Integration at the Application Interface Level between SIAMIK and SIPERPUS (a Case Study in XX)

E D Wahyuni¹, D Novitasari¹, M S Hadi¹ and N Hasanah¹
¹Information Systems Department, Faculty of Computer Science
Universitas Pembangunan Nasional “Veteran” Jawa Timur
Surabaya, Indonesia
E-mail: ¹ekawahyuni.si@upnjatim.ac.id

Abstract. Many business organizations have information systems, but most still operate separately as a silo. The problem that arises is that there is ineffectiveness in the process that is in the application. So the need for an integrated information system that can access two or more systems in the organization is needed. The system integration process is carried out using Enterprise Application Integration (EAI) method at the application interface level. This method is used to integrate academic information systems (SIAMIK) and library information systems (SIPERPUS). The results of this study are that both applications are connected without changing source codes from both applications. The integrated system design can facilitate the admin in the process of registering new users from the SIPERPUS application.

1. Introduction

The development of information technology (IT) has brought a revolution in playing a central role in the business strategy section. Rapid changes and developments within the IT area have shaped new leaders in the market, including in the educational institution sector [1]. A good IS solution allows organizations to get several benefits ie: handle a much greater amount of information, achieve a much higher level of accuracy (minimize error rates), reduce the costs of all existing processes, and improve the services offered to internal and external customers [2]. The rapid development of IT, as well as the growth of higher education institutions in Indonesia, has made competition conditions to seize market share become tighter. Higher education institutions are required to be able to provide maximum services. Those services include services that are supported by sufficing information systems.
Many business organizations now have information systems, it's not uncommon in one organization if they have more than one application or information system. But unfortunately, most of the existing applications, still operating separately, they work independently, not yet integrated. Take for example, in a university, mostly they have SIAMIK applications, which is generally used by students to manage their academic activities, ranging from the entry of courses to be taken in the following semester, to seeing the transcripts of the current semester or the previous semester. These institutions also generally have e-learning applications for their student lectures activity or library applications used for book loan transactions. These three applications actually have the same user, that is students. Ideally, to use these three applications, we only need to use the same username and password. But in reality, each application, both used by students, uses different usernames and passwords (each requires its own registration process). For SIAMIK applications, the username and password used are a, b. For e-learning applications, you need another username and password, as well as the library application.

The conditions that cause this problem are varied, starting from, the applications are not developed simultaneously, or the applications developed by different developers, or it could be when it is at the design stage, the developer did not think how the connection between the applications developed with other applications that are already up and running, or because of the lack of good planning in the development of IT blueprint organizations. Problems that arise due to lack of analysis, trying to be addressed in several studies by implement the TOGAF method [3] [4] or Zachman Framework [5] for developing applications in an organization, to produce integrated applications.

Enterprise application integration has been widely researched, [6] tried to classify various kinds of integration patterns that can be done. Gorkhali [7] conducted a literature review of various application integration techniques from 1992 to 2015 and concluded that great effort was needed to respond to the trend of cloud computing and IoT. There are many methods for integrating applications, like SaaS-based EAI approach [8] or application interface approach, such as by using SOA as did by [9] [10] [11], or by using REST [12] or RESTful [13]. Because SOAP and REST / RESTful technology is often researched and implemented by programmers, this condition encourage [14] to compare the performance between SOAP and REST in order to be able to decide what technology is most appropriate to use when creating a web service.

Application integration can also be done by using ESB, which acts as a bridging for legacy applications, [15] tries to compare the use of ESB and without ESB when creating a service. [16] tried to summarize the best practices of various studies related to the use of ESB (when and how) in the form of a tutorial which will later be used as a foundation in this research methodology. There are also many studies on ESB, such as [17] who tries to integrate ESB with client applications using the RESTful API, while [18] uses ESB to implement SOA and EDA concepts. [19] attempts to analyze the features, security, convenience and performance of various open source ESBs while [20] tested the performance (total and average message counts, overall bytes measure, overall message received and sent, processing time of messages, and memory allocation) of several ESBs used on the market.

From the studies above, this study aims to integrate applications in University XX, between SIAMIK and SIPERPUS applications. In University XX, academic information systems (SIAMIK) and library
information systems (SIPERPUS) are still not integrated with one another. Judging from the structure of the database of SIAMIK and SIPERPUS, those systems actually can be integrated. If the registration process only happens once, which is in the beginning, in the SIAMIK application (during the registration process of new students). Then the same user data can be used in other applications. No more registration process is needed when we want to use SIPERPUS, because the database of SIAMIK and SIPERPUS are already integrated.

