Improving REST API Performances Using Design Patterns: A Case in Healthcare Business Intelligence Systems

Devita Dwitama Putri Baron
Department of Informatics
Universitas Jenderal Achmad Yani
West Java, Cimahi, Indonesia
devitadwitama18@if.unjani.ac.id

Faiza Renaldi and Irma Santikarama
Department of Information Systems
Universitas Jenderal Achmad Yani
West Java, Cimahi, Indonesia
faiza.renaldi@unjani.ac.id, irma.santikarama@lecture.unjani.ac.id

Abstract

Managing a lot of data using a well-designed health information system can prevent medical errors and support doctors and medical service providers in providing the proper diagnosis to patients. This condition is inseparable from advances in data exchange technology commonly used, namely REST and SOAP. The REST API is beneficial in exchanging medical record data faster in real-time. Problems then arise as time goes by, where transactions significantly increase, creating a longer access time. This study solves the problem by adding design pattern technology using a caching pattern. The data retrieval process on the web service is stored in the cache so that when retrieving data, there is no need to re-query the database and then update the cache on the web service and update daily using Cron. This research uses a case study of a healthcare business intelligence system with various data taken from 3 different systems that have been integrated. Tests were carried out in on-premise and cloud server environments with variations between 5,000 and 10,000 data and tested 50 times each. After using the catching pattern, the measurement results of on-premise web services show a reduction in response time of up to 87%. For further development, an additional service is needed to manage caches that are simultaneously entered into the server so that RAM usage occurs optimally and does not cause data access time delays.

Keywords
Web Service, REST, Design Pattern, Business Intelligence, Cache.

1. Introduction
The health industry is currently one aspect of life that continues to change and develop following technological advances that continue to grow until now (Hamson et al. 2021)(Erdal 2018). Digital health services provide convenience and benefits to increase their efficiency (Almotawkel et al. 2021) in the area of managing patient medical record data, hospital records, and health examination results (Dash et al. 2019)(Andreu-Perez et al. 2015). In a study conducted by Leila Ismail et al. in 2020 on eHealth, the research results show that managing many data using a well-designed health information system can prevent medical errors. Supporting medical service providers in providing proper diagnoses to patients and improving the quality of data access in real-time is the main thing in the health data management system. It can reduce medical incidents due to delays in data updates by doctors (Ismail et al. 2020). The study found that health data was evolved using a client-server-based data management system that allows doctors to access and update patient medical records in real-time.

Data exchange technology in the healthcare industry uses web services for communication and exchange of information (Islam et al. 2018)(Ivanova 2014), such as REST and SOAP (Soni and Ranga 2019). There are also several studies and many organizations that use REST API as a web service architecture used to accelerate data retrieval and recording, such as in the field of Restaurant Business (Ahmad et al. 2021)(Hidayat et al. 2020), School Learning (Jamuna et al. 2009)(Kautsar et al. 2016), Online Shopping (Adryamarthanino et al. 2019)(Islamiati et al. 2019), and Health(Khan and Karim 2020)(Guntupally et al. 2019). Especially in health, REST API is beneficial in exchanging essential documents, and medical record data becomes faster in real-time (Pai et al. 2021)(Boutas et al. 2020). REST architecture is easy to use, fast, lightweight and supports all types of data, requiring less bandwidth (bandwidth, response time, and memory requirements). Thus, it concludes that it is an excellent web service architecture in the current era (Toman 2020). Over time health data is not only used in systems or applications in the medical field, such as hospitals or health clinics but also used by organizations and companies as indicators of company health through the integration of data used in business intelligence technology, so that REST API technology continues to develop and significantly increase the demand for users to innovate constantly, especially in the process of data integration in business intelligence. Business intelligence uses a lot of data on employees, health, salary, and employee health care costs.

Business Intelligence is included in one of the technologies in the field of information technology that continues to be updated every year (Talaoui and Kohtamäki 2020). BI is used as a combination of processes and technologies that turn raw data into information built to help companies in decision-making (Obeidat et al. 2015). The use of a lot of data in the business intelligence process results in longer data exchange time. It affects how an organization or company can make fast decision-making. Preparing prompt information is needed to support success in decision making (Rouhani et al. 2016)(Azeroual and Theel 2018)(Tuncay and Belgin 2010). There is a need for technology that can be used to speed up the process.

