Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Development of an Academic Application Server for Managing Student Data and Grades Based on Web Services

A. Hamdani[1]*, Anak Agung Indah Parwati[2], Shefia Nur Arifin[3]

[1,2,3]*Faculty of Science and Technology, Ibrahimy University, Banyuputih, Situbondo, East Java, Indonesia*
dan.kidz88@gmail.com 1*, chell.ntliaa@gmail.com [2] , shefianurarifin01@gmail.com[3]

Abstract

The development of information technology encourages higher education institutions to digitalize academic systems in order to improve the efficiency and accuracy of managing student data and academic grades. However, in practice, many institutions still rely on manual methods or separate applications, which leads to data synchronization issues, delays in accessing information, and a high potential for input errors. This study aims to design and develop an academic application server based on web services that is capable of managing student data and grades in an integrated, flexible, and real-time manner.The system development method used in this research is Rapid Application Development (RAD), which emphasizes fast development through interactive processes, prototyping, and direct user involvement. The research stages include requirement planning, design and development of the API server prototype, implementation of RESTful web services, as well as system testing and evaluation.The results show that the developed academic application server is able to provide stable services for managing student and grade data, is easily integrated with other applications, and has a faster response time compared to previous manual systems. User Acceptance Testing (UAT) and Black Box testing indicate that all main functions operate as expected without significant errors. In addition, the server architecture is modular, allowing further development, including integration with mobile-based applications. Thus, this study proves that the implementation of web services using the RAD method is effective in supporting efficient, accurate, and integrated academic data management in higher education environments

Keywords: *Academic Information System, Web Service, Rapid Application Development, Application Server, Student Data*

1. Introduction

The rapid development of technology has encouraged many higher education institutions in Indonesia to digitalize academic systems to improve the efficiency of managing student data and grades. Previously, many institutions still used manual methods, such as recording student data and grades using spreadsheets, which often caused problems related to data accuracy and synchronization. According to Fitriani et al. in the *Journal of Information System, Applied, Management, Accounting and Research (JISAMAR)*, the implementation of web-based information systems has proven to accelerate data processing and minimize input errors that frequently occur in manual processes.

Along with the growing need for academic application integration, there is still a need to develop web service–based application servers that can connect various data sources in real time. The implementation of web services enables distributed data exchange between systems without altering the core application architecture. In addition, the Rapid Application Development (RAD) method has proven effective in accelerating system development by emphasizing iterative processes and direct user involvement during the design stages.

Therefore, research on the development of a web service–based academic application server using the RAD method is highly relevant to support efficient, secure, and integrated academic data management. Previous research published in the *Journal of Information Systems and Informatics Engineering (JISTI)* states that web-based servers can act as central academic data distribution hubs that connect various applications within a campus environment. Based on this, the objective of this study is to design and develop an academic application server capable of effectively managing student data and grades, facilitating fast, flexible, and easily extensible service integration.

2. Literature Review

In the *Journal of Informatics and Educational Technology (JITS)*, an academic information system based on web technology was designed to support campus administrative activities such as student data processing, scheduling, and grading in a centralized manner. The study employed the Waterfall method to design student data and grade modules that could be accessed online through a browser. The results showed that the system reduced grade input errors and accelerated the distribution of academic information.

Another study in JISAMAR discusses the implementation of Rapid Application Development (RAD) in a web-based academic information system. The research focuses on accelerating system development through iterative design, prototyping, and user feedback. The testing results indicate a 40% improvement in development time efficiency compared to traditional methods. This is relevant to the present study, as it maintains system quality through modular interaction and functional validation.

Research published in JISTI examines the use of the Laravel framework in developing a web-based academic information system. The study applied a software engineering experimental method by testing API performance using routing features, middleware, and Eloquent ORM. The results show that Laravel improves data retrieval efficiency by up to 25% and enhances API security through token-based authentication. This supports the development of modern academic servers, as frameworks like Laravel simplify the creation of secure, modular, and maintainable server-side APIs.

