ELKOM (Jurnal Elektronika dan Komputer)
p-ISSN : 1907-0012 (print) e-ISSN : 2714-5417 (online)
Vol. 19, No. 1, Juli 2026
https://journal.stekom.ac.id/index.php/elkom

Development of Higher Education Key Performance Indicator Monitoring System using Rapid Application Development Method

Adi Widianto[1], Laurentinus[2]
[1] Department of Digital Business, Universitas Bangka Belitung, Bangka, Indonesia
[2] Department of Digital Business, Universitas Bangka Belitung, Bangka, Indonesia
Email author: adi-widianto@ubb.ac.id[1], laurentinus@ubb.ac.id[2]

Article Info

Article history:

Received May 19, 2026
Revised May 25, 2026
Accepted June 4, 2026
Available: July 12, 2026
Published: July 30, 2026

Keywords:

Rapid Application;
Development;
Key Performance Indicator;
Higher Education;
Monitoring System;
Web-based Application

ABSTRACT (10 PT)

Key Performance Indicators (KPIs) have a very essential role in higher education institutions in order to maintain the quality standards and to meet the requirements of the accreditation. But manual input and tracking typically results in data silos, data loss and delayed reporting. Furthermore, the system must be integrated with the subsystems that have already been built, which makes the development process longer and more complex. The purpose of this project is to design a university web-based KPI monitoring system by using Rapid Application Development (RAD) method to increase efficiency and reduce data loss. This choice is done because it allows a fast implementation in a real environment with an already existing subsystem. It started development in June 2025 and continued through August. The user input was looped as per criteria. The KPI tracking system has been implemented in one semester already. RAD has already proven to be the right methodology, as it achieved the goal of fast implementation and satisfied functional requirements.

Corresponding Author:

Adi Widianto,
Universitas Bangka Belitung
Jl. Merdeka No. 4, Kec. Merawang, Kabupaten Bangka
Email: adi-widianto@ubb.ac.id

1. INTRODUCTION

Key Performance Indicators (KPI) are metrics used to calculate the qualitative and quantitative performances of an organization or institution. In higher education institutions, KPI is a tool to measure the output and administrative quality of the university [1]. The Ministry of Higher Education, Science, and Technology determined the indicator used for Indonesian higher education institutions. Currently, there are officially 8 indicators, but the university can add another for internal purposes. The subjects included are a graduate's job, student achievement, lecturer, partnership, learning activity, and

international accreditation. The various data sources make the reporting task with manual input tedious and challenging. The problems appear with manual input and fragmentary systems: data silos, loss of data, and delayed reporting.

A Key Performance Indicators monitoring system is an online application that can be accessed anytime and anywhere to monitor and evaluate KPI [1]. Universities or higher education institutions need this system to keep their quality standard and fulfill accreditation criteria. Without this system, the various data needed for the KPI are scattered everywhere, resulting in a long and difficult report process, even when there are already finished subsystems that produce that data. The KPI report is also crucial as a document that decides the accreditation of the study program and university, making the integrated systems that handle various data necessary to produce the report document faster and more efficiently. The design and implementation of a KPI monitoring system require a robust development method to effectively integrate existing subsystems. The agile methodologies and modular architecture allow smooth data flow and improve collaboration between components, which results in fast and more accurate KPI reports supporting the accreditation process.

There are already works that implement KPI monitoring systems and other information systems in higher education institutions. KPI monitoring systems were found to be perceived as useful and easy to use by users in Malaysian universities [1] and aid in the overall effectiveness of information systems in improving educational outcomes and operational efficiency. [2] presents a systematic review on the success and impact of different information systems in higher education. It obtains the results; the systems greatly enhance the institution's business process by streamlining operations, improving communication, and providing data-driven insights that make it easier for better decision-making. Another example is a guidance system for students in higher education, already developed using the Software Development Life Cycle (SDLC) Waterfall model, which can accommodate the interests of each stakeholder [3]. The waterfall method was also implemented in a web-based tracer study system in higher education, resulting in a system that allows an efficient survey process [4]. Other systems, like the training application system, are implemented in a university using Rapid application development (RAD), which resulted in quick implementation and still gave developers control over the development process [5].

