the system that has been built and ensuring the system is free from errors; (6) Maintenance: Maintenance can help the development process, starting from specification analysis to changes to existing software, but not to create new software and carry out periodic maintenance or repairs.

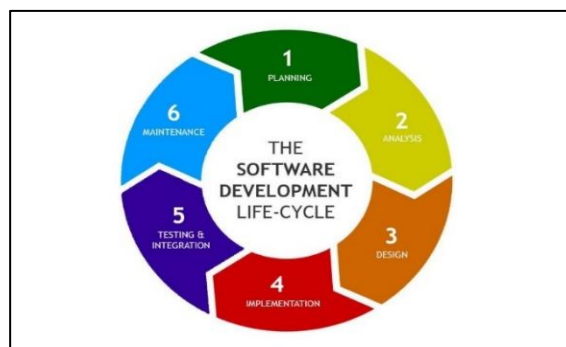


Figure 1. Software Development Life Cycle (SDLC) Diagram (S, 2017)

3. RESULTS AND DISCUSSIONS

3.1 Designing Use Case Diagrams

The web service design in this research uses the Use Case Diagram. A use case diagram is used to briefly describe the scope of the system, such as who is using it and what things it can do. The RESTful web service, using the Use Case Diagram, is tailored with the Slim Micro Framework, which uses the MVC concept. The use-case diagram will explain the relationship between web services and actors. This relationship can be a communication request from an actor to a web service or a response from the web service to an actor.

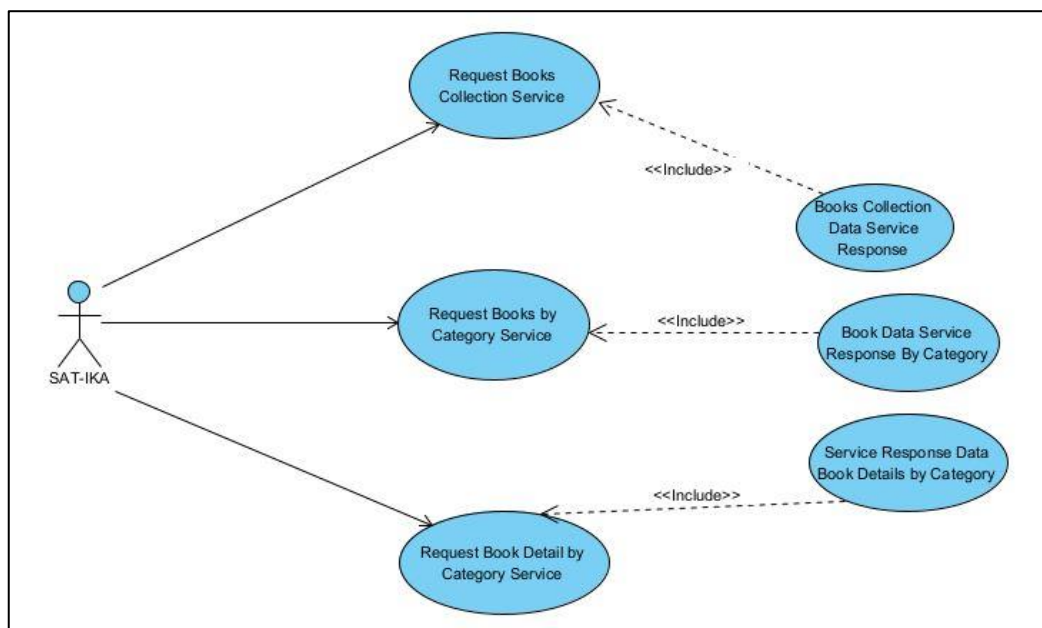


Figure 2. Use Case Diagram

The RESTful web service use case design is shown in Figure 2. In this design, there is one actor, namely the SAT-IKA information system. The functions contained in the designed RESTful web service are "Request Books Collection service, Request Books By Category Service, dan Request Book Detail by Category Service". The requested data (request) will be responded to with JSON data. The REST web service design structure that will be created will integrate existing applications so that they communicate with each other. The structure of the RESTful web service can be seen in Figure 3. The rest client will make a request using an HTTP request to a RESTful web service, and then the request will be identified (PUT, GET, POST, and DELETE). After the request is recognized, it then performs operations on the database according to the action of the method that has been identified. Furthermore, after performing operations on the database, RESTful web services will return results in the form of JSON data to the client.