Improving the performance of data exchange rates in web services in various architectures has been done before using design patterns. For example, Li Li et al. use several important design patterns to facilitate hypertext-based navigation for rest APIs and their implementation to clients (Li et al. 2016). Other research was conducted on the REST API design pattern for the SDN Northbound API by addressing many critical issues in the design of restful network protocols and presenting a framework on how network protocols can be designed genuinely RESTful way(Zhou et al. 2014). Real-world contribution of design patterns was reported in the design of E-Business applications carried out in building search web services efficiently with research results that show success in improving response time and throughput faster (Elshater et al. 2015).

Although some research has been done a lot on improving the speed of data exchange with design patterns in web services. Still, no one has tried to implement these performance acceleration innovations in data integration with REST APIs using design patterns in business intelligence. We find this an important issue to ensure that business intelligence used by companies can provide information faster so as not to cause delays in decision-making due to the length of time business intelligence is used. Companies or business organizations need to increase the speed of data exchange performance in business intelligence in today's digital era because the need for business intelligence increases every time. Several studies have discussed the impact of business intelligence systems on the quality of strategic decision-making (Schultheis 2016)(Ruzgas and Bagdonavičienė 2017), which further strengthens the position of business intelligence as a technology-based tool that is important to continue to be developed for a company or organization.

In this study, we built a web service with a rest architecture using design patterns to increase the speed of data exchange between databases and business intelligence. The design pattern used in this web service uses caching patterns, where every data retrieval process on the web service is stored in the cache. There is no need to re-query to the database when retrieving data again. Then the cache update on the web service will constantly be updated every day using cron. In this case, the data exchanged is data on the health, salary, attendance, and treatment records of employees in a company. All the data is obtained in the health system, staffing system, and integrated corporate financial system.

2. Methods

Several stages are carried out in this study, as seen in Figure 1. The first stage is the identification and collection of data. This stage is carried out to collect data needed for research. This process is an essential first step when working on research in the field related to web services(Neumann et al. 2021). The research methods in this study are shown in Figure 1.

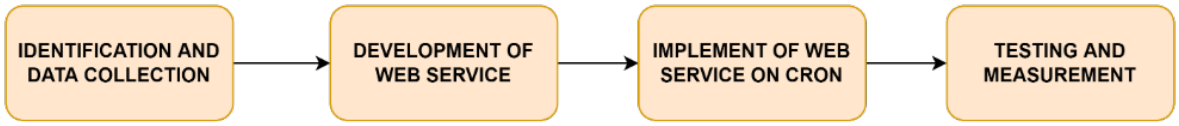

Figure 1. Research method

The first stage is critical in this research because the development of the web service must be data retrieval or delivery of services to data collected in a database (Overeem et al. 2021)(Khan et al. 2020). After identifying and collecting the data obtained, the next stage is continued by building a web service. We use a data retrieval service for business intelligence systems with built-in queries stored in web services. Next is the implementation of cron jobs on the web service that has been created, followed by testing and measurement. This action needs to be done at the final stage to provide the final results of software performance and conformity as an assessment of the quality of the product (Dadkhah et al. 2020). This stage is also the conclusion of the research results that have been done from each stage.

2.1 Identification and Data Collection

The data collection stage is carried out to collect all data used in research and identification by observations made before software development on all data used to determine the structure and attributes used. This study was conducted at Getwell, a healthcare startup company based in Indonesia. The company has a system that regulates all matters related to corporate healthcare management. We found three collections used in the system, namely health complaints, medical expenses, and employees. The data structure of the health complaint collection in this study is shown in Figure 2.

Figure 2. The data structure of the health complaint collection

Here is a HealthComplaint data with 11 variables starting from complaints, location of complaints, up until the duration of complaints. Attributes of id_company, id_user, NIK, sickLeave, status, createAt, and updatedAt are used as indicators. Employee health is seen from the track record of complaints given. The data structure of employee and medical expenses collection in this study is shown in Figure 3.