Rapid application development [6] is already used as a methodology to build software in other fields. Agricultural robot navigation system developed by RAD significantly shortened the development time [7]. In the industry field, the Enterprise Resource Planning (ERP) system was built in a relatively short time by using RAD [8]. The ERP system was tested using the User Acceptance Test (UAT) method and still matches the user's needs. Customer Relationship Management (CRM) is also designed using RAD [9]; the result is time improvement in the payment process. The payment system for the fashion boutique, developed using RAD, increases the speed of system development while improving software quality [10]. RAD implemented in the Library Information System can solve the library's business processes within a short development process [11].

For the development of the KPI monitoring system, we selected rapid application development as the method due to its importance and flexibility in integrating the completed subsystems into the working environment. There is no study that implements and evaluates RAD on top of already-built subsystems, particularly in higher education information systems. The RAD technique is supposed to satisfy the functional requirements of the system and to accelerate the project work.

2. METHOD

The system development method used in this research is Rapid Application Development (RAD). RAD can accelerate the cycle of the development of a system or application, thus saving the valuable resources of time and labor [6]. RAD is suitable for this research because the system is being developed by a small team that includes a user who understands the exact business process [12]. There are 4 stages of RAD: the Requirements Planning Phase, the User Design Phase, the Construction Phase, and the Cut-Over Phase [13], which is illustrated in Figure 1.

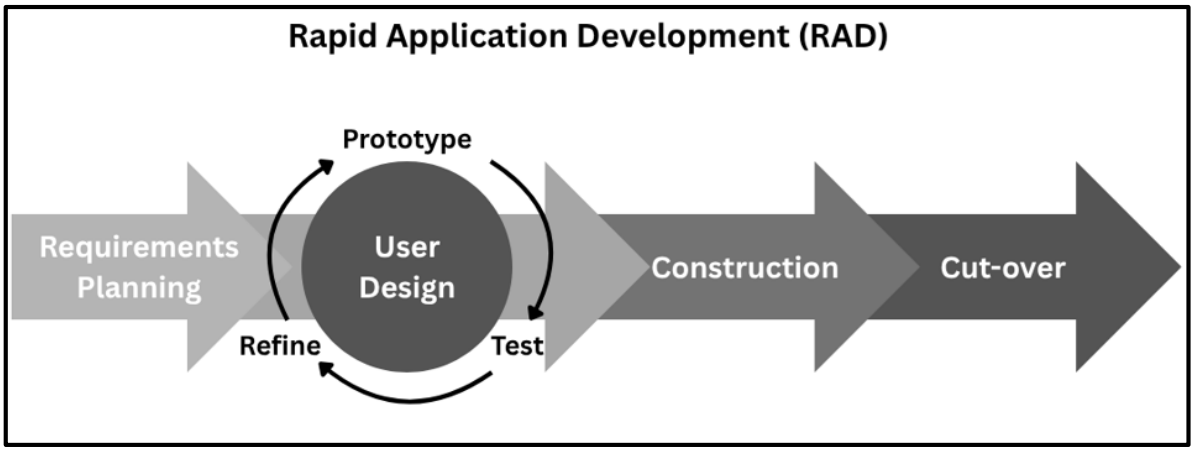

Figure 1. Rapid Application Development (RAD) phases

The steps of this research development, which uses the RAD method, are explained as follows [14]:

1. *Requirement Planning Phase*
   During this phase, we gather functional requirements for the system from users [15]. In this case, we collected the requirement from our development team members. The members are campus officials and consist of Dean and Head of Department.
2. *User Design Phase*
   At this stage, we design the prototype using diagrams and mockups. The prototype is built per requirement, which is then tested by the user with feedback. Then the developer can refine the prototype based on user feedback. This process is iterated until all requirements are fulfilled [16]. We visualize the prototype using a use case diagram and mockup page. We also design the database schema for the system in this phase.
3. *Construction Phase*
   In this stage, we build the monitoring system using the prototype and diagrams from the user design phase. We integrated the third-party cloud storage [17] subsystem in this stage. In implementation, we use a web-based application LAMP stack (Linux, Apache, MySQL, PHP) due to rapid prototyping and broad compatibility with existing web servers [18]. Additionally, we developed the system using the Laravel framework based on the Model-View-Controller (MVC) architecture [19]. The result or output in this stage is a system for monitoring KPIs in higher education.
4. *Cut-Over Phase*
   Lastly, we end the development process with testing the system using the black-box method to ensure the system meets all requirements and functions properly [20]. This test does not evaluate the source code of the program.

