Development of SIAKAD Applications in Balikpapan Schools using APXP: Advanced Personal Extreme Programming

Aidil Saputra Kirsan[1], Nursanti Novi Arisa[2], Vika Fitratunnany Insanittaqwa [3]
Institut Teknologi Kalimantan [123]
Jl. Soekarno Hatta, Kota Balikpapan, Kalimantan Timur, Indonesia
Email: aidil@lecturer.itk.ac.id[1], nursanti.novi@lecturer.itk.ac.id[2], vika.fitra@lecturer.itk.ac.id[3]

Abstrack - The iteration process in Personal Extreme Programming (PXP) consists of Initialization, Design, Implementation, Testing, and Retrospective. There are still areas for improvement in this method. In the future, this method will be a problem because it can make the application unable to run after the retrospective process. One of the problems is the version update app on Android. We propose an Advanced PXP model (APXP) for an academic information system (SIAKAD) in Balikpapan Schools that contains PXP modifications at the beginning and end of the process. At the middle stage of APXP is system testing. Each process from 1-11 iterations produces Valid results. We add Deployment methods such as Blue-Green Deployment Strategy. The deployment process can change very quickly once the switch service changes in the APXP method. Furthermore, the deployment process has almost no issues with different versions of the service.

Keywords - PXP, APXP, Retrospective, Android, SIAKAD, Blue-Green Deployment.

I. INTRODUCTION

In software development, teams work together to create a working product by implementing a set of practices or a framework to ensure that the process is manageable and organized. The oldest methodology to develop software is the Waterfall Model, first documented by [1] and is modified by [2]. With this model, the software is developed straightforwardly in its life cycle, consisting of a few stages: planning, analysis, design, development, testing, implementation, and maintenance. As people discovered more ways to develop software, several other methodologies emerged to provide alternatives to develop software with varying conditions, such as Incremental Model, Spiral Model, Rapid Application Development, and others [3]. Agile software development is one of the emerging methods which is more suitable for smaller-scale projects and is widely used due to its high productivity and flexibility to changes [4].

Agile software development is one of the widely used practices to deliver faster products to adapt to quick software development processes. With an iterative concept at its core, this method ensures that the customer can receive the products faster and in incremental phases. As a result, the customers are more satisfied with a shorter development life cycle, lower bug rates due to the evaluation of each iteration, and more flexibility in requirement changes [5].

Extreme Programming (XP) is one of the Agile development methods that focus on the programming process, feedback, and communication with the clients [6]. It is implemented in iterations in short spans in response to the problem where requirements change frequently. Over time, there is a need to modify this method to suit a single-person team better to accommodate programmers who work mainly independently instead of a programmer who works as a part of a team. This method is known as Personal Extreme Programming or PXP [7]-[8]. It combines

XP and Personal Software Process (PSP) practices, a set of guidelines used by engineers to measure and improve their process in a software development project.

PXP consists of Requirement, Planning, and Iterative Development Phase. The Iterative Development Phase consists of more minor phases: Initialization, Design, Implementation, Testing, and Retrospective [8]. There is still room to improve this method to ensure the programmer has essential knowledge about the project. The software is received well by the client until it is deployed and integrated well within the system. For this reason, we proposed an improved model of Advanced PXP (APXP), which contains modifications of the PXP at the beginning and the end of the process. The modifications are meant to prepare the developers to undertake the project by acquiring more information about the client's needs beforehand. The final product can be delivered with minor software failure.

PXP is based on the combination of XP and PSP to accommodate a single-person team or autonomous developers. The research [7] laid out the Process Script of PXP as basic instructions and decision-making processes used as a guideline. PXP development phases are introduced by the research [8] and are widely used as the main PXP framework. PXP begins with the Requirement and Planning phases as a preparation for the Iterative phases. The main development activity (i.e., programming) is done in the Iterative phases: Initialization, Design, Implementation, Testing, and Retrospective. The developer keeps the Logs with information regarding tasks planning and actual duration, improvement suggestions, defects count, and details.