A web server is a web-based service concept that allows an application to provide data or specific functions to other applications through standard protocols such as HTML, XML, and JSON. In academic systems, web servers are used to provide structured services such as student data, grades, and other academic processes, enabling client applications to access data without directly connecting to the database.

An Academic Information System (AIS) is a platform used by higher education institutions to manage administrative and academic student data, including biodata, grades, course registration (KRS), and academic reports. The development of an academic application server aims to extend AIS capabilities by separating data services through web servers so that they can be accessed by other applications, thereby making system integration more efficient.

A database is a collection of data stored in a structured manner to facilitate access, management, and updates. Database processing is handled by a Database Management System (DBMS), which provides data storage, security, access control, and processing functions. In academic systems, databases serve as the central repository for student and grade information, enabling web services to retrieve data quickly and consistently. Database structures are generally designed using relational concepts.

3. Research Method

The research method used in this study is Rapid Application Development (RAD), a system development method that emphasizes speed, flexibility, and iterative prototyping. This method was selected because it is suitable for academic systems that have relatively complex workflows while requiring rapid development and direct user testing.

The first stage in RAD is Requirements Planning**,** which involves collecting system requirements through interviews, observations, and academic document analysis. The second stage is Workshop Design and Prototyping, where the researchers design and develop API server prototypes, databases, user flows, and endpoints for managing student and grade data. These prototypes are tested directly by academic administrators to ensure functional suitability and ease of use. The process is repeated until the system meets user requirements. Previous studies by STEKOM (2022) and UNIS (2024) support the effectiveness of RAD in accelerating system development.

The next stage is Implementation, which includes REST API development, functional testing, security integration using tokens or JWT, and API integration with client applications such as academic dashboards or grading systems. The final stage is Testing and Evaluation, where the system is tested through User Acceptance Testing (UAT). Academic administrators, lecturers, and operators conduct direct testing of the API and its integration with front-end applications. User feedback is then used for final improvements to ensure system readiness for deployment.

4. Research Results

The results indicate that the development of a web service–based academic application server using the RAD method significantly improves the speed of processing student and grade data. The developed system produces a stable API that is easy to integrate and has a faster response time compared to previous manual systems.

UAT results show that academic administrators find it easier to manage student data, perform data searches, and access grades. Similar findings are reported in the *EduTIK Journal* (2022), which states that web service integration reduces data input errors and accelerates academic reporting processes. Black Box testing confirms that all main functions—such as grade input, student data updates, grading, and monthly reports—operate correctly without errors.

Additionally, the research produces a server architecture that can be further developed into mobile applications. These findings are consistent with studies published in UNIS (2024) and JISTI (2025), which state that modular API systems provide high flexibility for feature expansion. Therefore, the system is considered suitable as a foundation for digital-based academic systems.