3. RESULT AND ANALYSIS

3.1. Requirement Planning

Based on the interviews with users, we obtained a comprehensive understanding of the system's functional and non-functional needs. Functional requirements include the user can create, read, update, and delete quarterly KPI reports per study program. System also can show a summary dashboard of the quaterly faculty-level KPI report and download the report in spreadsheet format. Non-

functional requirement is a web-based application that can be accessed from desktop and mobile devices.

3.2. User Design

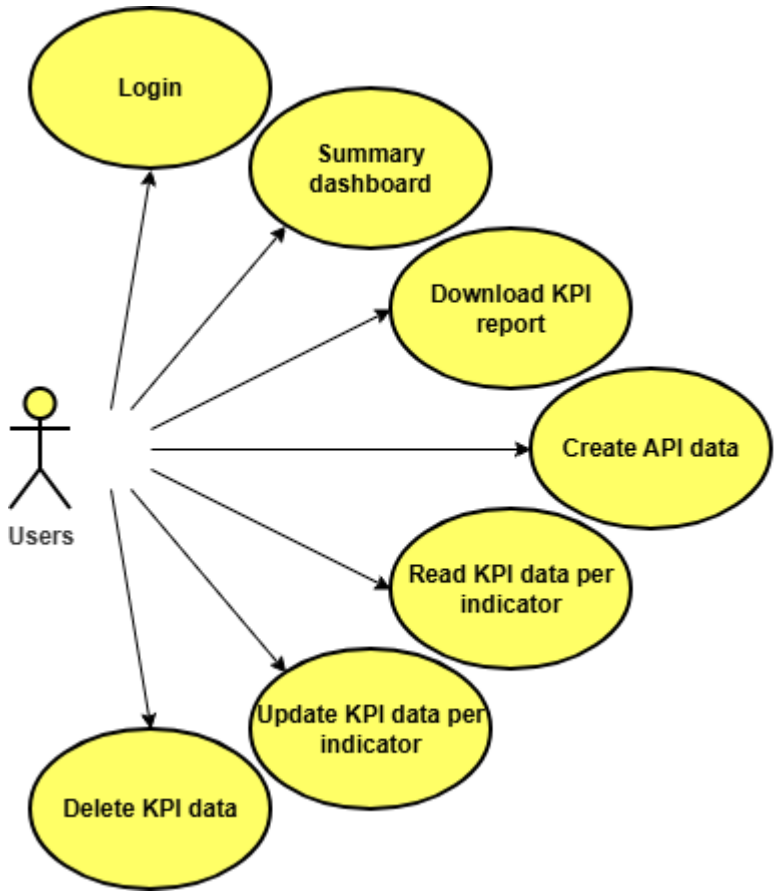

Figure 2. Final prototype use case diagram

Figure 2 shows the final use case diagram, which was created to fulfill all specifications in the requirement planning phase. At first, the developer designs the initial use case diagram. Initial design is translated to mockup pages, which are then tested by users who provide feedback when needed. Then, the next design is created to complete the feedback. This process iterated until it produced the final use case diagram and mockup prototype that met all requirements.