The use of PXP as a software development method had been done in the following research. The research [6] built a library system with the PXP method and Moscow prioritization technique to categorize the system requirements. This study explained that the PXP method developed the application quickly and adapted to clients' needs who potentially have changed. The Moscow method used in the PXP methodology can optimize the client's needs in the planning and processing process. The sorted requirements are distributed in the development iterations. The Research [9] used the PXP method to develop web-based information systems to help universities' job training activities. After the system is released, a survey is conducted and concluded that the application has satisfied the customer's needs. Moreover, the PXP method has provided convenience to determining the development cycle duration and facilitated the developer to build the code directed according to user stories.

An improvement to PXP has been proposed in the literature. An effort to increase the velocity (i.e., productivity measured in story points completed in an iteration) is made by integrating the Pomodoro technique. Timers became an essential part of the development process [10]. This study found that the Pomodoro technique made the team more aware of the time. Consequently, the researchers are more focused at work, actively avoid interruptions, and give priorities more explicitly.

The research [11] implemented a blue-green strategy to manage the re-deployment of microservices in a system on a cloud environment. With this strategy, the two deployment environments of the service are running while changes are being carried out in one of them. The benefits gained are eliminating downtime due to application deployment, decreasing the risk of errors during version changes, and the availability of a backup system in case of worst-case scenarios to the primary service.

In website development, one of the PHP programming language frameworks used is CodeIgniter. CodeIgniter is an open-source application that using a PHP framework with MVC (Model View Controller). The framework has advantages in terms of complete libraries and packages. Therefore, it makes it easier for developers to design a website [12]. Currently, the creation of information systems is not only developed on a website-based but also mobile-based. Mobile devices have become a primary need that everyone must have to collect information. One of the mobile operating systems used is Android. Android is an open-source operating system. Its open-source nature makes it easier for developers to develop android applications [13]. Operating system developers permit programmers to create, modify, develop and distribute applications. By utilizing mobile technology that implements the Android operating system, it is possible to create an application that can reach people according to its purpose.

With more innovations in science and technology, there is a demand to make every institution's business process more efficient and accessible. Al-Azhar 58 Balikpapan School still traditionally manages their students' data. The data is stored in a local computer and is managed with standard Office software. For example, this data can only be shared on certain occasions during the semester to inform students and parents about their grades or payment records. There is a need to make the data more accessible to keep the students and parents informed of the student's progress at school solution to this problem. Two software systems are developed: a web-based Academic Information Systems (SIAKAD) and a mobile-based notification system to access SIAKAD from any mobile device. The systems are developed with the proposed APXP method. Furthermore, we introduce the use of the Blue-Green Development Strategy [14]-[15] to minimize the probability of error in product development by having two different versions of the systems developed simultaneously. With this strategy, the other version can act as a backup if something goes wrong in developing the systems.

II. STUDY SIGNIFICANCE

This chapter describes the method used in this research. This chapter consists of architecture, flow diagrams, and research procedures in developing a website and android-based SIAKAD application to support ecosystem education in Balikpapan schools. The following is the application architecture in Figure 1.

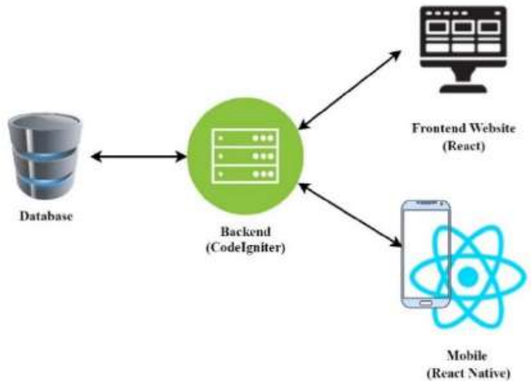


Figure 1. Siakad Application Architecture

Figure 1 describes the application architecture created. The application flow starts when the user accesses the website or mobile, and the user will give orders to the website or mobile system to do something. Then, the website or mobile system will send the command to the backend. The backend is a part of the system that will process incoming data and commands. After the command sent has been received by the backend, if the command given does not require data from the database, the backend will respond to the system that gave the command. If the given command requires data from the database, the backend will retrieve data from the database, process the data again, and respond to the command system.

Figure 2 describes the proposed method for developing a website and android-based SIAKAD application to support ecosystem education in Balikpapan schools.