Figure 3. Data structure from employee and medical expenses collections

The company's employee data used in this study is in the form of employee data with attributes _id, id_company, code_company, nameCompany, NIK, NIP, name, DayOfBirth, birth_Place, gender, phone_number, email, branch, department, departmentID, joinDate, salary, timeWorkYears, status, statusMcu, created_at, and updated_at used as employee identity data. Furthermore, this study also used attendance data attributes with detailed company data and finances related to company employee health data. The latest data used in this study is medical expenses data with attributes _id, nama_karyawan, departments, id_department, branches, jenis_pelayanan, lokasi_pemeriksaan, tanggal_perawatan, tanggal_pembayar, total_biaya_perawatan, pembayaran_oleh_perusahaan, company_code, company_id, diagnoses, id_employee, icd_diagnosa, isDeleted, createAt, and updateAt used as financial records. Issued by the company for the health needs of working employees.

2.2 Development of Web Service

Web service is a technology that is constantly developing. In its construction, web service has several architectures that are quite popularly used, namely SOAP (Simple Object Access Protocol) and REST (REpresentational State Transfer) (Halili and Ramadani 2018). Both architectures have their shortcomings and advantages. SOAP has data output in XML format, while REST with data output format is JSON. With all the considerations in it, when viewed in terms of reliability, the REST architecture in the data delivery process is often chosen to develop web services (Ranga and Soni 2019)(Giessler et al. 2015). In our research, web services were created as a medium of data exchange between servers and clients with the implementation of pattern catching design as an implementation to improve the performance of data exchange rates in web services created. This web service will also be done to create business intelligence queries. Business intelligence is defined as knowledge gained through technology that allows organizations to turn data into useful information (Somya and Manongga 2018).

2.3 Implement Web Service on Cron Job

The next stage is the implementation of web service against cron job. This action is done to update the cache to data updates in the database periodically. Previous studies provide results in a lack of use of caching patterns related to expired data (not yet updated). Hence, using cron jobs in web services can help solve these problems. The process of web service implementation on the cron job can be seen in figure 4.

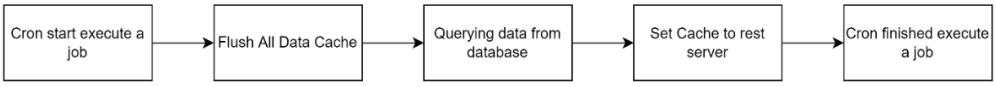

Figure 4. Web service implementation process on a cron job

The cron needs to set up the update schedule for the first implementation. In this study, the cron job was set to constantly schedule a time of 1x24 hours to update data every 00.00 AM. This condition is done because the server and database have low traffic or transaction. After the setup, the cron will start updating the data according to the

schedule that has been given. In the next stage, the cache is deleted on the REST server. Re-query the database to get the latest data, then update the query data in the set or stored in the REST server, and the data updating processes on the cron are declared complete.

2.4 Testing and Measurement

At this stage of testing and measurement, several test cases will be created that will be tested for data integration testing and measure the execution time in running the query made. The amount of data exchange process time needed.

2.4.1 Execution Time Measurement

This measurement is done to see how long it takes to take a lot of data and execute the process until it can be displayed in the business intelligence dashboard. This measurement is carried out gradually with the amount of data that will be tried to continue to grow in each test session. Here is the test case table in this measurement shown in Table 1.

Table 1. Table test case in rest API request measurement

Test Case Code	Total Data	Number of Requests
EXT01	5000	50
EXT02	6000	50
EXT03	7000	50
EXT04	8000	50
EXT05	9000	50
EXT06	10000	50

Testing starts from code EXT01 to EXT06 using 5000 to 10000 with a data retrieval request of 50 times for each amount of data tested.

3. Results and Discussion

We continued our research and produced a single web service with a REST architecture using caching patterns as improved service performance speed, implementation, and setting of cron jobs, testing, and measurement. All components in this study can be seen in the architecture shown in Figure 5.