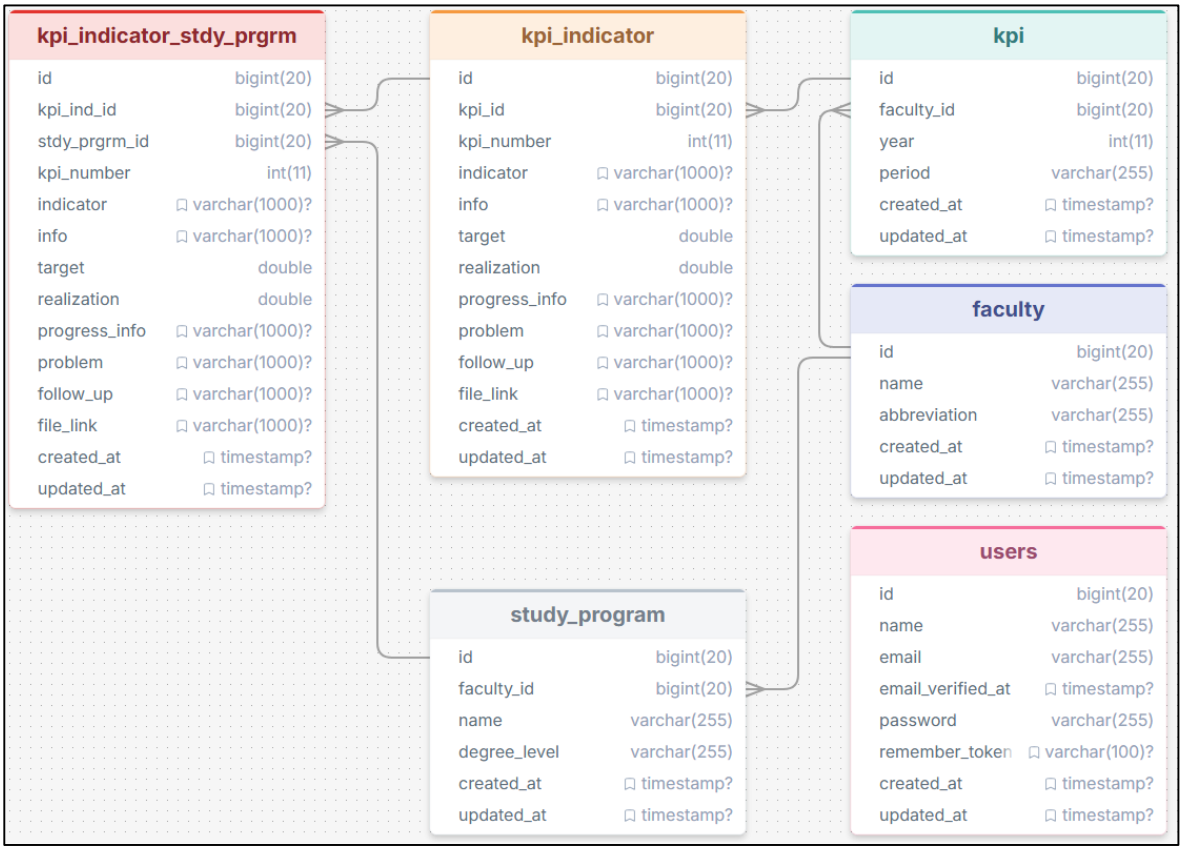

Figure 3. Database schema of Higher Education KPI Monitoring System

The database schema was then designed based on the use case diagram as shown in Figure 3. The schema integrates multiple study programs into a faculty-level KPI report. We also combine all users into one role to simplify the system, which allows for easier management and reduces the overhead associated with maintaining multiple roles. For now, the system does not need the complex role system to meet all the requirements.

3.3. Construction

Based on the final prototype and the diagram in the user design phase, the rapid development was started. We construct the system on our server to ensure immediate access upon completion. Figure 4 shows the system's login page. After logging in, the user can access the home dashboard and KPI menu. To fulfill the non-functional requirement of the system, we implement a responsive web design so it can be accessed with a fluid user interface on both desktop and mobile devices, as shown in Figure 5. We also add a summary graph and download a KPI report in this dashboard based on requirement planning.