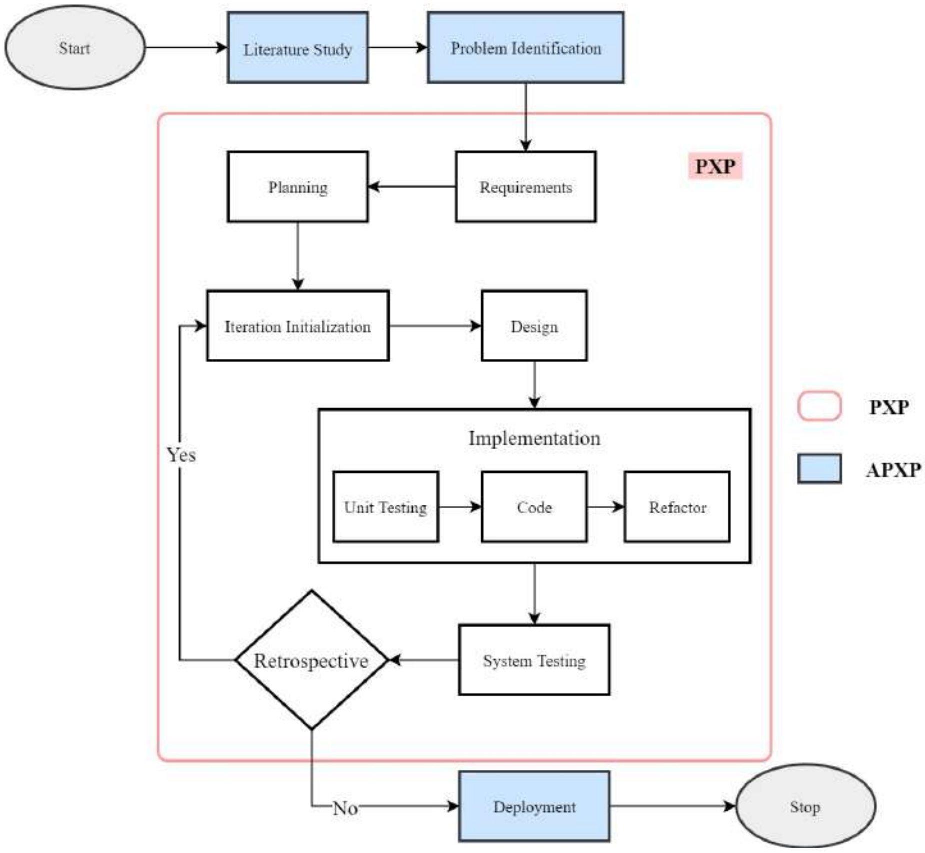


Figure 2. Flowchart Research of APXP Method

1. *Literature Study*

We collect literature related to the development of the website and android-based SIAKAD application. Then carried out the collection of literature on the development of systems similar to SIAKAD. Then, collect literature on the methods used in developing the website and android-based SIAKAD application.

2. *Problem Identification*

In this section, we collect information related to problems that occur in the object of research, namely the Al-Azhar 58 School Balikpapan. We interview the school, namely the headmaster. Based on interviews, there are problems in manual academic information systems using Microsoft word and excel applications. An analysis of the educational system process is carried out manually and saw the actual implementation conditions from this problem.

3. *Requirements*

Analysis of system requirements is carried out by conducting interviews with parties involved in the educational information system process. From the results of interviews conducted, it can be concluded that user needs will be developed into notifications on mobile to document user needs that must exist in the application.

4. *Planning*

This stage aims to provide an estimated time for working on all the requirements obtained at the requirements stage. Then, we determine the priority needs that must be done first and divide them into each iteration.

5. *Iteration Initialization*

Use case diagrams to show the relationship between the actor and each feature in the application. These use case diagrams describe the iteration process for the SIAKAD application.

6. *Design*

We perform designing functions, classes, and models to develop the SIAKAD application in the design section. The features will design structure and database relationships for application development and appearance. At this stage, It will produce class diagrams and application interface designs.

7. *Implementation*

We implement the previous design. At this stage, the process carried out is to implement the display that will be created, then provide functions into the display and integrate the functions with the database. The display and database will become a single unit.