2. Methodology

The methodology used in this research is to use the EAI (Enterprise Application Integration) approach at the application interface level to integrate academic information systems (SIAMIK) and Web-based SIPERPUS. EAI or Enterprise Application Integration is an unlimited sharing of data and business processes between applications and interconnected data sources. The reason for using EAI at the application interface level is to meet the organization’s demands for sharing data and business processes without making changes to the application or data structure, the changes made do not interfere with the application source code or database structure. In implementing EAI, it takes someone who truly understands business processes in an organization, because not all organizations have the same system and of course have different problems. [3]

In applying the application interface level method, a developer can combine many applications together and share their business logic and information. This method uses an interface (usually called API) to access processes and data, extract its information and put it in a format that can be understood by the target application and sends the information. [3] The integration of SIAMIK and SIPERPUS conducted in this research uses Mule ESB. Mule, the runtime engine of Anypoint Platform, is a lightweight Java-based enterprise service bus (ESB) and integration platform enables easy integration of existing systems, regardless of the different technologies that the applications use, including JMS, Web Services, JDBC, HTTP, and more. The ESB can be deployed anywhere, can integrate and orchestrate events in real time or in batch, and has universal connectivity. Following are the stages of integration of SIAMIK and SIPERPUS:

![Figure 1. Integration phase](image)

The first stage is the design phase, aiming to design the integration process mechanism that is going to be developed. This step consists of designing the schedule and process. The next stage is the implementation phase which is carried out using Mule ESB. The last stage is testing which aims to check whether the flow that has been created using Mule ESB can run properly and run at the scheduled time so that the SIPERPUS database can be filled with data from SIAMIK. Testing is done by using blackbox testing techniques.

3. Result and Discussion
The results of each stage are as follows

3.1. Design

The design stage consists of two step, i.e schedule design and flow design.

3.1.1. Schedule Design

To determine the data integration period when the application is up and running, we must know the nature of the data. The frequency of data changes from SIAMIK for one year is not very frequent. Changes in data, which are generated from the process of input new student data, do not occur every hour or every day but occur every new academic year, which is in the beginning of the odd semester. Then the integration technique that is felt to be the most appropriate for integrating student data between these two applications is to create a flow that will be run at the beginning of each odd semester (once a year). This flow will be built later using Mule ESB.

3.1.2. Flow Design

Flow sequence that will be built, according to Figure 2, basically is, taking data from the SIAMIK. The data taken comes from several columns from the "mahasiswa" table. The results of this select process will be stored in a variable. The next process is to store this variable to the SIPERPUS database, into “mahasiswa” table (in the corresponding columns).

3.2. Implementation

The implementation of flow design from Figure 2, becomes a flow in the Mule ESB Anypoint. Connectors that used in this flow are http, database, variables, expressions and for each looping.

3.3. Testing
The flow that has been developed were then tested and compared against the scenario prepared, i.e. can function properly (fill the “mahasiswa” table in SIPERPUS database with data from “mahasiswa” table in SIAMIK database). When the flow is run for the first time, it will take student data from the SIAMIK database and store it in the SIPERPUS database. When new student wants to borrow some books, administrators can immediately choose which members that want to borrow their books, even though these members are new students in this odd semester. Flow that has been created at anypoint, has automatically filled the student table in the SIPERPUS database with the latest data from SIAMIK. It can be proven, that the results of the first scenario are success. The next scenario is to check whether the flow can run on a predetermined schedule (early odd semester). The easiest scenario is to change the time setting available on the server where the mule is located. Before the time setting is changed, it is necessary to enter some new data in the SIAMIK.mahasiswa table. then the time setting is added one year from the date the flow was first started. The result, the SIPERPUS.mahasiswa have been filled in with data that was just entered in the SIAMIK.mahasiswa table. After scenario 2 executed, data that previously existed in the SIAMIK.mahasiswa duplicated into SIPERPUS.mahasiswa.

4. Conclusion

Based on the results and discussion above, it can be concluded that the integration of SIAMIK and SIPERPUS aims to connect the two applications without changing source codes in both applications by using mule ESB. With the integration between both systems, it makes administrators manage data in the system easily. It is hoped that by using this technique, both application can run more effectively

5. Reference

[1] Suroto, Nindiani A, Yuliani ENS. ANALISIS SISTEM INFORMASI AKADEMIK DENGAN PENDEKATAN ENTERPRISE RESOURCE PLANNING(STUDI KASUS DI SEBUAH UNIVERSITAS DI INDONESIA). PASTI [Internet]. 2017;XI(1):76–83. Available from: http://publikasi.mercubuana.ac.id/index.php/pasti/article/view/1356/1084

[2] Ardani A, Zaidir. Analisis Dan Perancangan Sistem Informasi Terintegrasi Untuk Manajemen Produksi , ( Studi Kasus : Fabrik Kemasan Kertas Cv . J Teknol Inf [Internet]. 2017;XII(35):1–10. Available from: http://jti.respati.ac.id/index.php/jurnaljti/article/viewFile/171/154