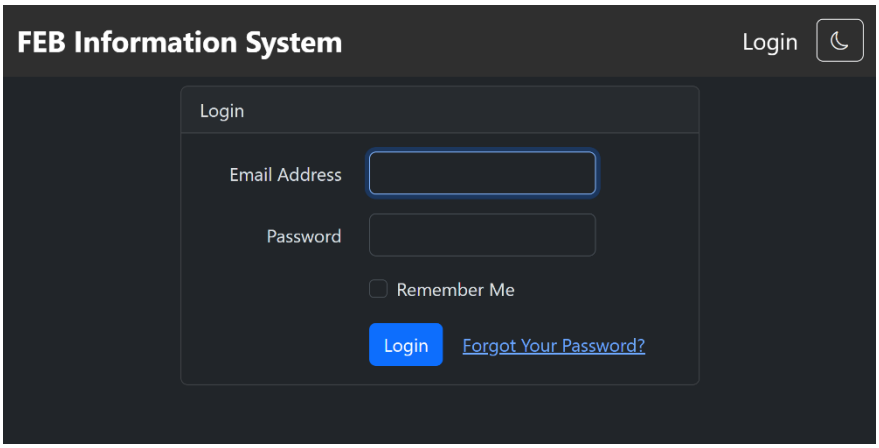

Figure 4. Login page when system accessed by browser

Figure 5. Dashboard page in desktop version (a) and mobile version (b)

Figure 6 shows the KPI menu where the user can create, read, update, and delete (CRUD) the quarterly faculty-level KPI report. After creation, the system automatically loads the template for the current KPI standard and criteria, as shown in Figure 7. The user must enter the percentage progress for each indicator and attach the URL of the supporting document. That URL will connect to a third-party cloud storage subsystem. A form to fill out for each indicator is shown in Figure 8. All the changes are saved in the server database, which makes monitoring and editing KPIs possible anytime and anywhere via the internet.