8. *System Testing*

Tests of Admin, teacher, and student will use SIAKAD application testing. The implemented functions have been running as desired, or there are still problems or errors. Furthermore, the previous implementation is by design made.

9. *Retrospective*

This chapter summarizes the data iterations and considers the SIAKAD system that needs to be improved. Then, compare the results of the work with the notes at the iteration initialization stage. If this note still contains deficiencies in the system, then the iteration process will be carried out back to the iteration initialization stage to the retrospective stage again. According to the results, the iteration process will be completed if there are no deficiencies in the system.

10. *Deployment*

We will do a Blue-Green Deployment Strategy. Blue-Green Deployment is a technique for releasing applications predictably. This Deployment is a quick way to switch applications prioritized when in production and roll back them if an issue occurs.

III. RESULT AND DISCUSSION

The results and discussion section contains the results of the research and a thorough discussion of each result obtained from the research discussed. Give a detailed discussion of the results obtained therefore can answer the problems mentioned in the Introduction section.

1. *Literature Study, Problem Identification, and Requirements*

We will discuss with teachers at the Al-Azhar 58 school in Balikpapan to collect data and extract system requirements. During the discussion process, the teacher will explain the process of the academic system and the existing problems. Problems found include student recapitulation, student grades, and student financial payments. Based on these problems, we direct the teacher to describe all the requirements implemented into a list of features in building an academic information system.

2. *Planning*

At this stage, it aims to prepare the needs to proceed to the next stage. The goal is to determine story points that mean priority, provide an estimate of development time and the number of iterations used to complete the development of the education system. The higher the number, the more complex it is and the longer it takes to perform each iteration. Table 1 describes the story point values (1-5) of each defined feature.

TABLE I
STORY POINT OF ITERATION APPLICATION

No	Iteration Application	Story Point
1	Login (Feature 1)	3
2	Logout (Feature 1)	2
3	Edit Profile (Feature 2)	3
4	Add Student Data (Feature 2)	3
5	Edit Student Data (Feature 2)	3
6	Delete Student Data (Feature 2)	2
7	View Student Data (Feature 2)	1
8	Manage Student Grades (Feature 3)	5
9	Manage Student Payments (Feature 3)	5
10	View Student Grades (Feature 4)	3
11	View Student Payments (Feature 4)	3
	Total Story Point	**33**

In table 1, the number of story points is 33, divided by four features. The placement of features in each iteration must match the grouping of the iterations. If we iterate on the second feature without iterating on the first feature, the iteration will fail. For example, in table 1, the login and logout features are placed in the first iteration as a gateway to perform or access the next feature. Another example is the feature of adding student data in the second iteration is done first. Then the feature of managing student grades in the third iteration.

3. *Iteration Initialization*

The iteration initialization chapter will determine the use case diagram based on each feature to describe the system's functionality to be created, namely the SIAKAD application. Figure 3 is a use case diagram for the SIAKAD application.

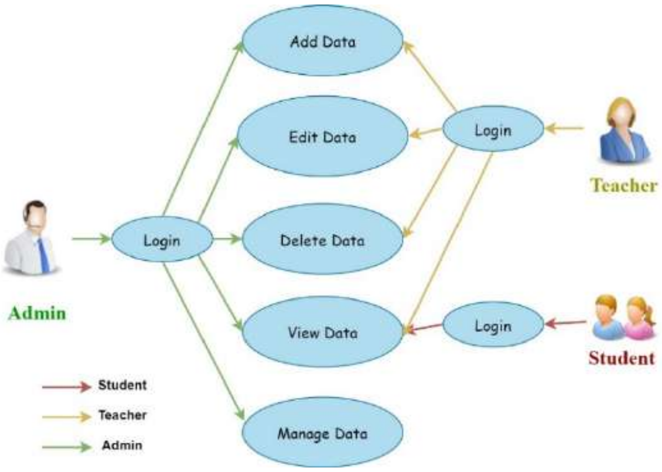


Figure 3. Use Case Diagram of SIAKAD Application

Figure 3 explains that there are three actors such as admin, teacher, and students. Despite this, before the system can determine the user type, the user must first log in. Once logged in, each type of user can access the usage associated with that type of user. First, admins can access, edit, add, delete, view, and manage data. Second, teachers can access data, add, edit, delete and view. Lastly, students can only access view data.