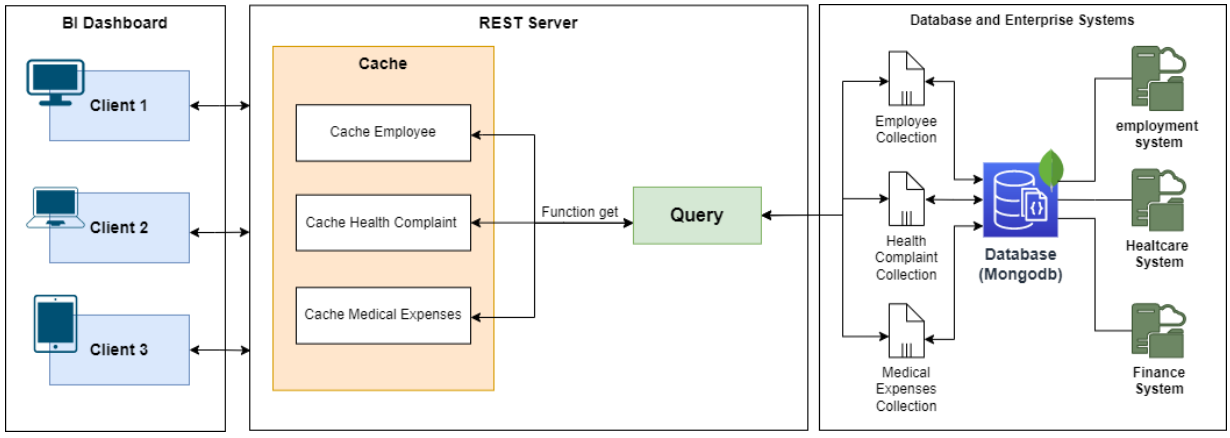

Figure 5. Rest API architecture using pattern caching

As seen in the image above, the request and response process will occur between clients who use BI Dashboard with REST Server or a web service created. The first request process that the client has done will be automatically stored in the cache. The cache building in this study is made by the 3 data used, namely Cache Health Complain, Cache Medical Expenses, and Employee Cache. So that when at another time the client makes a request back to the same data does not need to take back into the database but directly retrieved from the available cache.

3.1 Web Service

This web service development uses the Fastify framework developed in the cloud with its domain name. The results of data retrieval using postman can be seen in Figure 6.

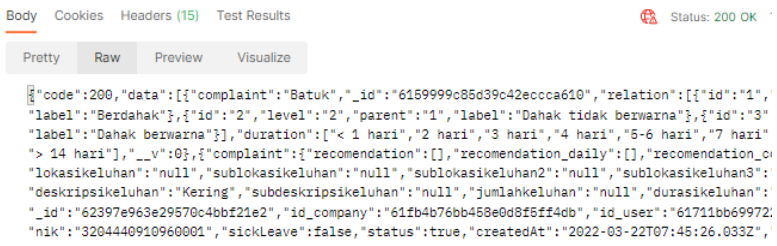

Figure 6. Data retrieval results in the collection of health complaint

The data search results with the response code 200 show that the data is filled with the id "6159999c85d39c42eccca610" and the complaint details from the employee. This result shows that using a web service can succeed in data retrieval.

3. 2 Web Service on Cron Job

Web services are then implemented on cron jobs to synchronize data in the cache to be updated constantly with data changes in the database. After the web service is uploaded to the online host service provider, the domain address can be accessed using Cron with a predetermined scheduling time. The implementation of the cron job in web service is shown in Figure 7.

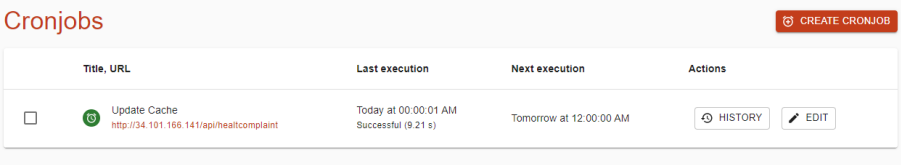

Figure 7. Cron job execution results in rest API

The schedule has been determined repeatedly at midnight. After an experiment using the appropriate domain address, the renewal was successfully carried out with a processing speed of 9.21 s.

3. 3 Testing and Measurement Results

It is always done using an internet connection at this stage because the entire web service and the implementation of the cron job are only presented in a cloud environment. The web service in this study was on a cloud service with specifications of 2 processor cores and 4 GB of ram. The client who makes the request has an internet download speed of 20Mb/s and an upload speed of 5 Mb/s. The data used in the study was taken from the MongoDB database.

3.3.1 Execution Time Measurement Result

Measurements are made using the postman application. Postman is an API platform for developers to design, build, and test APIs. The measurement results on the rest API that have and have not used caching patterns can be seen in figure 8.