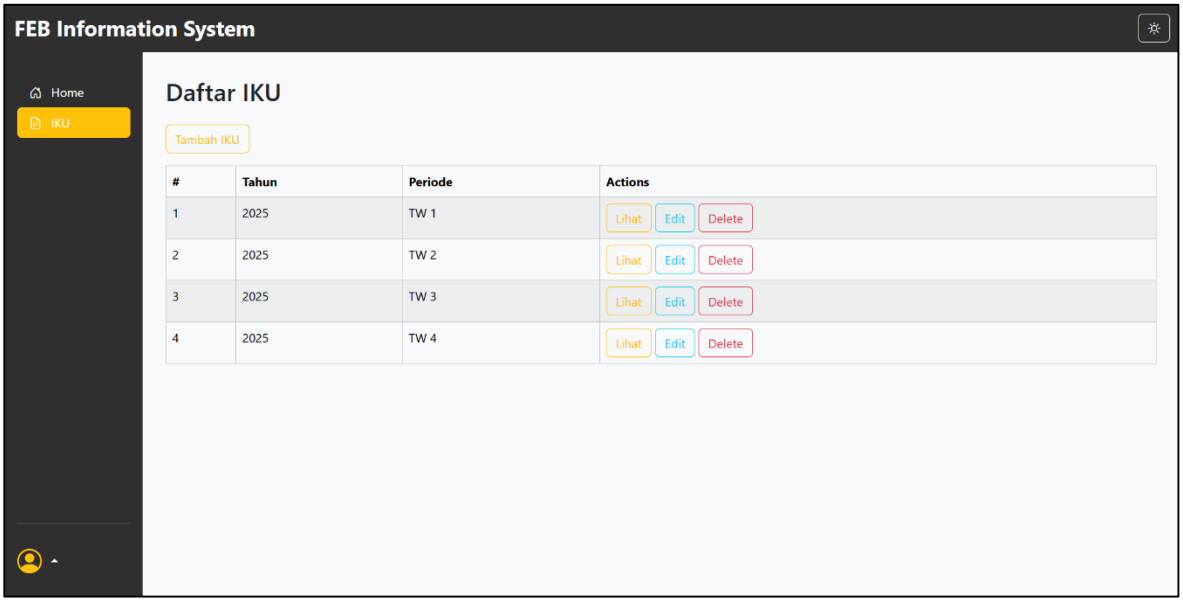

Figure 6. Faculty-level KPI CRUD page

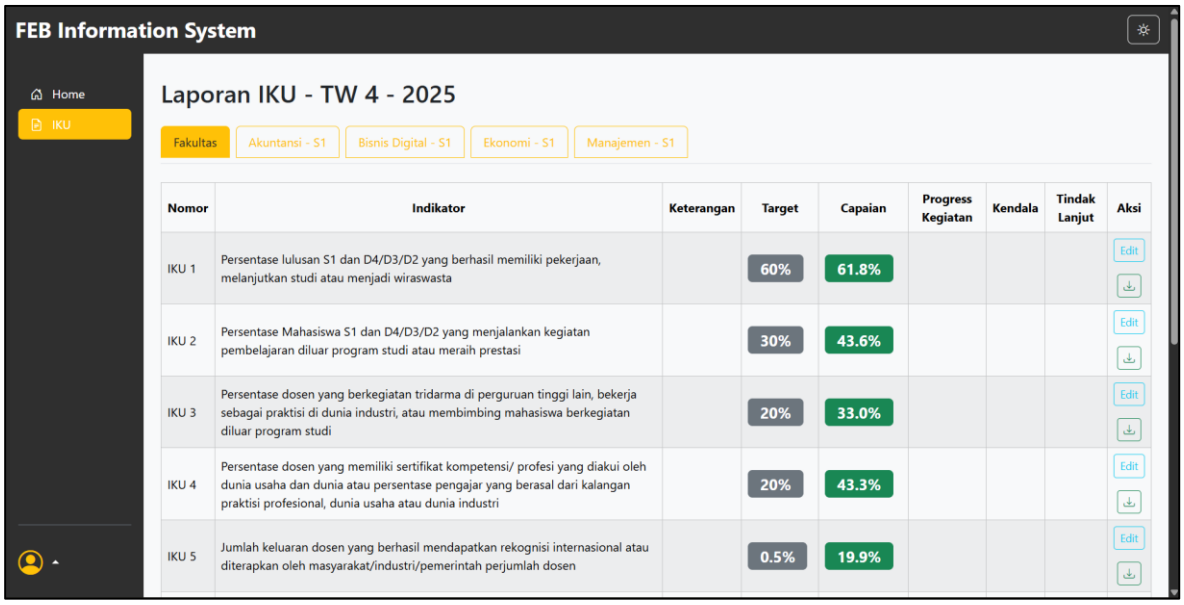

Figure 7. Study program-level KPI read operation page

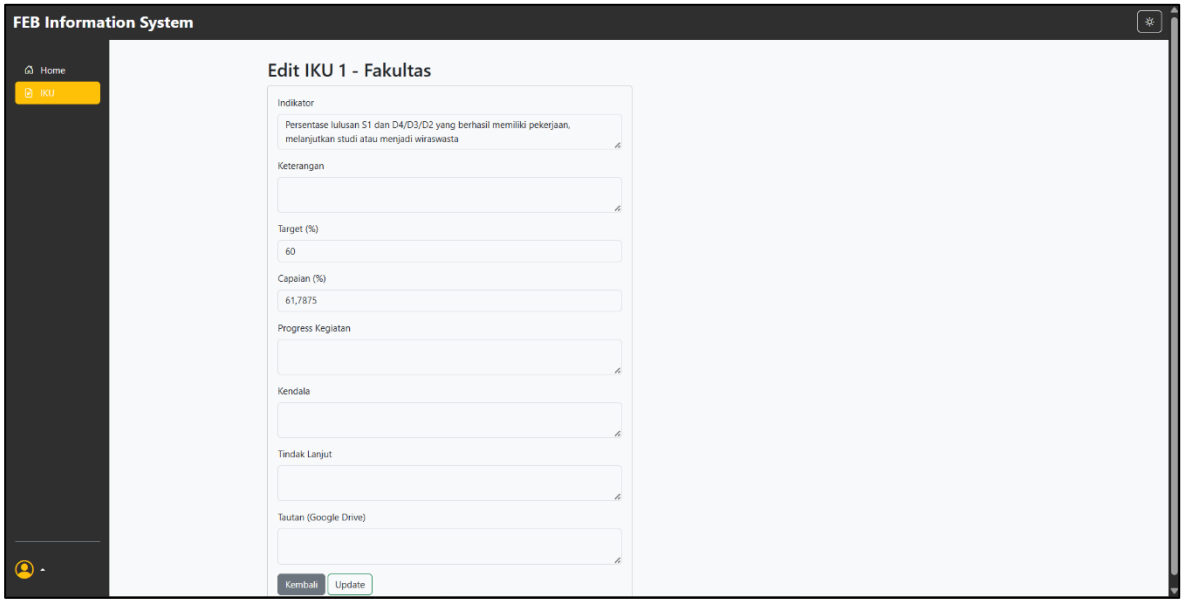

Figure 8. Study program-level KPI edit operation page

3.4. *Cut-Over*

Black-box testing is done to evaluate the proposed systems. We test seven main scenarios from the use case diagram. Table 1 shows the black-box test result.