[3] Wahyuningthias ET, Prasetyo YA, Gemilang SFS. Perancangan Enterprise Architecture Menggunakan Framework Togaf Adm Pada Bidang Perpustakaan Bapapsi Kabupaten Bandung. e-Proceeding Eng [Internet]. 2016;3(2):3103–10. Available from: https://libraryproceeding.telkomuniversity.ac.id/index.php/engineering/article/view/2191

[4] Suryadi, Andry JF. Perancangan Enterprise Architecture Menggunakan Toagf Architecture Development Method ( Studi Kasus : Yakuza Gym Jakarta Barat ). In: Seminar Nasional Teknoka [Internet]. 2017. p. 137–50. Available from: http://journal.ubahapka.ac.id/index.php/teknika/article/view/748

[5] Rahayu S, Hadiana A. Perancangan Enterprise Architecture Berbasis Service Menggunakan Zachman Framework: Studi Kasus PDAM Kota Sukabumi. J Teknol Rekayasa. 2017;1(1):59.

[6] Ritter D, May N, Rinderle-Ma S. Patterns for emerging application integration scenarios: A survey. Vol. 67, Information Systems. Elsevier Ltd; 2017. p. 36–57.
[7] Gorkhali A, Xu L Da. Enterprise Application Integration in Industrial Integration: A Literature Review. J Ind Integr Manag. 2016 Dec;01(04):1650014.

[8] Wang BL, Liu H, Song J. SaaS-based enterprise application integration approach and case study. J Supercomput. 2016 Jul 1;72(7):2833–47.

[9] Andy. Penerapan Soa Pada Pengembangan Aplikasi Keanggotaan Kafe Berbasis Mobile Apps Android Di Matador Cafe. TIMES [Internet]. 2017;VI(2):1–14. Available from: http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/article/download/577/137

[10] Mulya MF, Rismawati N. Analisis dan Perancangan Sistem Mediation dengan Protokol Soap pada Web Service untuk Mengintegrasikan Antar Sistem Informasi yang Berbeda Platform. J Ultim InfoSys. 2018 Apr 2;8(2):107–11.

[11] Putrama IM, Dantes GR, Divayana DGH. Implementasi SOA menggunakan Apache ServiceMix dalam rancang bangun Data Implementasi SOA menggunakan Apache ServiceMix dalam rancang bangun Data Warehouse. In: Seminar Nasional Vokasi dan Teknologi (SEMNASVOKTEK). 2016. p. 127–38.

[12] Perkasa MI, Setiawan EB. Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token. J Ultim Comput [Internet]. 2018;10(1):19–26. Available from: http://ejournals.umn.ac.id/index.php/SK/article/download/838/648

[13] Rizal R, Rahmatulloh A. RESTFUL WEB SERVICE UNTUK INTEGRASI SISTEM AKADEMIK DAN PERPUSTAKAAN UNIVERSITAS PERJUANGAN. J Ilm Inform. 2019 Mar 22;7(01):54.

[14] Kumari S, Rath SK. Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration. In: 2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015. Institute of Electrical and Electronics Engineers Inc.; 2015. p. 1656–60.

[15] Bhadoria RS, Chaudhari NS, Tharinda Nishantha Vidanagama VG. Analyzing the role of interfaces in enterprise service bus: A middleware epitome for service-oriented systems. Comput Stand Interfaces. 2018 Jan 1;55:146–55.

[16] Luo M, Goldshlager B, Zhang LJ. Designing and implementing Enterprise Service Bus (ESB) and SOA solutions. In: Proceedings - 2005 IEEE International Conference on Services Computing, SCC 2005. 2005.

[17] Harcuba O, Vrba P. Unified REST API for supporting the semantic integration in the ESB-based architecture. In: Proceedings of the IEEE International Conference on Industrial Technology. Institute of Electrical and Electronics Engineers Inc.; 2015. p. 3000–5.

[18] Maréchaux J. Combining service-oriented architecture and event-driven architecture using an enterprise service bus. IBM Dev Work [Internet]. 2006;(April):1–8. Available from: http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/IBM/I060328M.pdf

[19] Martínez-Carreras MA, García Jimenez FJ, Gómez Skarmeta AF. Building integrated business environments: analysing open-source ESB. Enterp Inf Syst. 2015 May 19;9(4):401–35.

[20] Chaudhari N, Bhadoria RS, Prasad S. Information Handling and Processing Using Enterprise Service Bus in Service-Oriented Architecture System. In: Proceedings - 2016 8th International Conference on Computational Intelligence and Communication Networks, CICN 2016. Institute of Electrical and Electronics Engineers Inc.; 2017. p. 418–21.

Acknowledgments

Authors would like to thank Universitas Pembangunan Nasional “Veteran” Jawa Timur for fully support this project.