Figure 8. Rest API measurement results in different amounts of data

The measurement results showed a significant decrease in access time. The access time is reduced to (±) 1029 ms for each different amount of data. At the time of measurement, the first iteration is a request on the rest server with the response data coming from the query to the database, and it causes the high time needed at the time of data access. Measurements in the second iteration and so on are requests to rest servers whose data response comes from the cache. The second iteration on each amount of data displays access time faster than the first iteration. This significant time reduction occurs in iterations 2-8. Depending on how much data it uses, such as measurements with a data count of 10000, a significant decrease in access time is only found in iterations 1-4. After several iterations of requests made on the cache on the rest server, the access time on the cache increases, and the access time increases to (±) 50% of the rest server access time before storing the data in the cache. The access time is stable until the last iteration. A comparison of the access speed of the rest APIs that have not used caching patterns can be seen in figure 9.

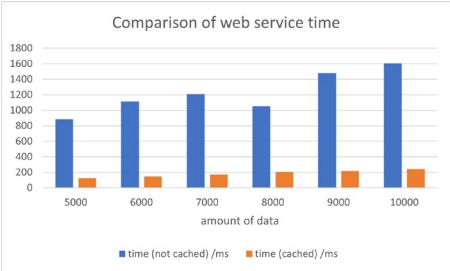

Figure 9. Comparison of rest API measurements before and after using caching patterns

The results of this measurement also found another fact, every time the amount of data is requested, the increase in access speed continues to decrease. Requests using a data count of 6000 experienced an increase in access speed by 87%, unlike requests with the amount of data of 10000 increased by 85%. The increased access speed can be seen in figure 10.

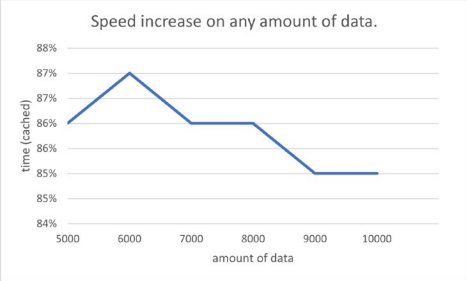

Figure 10. Rest-API access speed improvement (after using a caching pattern)

This situation appeared due to bottlenecks when downloading data to be retrieved. The downloading of requested data is not reduced much when it has or has not used caching patterns. The download time at the time of request data can be seen in figure 11.

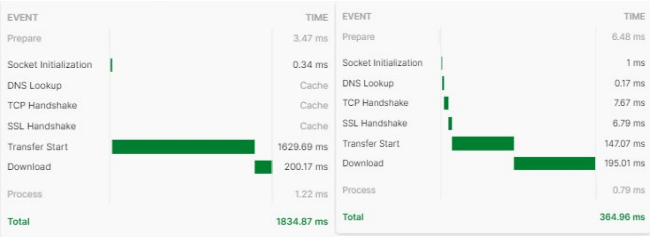

Figure 11. Description of rest API access time before and after using caching pattern

The access time before using the caching pattern requires a transfer request time of 1629.69 ms, and the time used to download data is 200.17 ms. After using the caching pattern, the transfer request time became 147.07 ms, and the time used to download data did not change significantly, around 195.01 ms. From this, it can be seen that the use of caching patterns affects the request and response time, not the download time, and internet speed will affect the time of request data from the rest API.

4. Conclusion

This research aims to increase the speed of web service access by using design patterns in the health intelligence business. We have presented a caching design pattern approach on the rest server. The data used is data related to health, such as health complaint data, medical expenses, and employees. Any query data from the database is entered in the cache on the rest server. The database in this study is stored on the MongoDB database system. Caching patterns have weaknesses, such as the expiration of data that has to be updated. This study performed data updates on the cache using cron jobs that will be triggered automatically every time specified. Rest API are stored on cloud services with two processor cores and 4 GB of ram. Measurements were made on the client-side with a download speed of 20Mb / s and an upload speed of 5Mb / s. The results of the rest API measurement made with requests of 50 times and different amounts of data showed an increase in access speed of up to 87%. The speed of access rest API in the cache only increases in the request and response process, not the download speed. The result leads to a decrease in the speed of rest API access as the size of the data increases. This research proves the acceleration of access time after using caching patterns in rest APIs, which is very useful in intelligence business systems that use a lot of data. Suggestions for further development: An additional service is needed to manage the cache that enters simultaneously into the server so that the use of ram does not cause slow data access time lag.

References

Adryamarthanino, A., Persada, A. G., and Hidayah, I., Rest API Development of E-Commerce Auction System Using Spring Boot at PT ABC, *Universitas Gadjah Mada*, pp. 24–25, 2019.