Table 1. Black-box testing result

Scenario	Task	Expected Result	Test Result
Login	Enter email and password, click 'Login' button	User enters the main page	Compatible
Summary dashboard	Click 'Home' menu	User see summary graph	Compatible
Download KPI report	Click 'Download KPI Report' button	User download KPI report	Compatible
Create KPI data	Enter 'KPI' menu, click 'Add KPI' button, filling new KPI form	User add KPI data	Compatible
Read KPI data per indicator	Enter 'KPI' menu, click 'Look' button	User see KPI data per indicator	Compatible
Update KPI data per indicator	Enter 'KPI' menu, click 'Look' button, click 'Edit' button, filling edit KPI form	User update KPI data per indicator	Compatible
Delete KPI data	Enter 'KPI' menu, click 'Delete' button	User delete KPI data	Compatible

The development of the application started in June 2025 and continued till August. The user feedback was iterated as per the requirement. The KPI monitoring system was then implemented in a

real environment for one semester. Based on user reviews, the system successfully solves the data silos problem and makes KPI reports more efficient in time and labor.

4. DISCUSSION/CONCLUSION

The KPI monitoring system for higher education has been completed at the requirement planning, user design, construction, and cut-over stages. At the requirement planning stage, a list consisting of functional and non-functional requirements has been collected. At the user design stage, a requirement is then turned into a prototype that can be clicked on the buttons, which is tested to get feedback from the user. After all requirements are fulfilled, the full application is built during the construction stages. Based on the results of functional testing using the black-box method, it can be concluded that in all seven main scenarios, the application works as expected and is compatible with solving data silo problems in fragmentary systems. RAD, or Rapid Application Development, has proven to be a suitable method to be used in fast-paced development with finished subsystem integration. Future work includes calculating the performance and user satisfaction of the system from the user's perspective.

REFERENCES

[1] B. Muniandy, M. Y. Ong, K. K. Phua, and S. L. Ong, "Assessing Key Performance Indicators Monitoring System (KPI-MS) of a university using Technology Acceptance Model," *Int. J. Soc. Sci. Humanit.*, vol. 1, no. 3, pp. 171–176, 2011, doi: 10.7763/ijssh.2011.v1.30.

[2] T. Mijač, M. Jadrić, and M. Ćukušić, "Measuring the success of information systems in higher education – a systematic review," *Educ. Inf. Technol.*, vol. 29, no. 14, pp. 18323–18360, 2024, doi: 10.1007/s10639-024-12564-8.

[3] H. J. Christanto and Y. A. Singgalen, "Analysis and Design of Student Guidance Information System through Software Development Life Cycle (SDLC) dan Waterfall Model," *J. Inf. Syst. Informatics*, vol. 5, no. 1, pp. 259–270, 2023, doi: 10.51519/journalisi.v5i1.443.

[4] M. R. Priyanto and D. Ramayanti, "Design and Development of a Web-Based Tracer Study System in Higher Education Using the Waterfall Method (Case Study at Dian Nusantara University)," *Instal J. Komput.*, vol. 16, no. 7, pp. 370–379, 2025.

[5] N. M. N. Daud, N. A. A. A. Bakar, and H. M. Rusli, "Implementing Rapid Application Development (RAD) methodology in developing practical training application system," in *Proceedings 2010 International Symposium on Information Technology - System Development and Application and Knowledge Society, ITSim'10*, IEEE, 2010, pp. 1664–1667. doi: 10.1109/ITSIM.2010.5561634.

[6] M. A. Fauzi, H. Tribiakto, A. Moniva, I. Khalid Ilyas, and E. Utami, "Systematic Literature Reviews on Rapid Application Development Information System," *Bull. Comput. Sci. Electr. Eng.*, vol. 4, no. 1, pp. 57–64, 2023.