3.2 System Overview

An overview of the system is shown in Figure 3. The built-in web services can be accessed using SAT-IKA via a desktop or laptop web browser. When a user requests data via HTTP Request, the request will be identified first. Within a REST web service, requests are identified by the Rest Request Handler, whether they are PUT, GET, POST, or DELETE commands. Furthermore, after being identified in the Rest Request Handler section, the request will be forwarded to the Services Handler. In the Services Handler

section, processing will be carried out in accordance with the identification results on the Rest Request Handler. After that, it will enter the Database Handler to perform database processing (create, read, update, or delete). After processing the database, a response will be given via the Services Handler in the form of JSON data. Query results on the database will be wrapped in the form of JSON data. This JSON data will then be sent from the Service Handler to the Rest Request Handler. In the end, the response reaches the client in the form of JSON data. The JSON data will then be read by the client (SAT-IKA). On the client side, the JSON data response will be received and displayed using PHP and HTML language assistance.

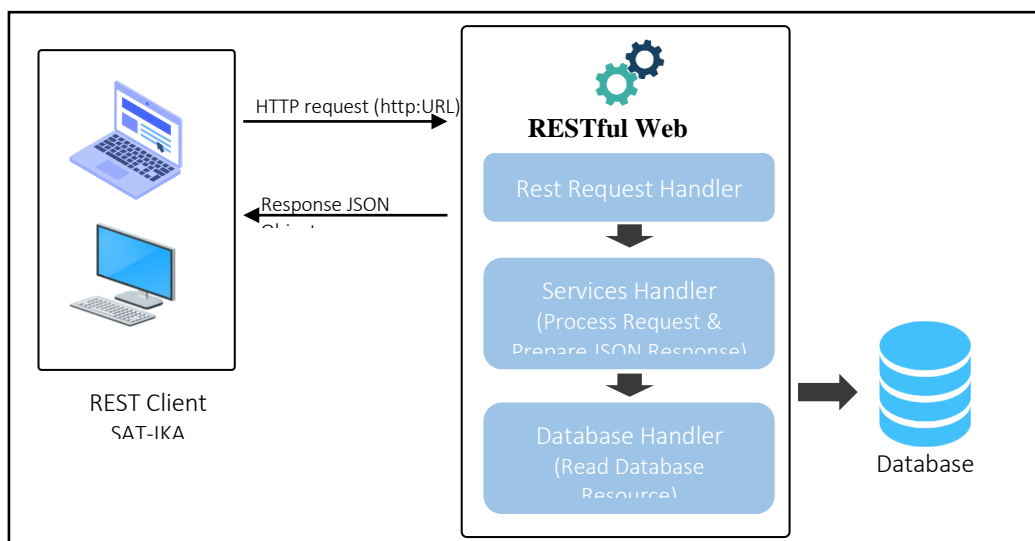


Figure 3. System Overview

The following is a code snippet for the “layanan buku by kategori” RESTful web services:

```

$app->get("/biblios/search_by_kat", function (Request $request, Response $response)
{
    $sql="SELECT mst_gmd.gmd_id, mst_gmd.gmd_name,COUNT(mst_gmd.gmd_id) AS
    'Jumlah_Koleksi' FROM biblio RIGHT JOIN mst_gmd ON(biblio.gmd_id=mst_gmd.gmd_id)
    GROUP BY mst_gmd.gmd_name";
    $stmt = $this->db->prepare($sql);
    $stmt->execute();
    $result = $stmt->fetchAll();
    return $response->withJson(["status" => "success", "data" => $result], 200);
});
  
```

This method is used to retrieve or read resources in the database. Requests are recognized or identified from the code snippet `$app->get`, and the keyword "get" is interpreted as a command to fetch or read resources in the database. The code snippet after get is `/biblios/search_by_kat,` which is a URI (Uniform Resource Identifier) identifier. After being recognized as a resource-reading command, the query will be executed. After the process of fetching or reading the resource, the data will be sent in the form of JSON. The response keyword is shown in the code snippet `"return $response->withJson(["status" => "success", "data" => $result], 200);"`. On the client side, it will display the JSON data that has been received. The following is an example of the results of a RESTful web services response for book services by category (Figure 4).

3.3 Implementation RESTful web services

The application of RESTful web services in this research maintains the concept of data integration. Where the database still refers to the library database, which aims to avoid data redundancy and resources that can be used together. The development of RESTful web services with the help of the Slim Micro Framework greatly facilitates program coding and does not require large storage space. RESTful web services that have been built are tested using the POSTMAN application; the test results are shown in Figure 4.