Ahmad, I., Suwarni, E., Borman, R. I., Asmawati, Rossi, F., and Jusman, Y., Implementation of RESTful API Web

Services Architecture in Takeaway Application Development, *2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System, ICE3IS 2021*, pp. 132–137, 2021.

Almotawkel, N. A. A., Shaddad, A. naji, and Qureshi, D. E. A., The Effect Of Information Systems On Healthcare Management And Diagnostic Services, *International Journal of Business and Management Invention (IJBMI)*, vol. 10, no. 7, pp. 1–7, 2021.

Andreu-Perez, J., Poon, C. C. Y., Merrifield, R. D., Wong, S. T. C., and Yang, G. Z., Big Data for Health, *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 4, pp. 1193–1208, 2015.

Azeroual, O., and Theel, H., The Effects of Using Business Intelligence Systems on an Excellence Management and Decision-Making Process by Startup Companies: A Case Study, *International Journal Of Management Science And Business Administration*, vol. 4, no. 3, pp. 30–40, 2018.

Boutas, E., Boulougari, C., Anastasopoulos, V., Veneti, I., and Kousiouris, G., A cloud-based REST platform for real-time health resources availability registering, discovering and matching in pandemic crisis conditions, *Proceedings - IEEE Symposium on Computers and Communications*, vol. 2020-July, , 2020.

Dadkhah, M., Araban, S., and Paydar, S., A systematic literature review on semantic web enabled software testing, *Journal of Systems and Software*, vol. 162, , pp. 110485, 2020.

Dash, S., Shakyawar, S. K., Sharma, M., and Kaushik, S., Big data in healthcare: management, analysis and future prospects, *Journal of Big Data*, vol. 6, no. 1, 2019.

Elshater, Y., Martin, P., and Hassanein, E., Using Design Patterns to Improve Web Service Performance, *Proceedings - 2015 IEEE International Conference on Services Computing, SCC 2015*, pp. 746–749, 2015.

Erdal, E., The Impact of Technology Trends on Healthcare Systems: A Study on Opportunities and Threats, *International Journal of Trend in Scientific Research and Development*, vol. Volume-2, no. Issue-6, pp. 1574–1578, 2018.

Giessler, P., Gebhart, M., Sarancin, D., Steinegger, R., and Abeck, S., Best Practices for the Design of RESTful Web Services, *International Conferences of Software Advances (ICSEA)*, *April 2016*, pp. 392–297, 2015. https://www.researchgate.net/publication/301694429

Guntupally, K., Devarakonda, R., and Kehoe, K., Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example, *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pp. 5328–5329, 2019.

Halili, F., and Ramadani, E., Web Services: A Comparison of Soap and Rest Services, *Modern Applied Science*, vol. 12, no. 3, pp. 175, 2018.

Hamson, Z., Supartha, G., Muntasir, M., Hasan Wahyudi, M., and Sugiyarto, *Informasi Teknologi di Bidang Kesehatan* (Issue May), 2021.

Hidayat, M. M., Dimas Adityo, R., and Siswanto, A., Design of Restaurant Billing System (E Bill Resto) by Applying Synchronization of Data Billing in Branch Companies to Main Companies Based on Rest API, *Proceeding - ICoSTA 2020: 2020 International Conference on Smart Technology and Applications: Empowering Industrial IoT by Implementing Green Technology for Sustainable Development*, pp. 1–5, 2020.

Islam, S., Britto, S., Kumar, R., and Rashid Dar, A., *A Comprehensive Study On Web Services Basics* (Issue July), 2018.

Islamiati, D. S., Agata, D., and Anom Besari, A. R., Design and Implementation of Various Payment System for Product Transaction in Mobile Application, *IES 2019 - International Electronics Symposium: The Role of Techno-Intelligence in Creating an Open Energy System Towards Energy Democracy, Proceedings*, pp. 287–292, 2019.

Ismail, L., Materwala, H., Karduck, A. P., and Adem, A., Requirements of health data management systems for biomedical care and research: Scoping review, *Journal of Medical Internet Research*, vol. 22, no. 7, 2020.

Ivanova, M., Web Services : Features , Technologies and, *PROCEEDINGS of the International Conference InfoTech*, *November*, pp. 156–165, 2014.