4.1. System Interface Descriptions

1. **Login Interface**

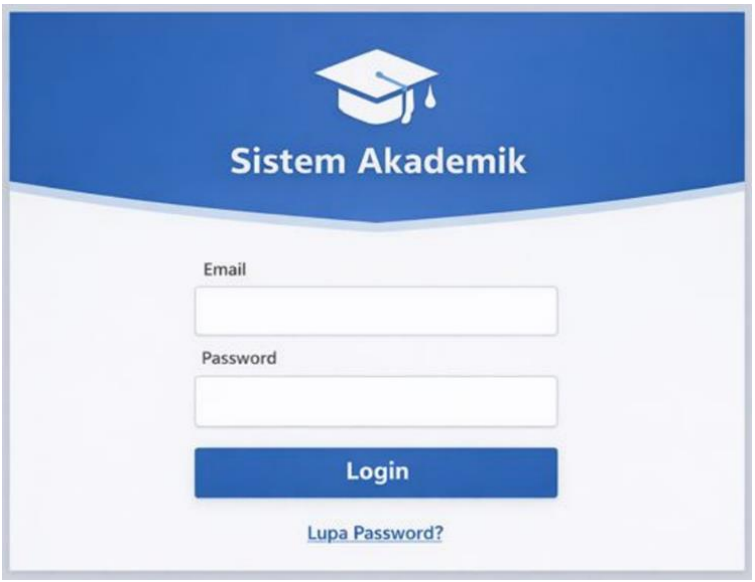

Fig.1 : Login

The login interface functions as the main gateway to access the academic system. It displays the title “Academic System” and a graduation cap icon representing the application identity. The login form consists of email and password input fields used for authentication. A “Forgot Password?” feature is also available to assist users in account recovery.

2. **Dashboard Interface**

Fig.2 : Dashboard

This interface appears after an administrator successfully logs in. It displays navigation menus and user account information. The dashboard summarizes key data such as the number of students, course registrations (KRS), study results (KHS), and grades in the form of information cards. It also includes announcements and academic statistics presented in graphical form to facilitate monitoring.

3. **KHS Interface**

Fig.3 :KHS

The KHS (Study Result Card) interface displays student academic results for a selected semester and academic year, including course codes, course names, credit units (SKS), letter grades, and grade weights. The total credits earned and GPA are shown as a summary of academic performance.

4. **KRS Interface**

Fig.4 : KRS

The KRS interface is used to manage students' study plans for a particular semester and academic year. Students can select available courses along with credit units and schedules using checkboxes. The selected data are saved by clicking the "Save KRS" button.

5. **Grades Interface**

Fig.5 : Grades

The student grade data interface displays grade recapitulation based on academic year and semester. Information includes student ID, student name, course name, assignment scores, midterm exam scores, and final grades. A search feature is provided to quickly locate specific student data.

5. Conclusion

Based on the research results, it can be concluded that the development of a web service–based academic application server using the Rapid Application Development (RAD) method improves the efficiency and accuracy of managing student data and grades. The developed system provides stable, responsive, and easily integrated API services, thereby minimizing data synchronization issues and delays commonly found in manual systems.

References

[1] A. Karmakar, S. S. Roy, F. Vercauteren, and I. Verbauwhede, "Efficient finite field multiplication for isogeny-based post-quantum cryptography," 2017, doi: 10.1007/978-3-319-55227-9_14.
[2] A. M. H. Pardede, M. Zarlis, and H. Mawengkang, "Optimization of health care services with limited resources," International Journal on Advanced Science, Engineering and Information Technology, vol. 9, no. 4, pp. 1444–1449, 2019, doi: 10.18517/ijaseit.9.4.8348.
[3] A. M. H. Pardede, Y. Maulita, and R. Buaton, "Application modeling IPv6 (Internet Protocol Version 6) on e-ID card for identification number effectiveness and efficiency in population registration," in Journal of Physics: Conference Series, vol. 978, no. 1, 2018, doi: 10.1088/1742-6596/978/1/012017.
[4] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities," IEEE Consumer Electronics Magazine, vol. 5, no. 3, pp. 60–70, 2016, doi: 10.1109/MCE.2016.2556879.
[5] W. A. Jabbar, W. K. Saad, and M. Ismail, "MEQSA-OLSRv2: A multicriteria-based hybrid multipath protocol for energy-efficient and QoS-aware data routing in MANET–WSN convergence scenarios of IoT," IEEE Access, 2018, doi: 10.1109/ACCESS.2018.2882853.
[6] D. Niyigena, C. Habineza, and T. S. Ustun, "Computer-based smart energy management system for rural health centers," 2016, doi: 10.1109/IRSEC.2015.7455005.
[7] F.-Z. Younsi, A. Bounnekar, D. Hamdadou, and O. Boussaid, "SEIR-SW: A simulation model of influenza spread based on small-world networks," Tsinghua Science and Technology, vol. 20, no. 5, pp. 460–473, 2015.