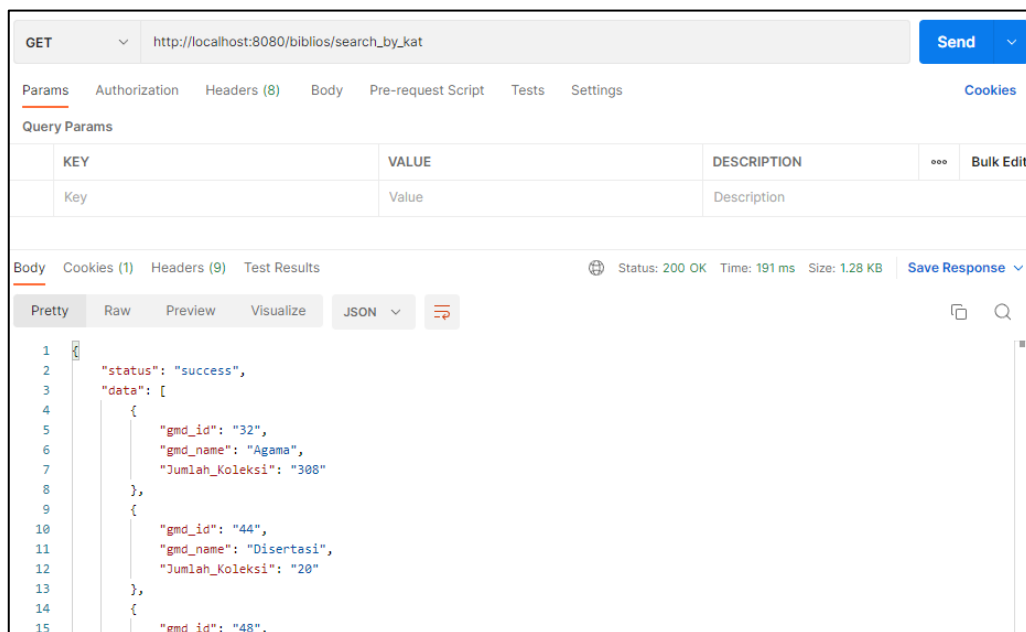


Figure 4. Response “layanan buku by katagori” RESTful web services

Based on Figure 4, the results of testing RESTful Web Services using the POSTMAN application can be explained by seeing that, from the client side, it sends a URI (`http://localhost:8080/biblios/search_by_kat`) using the GET method. The submitted URI will be identified by RESTful Web Services and redirected to the appropriate function. From Figure 4, it can also be seen that the JSON Data response provided by RESTful Web Services is successful and accompanied by data query results to the database.

4. CONCLUSION

This research has successfully implemented a RESTful (Representational State Transfer) Web Service using a slim micro framework to support data integration between the Library Information System and the SAT-IKA Information System (Integrated Database Information System IAHN-TP Palangka Raya). The application of a RESTful web service contributes to supporting consistency so that it is maintained, avoids data redundancy on the IAHN-TP Palangka Raya campus, and allows the use of the same database for several applications. The limitation of this research lies in authentication security. This allows anyone to access the data. Suggestions for future research, namely in the development of the next RESTful web service, is to have an authentication security feature by implementing certain algorithms. The authentication feature will ensure that the web service communicates with clients who have access rights. Security features are considered necessary so that data privacy is maintained.