Jamuna, R. S., Ashok, M. S., and Palanivel, K., Adaptive content for personalized E-learning using web service and semantic web, *2009 International Conference on Intelligent Agent and Multi-Agent Systems, IAMA 2009*, 2009.

Kautsar, I. A., Musashi, Y., Kubota, S. I., and Sugitani, K., Synchronizing learning material on Moodle and lecture based supportive tool: The REST based approach, *Proceedings of 2015 International Conference on Information and Communication Technology and Systems, ICTS 2015*, *pp. 187-192*, pp. 187–191, 2016.

Khan, M. M., and Karim, R., Development of Smart e-Health System for Covid-19 Pandemic, *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pp. 1–6, 2020.

Khan, S., Qader, M. R., Ka, T., and Abimannan, S., Analysis of Business Intelligence Impact on Organizational Performance, *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*, pp. 16–19, 2020.

Li, L., Chou, W., Zhou, W., and Luo, M., Design Patterns and Extensibility of REST API for Networking Applications, *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 154–167, 2016.

Neumann, A., Laranjeiro, N., and Bernardino, J., An Analysis of Public REST Web Service APIs, *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 957–970, 2021.

Obeidat, M., North, M., Richardson, R., Rattanak, V., and North, S., Business Intelligence Technology, Applications, and Trends., *International Management Review*, vol. 11, no. 2, pp. 47–56, 2015.

Overeem, M., Jansen, S., and Mathijssen, M., API Management Maturity of Low-Code Development Platforms, *Lecture Notes in Business Information Processing*, vol. 421, no. June, pp. 380–394, 2021.

Pai, M. M. M., Ganiga, R., Pai, R. M., and Sinha, R. K., Standard electronic health record (EHR) framework for Indian healthcare system, *Health Services and Outcomes Research Methodology*, vol. 21, no. 3, pp. 339–362, 2021.

Rouhani, S., Ashrafi, A., Zare Ravasan, A., and Afshari, S., The impact model of business intelligence on decision support and organizational benefits, *Journal of Enterprise Information Management*, vol. 29, no. 1, pp. 19–50, 2016.

Ruzgas, T., and Bagdonavičienė, J. D.-, Business Intelligence for Big Data Analytics, *International Journal of Computer Applications Technology and Research*, vol. 6, no. 1, pp. 001–008, 2017.

Schultheis, M. P., *The impact of Business Intelligence systems on the perceived quality of strategic decision making*, *March*, 2016.

Somya, R., and Manongga, D., Service-oriented Business Intelligence (SoBI) for Academic and Financial Data Integration in University, *2018 International Seminar on Application for Technology of Information and Communication*, pp. 1–5, 2018.

Soni, A., and Ranga, V., API features individualizing of web services: REST and SOAP, *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9 Special Issue, pp. 664–671, 2019.

Talaoui, Y., and Kohtamäki, M., 35 years of research on business intelligence process: a synthesis of a fragmented literature, In *Management Research Review* (Vol. 44, Issue 5, pp. 677–717), 2020.

Toman, S. H., *Review of Web Service Technologies : REST over SOAP*, vol. 12, no. 4, pp. 18–30, 2020.

Tuncay, E. G., and Belgin, O., Effects of Business Intelligence Techniques on Enterprise Productivity, *16th World Productivity Congress*, vol. 2, no. vol. 2, pp. 1–12, 2010.

Zhou, W., Li, L., Luo, M., and Chou, W., REST API design patterns for SDN northbound API, *Proceedings - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, pp. 358–365, 2014.

Biographies

Devita Dwitama Putri Baron is a final year undergraduate student in the department of informatics, Universitas Jenderal Achmad Yani, Indonesia. Her primary interests are systems modeling, optimization, and web service technology.

Faiza Renaldi received his Master of Business Informatics at Universiteit Utrecht, The Netherlands, in 2006 and now serves his time as an assistant professor in the department of informatics, Universitas Jenderal Achmad Yani Indonesia. His research interests are health informatics, information systems/information technology management, e-government, agile project management, and IT entrepreneurship.

Irma Santikarama received a Bachelor's degree in Information systems from Maranatha Christian University and a Master's degree in Informatics from Bandung Institute of Technology. She is a lecturer in the Information Systems Department, Faculty of Science and Informatics, Universitas Jenderal Achmad Yani. Her interest is related to information systems, eGovernment, and agile project management.