[7] R. M. Zhao *et al.*, "Rapid development methodology of agricultural robot navigation system working in GNSS-denied environment," *Adv. Manuf.*, vol. 11, no. 4, pp. 601–617, 2023, doi: 10.1007/s40436-023-00438-0.

[8] M. R. G. Qowindra and J. Wiratama, "Development of Enterprise Resource Planning (ERP) using the Rapid Application Development (RAD) Method for the Garment Industry in Indonesia," *G-Tech J. Teknol. Terap.*, vol. 7, no. 2, pp. 504–513, 2023, doi: 10.33379/gtech.v7i2.2296.

[9] R. Syabania and N. Rosmawani, "Perancangan Aplikasi Customer Relationship Management (Crm) Pada Penjualan Barang Pre-Order Berbasis Website," *Rekayasa Inf.*, vol. 10, no. 1, pp. 44–49, 2021.

[10] S. Rokhaila, A. Kurniawati, and D. Agusten, "Sistem Penggajian Berbasis Web untuk Butik Busana Fashion: Implementasi Metode RAD," *G-Tech J. Teknol. Terap.*, vol. 8, no. 2, pp. 1114–1124, 2024, doi: 10.33379/gtech.v8i2.4108.

[11] D. Hariyanto, R. Sastra, F. E. Putri, S. Informasi, K. Kota Bogor, and T. Komputer, "Implementasi Metode Rapid Application Development Pada Sistem Informasi Perpustakaan," *J. JUPITER*, vol. 13, no. 1, pp. 110–117, 2021.

[12] P. Beynon-Davies, C. Came, H. Mackay, and D. Tudhope, "Rapid application development (Rad): An empirical review," *Eur. J. Inf. Syst.*, vol. 8, no. 3, pp. 211–232, 1999, doi:

10.1057/palgrave.ejis.3000325.
[13] R. Naz and M. N. A. Khan, "Rapid applications development techniques: A critical review," *Int. J. Softw. Eng. its Appl.*, vol. 9, no. 11, pp. 163–176, 2015.
[14] J. K. Wiredu, N. S. Abuba, B. Atiyire, and R. W. Acheampong, "Design and Implementation of Online Crime Report System using Rapid Application Development (RAD) Methodology," *Asian J. Res. Comput. Sci.*, vol. 17, no. 8, pp. 100–115, 2024, doi: 10.9734/ajrcos/2024/v17i8493.
[15] M. F. T. Nugraha, A. Kusnadi, a nd F. A. T. Tobing, "Implementation of the Rapid Application Method Development (RAD) in Clothes Sale Website Design," *Asian Inst. Res.*, vol. 6, no. 2, pp. 10–18, 2023.
[16] S. Chandra, R. Sutomo, and J. Wiratama, "Design And Development of Car Sparepart Sales Information System For Web-Based Using RAD Method On UMKM Sinar Seroja," *G-Tech J. Teknol. Terap.*, vol. 7, no. 2, pp. 494–503, 2023, doi: 10.33379/gtech.v7i2.2019.
[17] I. Gupta, A. K. Singh, C. N. Lee, and R. Buyya, "Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Fu`ture Directions," *IEEE Access*, vol. 10, pp. 71247–71277, 2022, doi: 10.1109/ACCESS.2022.3188110.
[18] M. Slabinoha, S. Melnychuk, V. Kropyvnytska, and B. Pashkovskyi, "Comparative Analysis of Using Event Sourcing Approach in Web Application Based on the LAMP Stack," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 166, Springer, 2023, pp. 257–269. doi: 10.1007/978-981-99-0835-6_19.
[19] E. S. L. Cavero, R. A. S. Ramirez, and A. N. R. Caururu, "Software Rapid Application Development (RAD) for Web Application Generation and Web-Programming Learning Support at a University in Lima," in *ICCSIT 2025 - Proceedings of the 2025 18th International Conference on Computer Science and Information Technology*, 2026, pp. 144–154. doi: 10.1145/3783862.3783880.
[20] L. Laurentinus and A. Widianto, "Perancangan Aplikasi Antrian Pendaftaran Pasien Puskesmas Selindung Berbasis Android," *Elkom J. Elektron. dan Komput.*, vol. 18, no. 2, pp. 226–240, 2025, doi: 10.51903/elkom.v18i2.3267.