REFERENCES

- Alshraiedeh, F., & Katuk, N. (2019). SOAP and RESTful web service anti-patterns: A scoping review. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(5), 1831–1840. <https://doi.org/10.30534/ijatcse/2019/05852019>
- Arsana, I. N. A. (2020). Implementasi Web Service Pada Integrasi Data Kerja Praktik, Seminar Dan Tugas Akhir. ... “*Seminar Nasional Riset Teknologi Terapan.*” <https://jurnal.untidar.ac.id/index.php/senaster/article/view/2649>
- Baharuddin, Wakkang, H., & Irianto, B. (2022). Implementasi Web Service Dengan Metode Rest Api Untuk Integrasi Data Covid 19 Di Sulawesi Selatan. *Jurnal Sintaks Logika*, 2(1), 12–22. <https://jurnal.umpar.ac.id/index.php/sylogDOI:https://doi.org/10.31850/jsilog.v2i1>
- Choirudin, R., & Adil, A. (2019). Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 18(2), 284–293. <https://doi.org/10.30812/matrik.v18i2.407>
- Droettboom, M. (2020). *Understanding JSON Schema*. 50. <https://spacetelescope.github.io/understanding-json-schema/UnderstandingJSONSchema.pdf>
- Fitriani, Y., Sudirman, M. Y. D., Kusuma, D. T., & Imantara, A. W. (2022). Implementasi Restful API dalam Upaya Mensinkronisasi Data pada Sistem Otomasi Perpustakaan Berbasis Web Dengan Menggunakan Metode Uji Coba RMSE Dan White Box. *PETIR*, 15(1), 145–156. <https://doi.org/10.33322/petir.v15i1.1228>
- Hanani, M. W., & Supriyono, H. (2019). Pembuatan Augmented Reality Sebagai Media Informasi Peninggalan Sejarah Pada Museum Keris. *Eprints Universitas Muhammadiyah Surakarta*.
- Herdiyatomoko, H. F. (2022). BACK-END SYSTEM DESIGN BASED ON REST API. *Jurnal Teknik Informasi Dan Komputer (Tekinkom)*, 5(1), 123. <https://doi.org/10.37600/tekinkom.v5i1.401>
- Iqbal, M., & Arifianto, D. (2019). Analisa Augmented Reality Dalam Aplikasi Pengenalan Monumen Menggunakan Unity-3D. *Journal of Chemical Information and Modeling*, 53(9).
- Jatain, A., Bajaj, S. B., Chaudhary, S., & Batra, P. (2021). Rest Web Services: An Elementary Learning. *Research Journal of Engineering and Technology*, 75–78. <https://doi.org/10.52711/2321-581x.2021.00012>
- Khoirunnisa, L., & Kurniawan, F. (2019). Sistem Informasi Geografis Pemetaan Komoditas Pertanian dan Informasi Iklim Berbasis Slim Framework. *Sains, Aplikasi, Komputasi Dan Teknologi Informasi*, 1(1), 16. <https://doi.org/10.30872/jsakti.v1i1.2260>
- Lenzerini, M. (2002). Data integration: A theoretical perspective. *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 233–246.
- Lockhart, J., Smith, A., & Allen, R. (2020). Slim Framework. In www.slimframework.com. <https://www.slimframework.com/docs/v4/>
- Nugroho, S., & Primadewi, A. (2021). Penerapan Web Service untuk Integrasi Data Simperpus dan SIAK. *Jurnal Komtika (Komputasi Dan Informatika)*, 4(2), 71–81.
- Pautasso, C. (2009). RESTful Web service composition with BPEL for REST. *Data and Knowledge Engineering*, 68(9). <https://doi.org/10.1016/j.datak.2009.02.016>
- Putra, M. G. L., & Putera, M. I. A. (2019). Analisis Perbandingan Metode Soap Dan Rest Yang Digunakan Pada Framework Flask Untuk Membangun Web Service. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, 14(2), 1–7. <https://doi.org/10.33005/scan.v14i2.1480>
- Putra, Y. W. S., & Adhim, M. F. (2022). Sistem Informasi Presensi Online Menggunakan Teknologi Face Recognition dan GPS. *Jurnal Tekno Kompak*, 16(1). <https://doi.org/10.33365/jtk.v16i1.1470>
- Rahmatulloh, A., Muhammad, I., & Nuralam, F. (2019). *JSON Implementation with Zlib Compression for Database Efficiency in Handling Dynamic Upload Multiple Images*. 7(3), 166–173.
- S, S. (2017). A Study of Software Development Life Cycle Process Models. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2988291>
- Sunardi, A., & Suharjito. (2019). MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based. *Procedia Computer Science*, 157, 134–141. <https://doi.org/10.1016/j.procs.2019.08.150>
- Valle, L. D. (2012). Data Integration. *IT - Information Technology*, 54(3), 103–104. <https://doi.org/10.1524/itit.2012.9075>
- Wahyudi, A. (2018). Perancangan Sistem Menggunakan Metode Sdlc. *Jurnal Dinamika Informatika*, 4(2).

- Wickham, M. (2018). Practical Android. In *Practical Android*. <https://doi.org/10.1007/978-1-4842-3333-7>
- [www.json.org.](https://www.json.org/) (n.d.). *Introducing JSON*. <https://www.json.org/json-en.html>
- Yogiswara, Y., Wijono, W., & Dachlan, H. (2014). Kinerja Web Service Pada Proses Integrasi Data. *Jurnal EECCIS*, 8(1).
- Zhou, M. (2022). A Review of JavaScript Object Notation in Data Analysis. *Proceedings of the 2022 8th International Conference on Humanities and Social Science Research (ICHSSR 2022)*, 664(Ichssr), 1771–1774. <https://doi.org/10.2991/assehr.k.220504.321>