4. *Design*

After the use case diagram is defined in the previous step, then at the design stage, a class diagram is made on the database, as shown in Figure 4. The class diagram provides information about what classes, functions, and attributes need to be created in the development process, then describes Object-Oriented Programming (OOP) on the SIAKAD application.

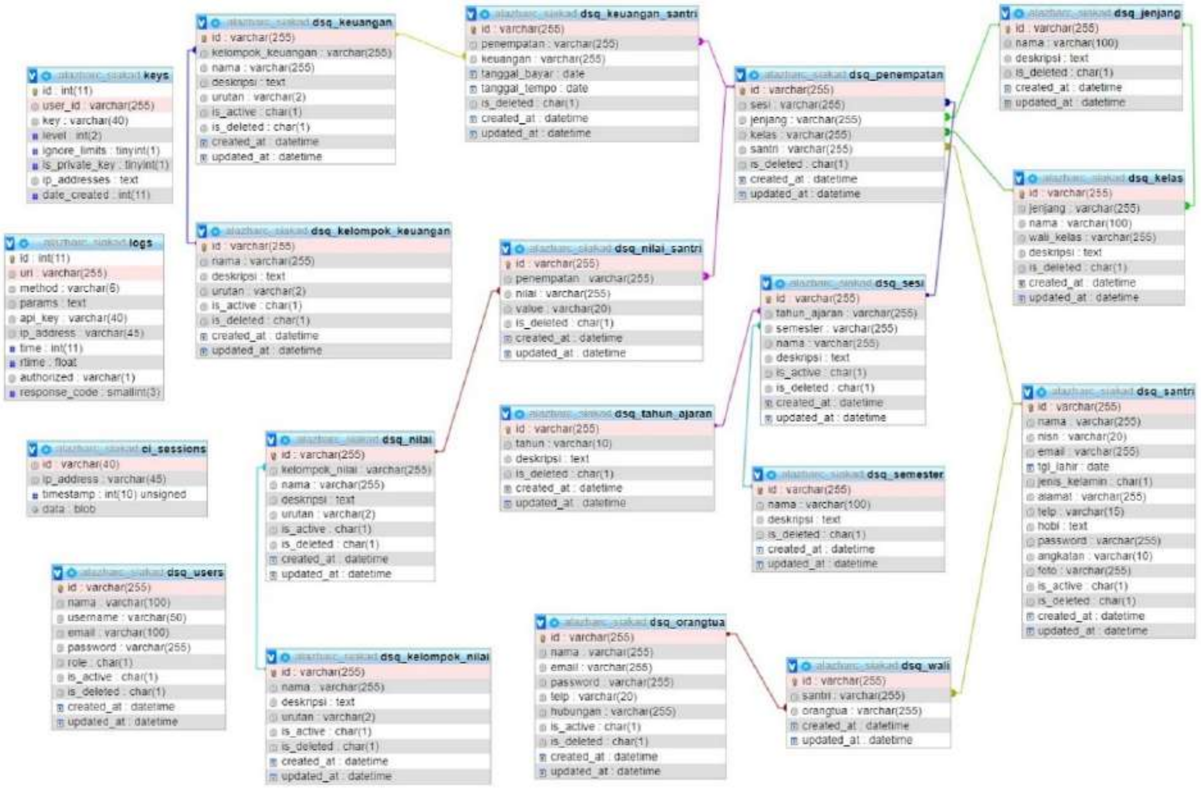


Figure 4. Class Diagram of Database SIAKAD

Figure 4 describes the class diagram in the SIAKAD application. In the class diagram, the student class serves as the parent class for the grades and finance classes. The child class also has all the attributes and functions in the student class. Then the grade class has a one-to-many relationship with the placement class because grades can create zero or more placement data. After that, the financial class has a one-to-many relationship with the placement class because one finance can have multiple transactions and one transaction.

Furthermore, if student data is deleted, then other data cannot be deleted as well. Similarly, the relationship between value class and financial class has a one-to-many relationship. If a student's data is deleted, the others cannot be deleted either. Then, there are class sessions, semesters, levels, classes, and parents who have a one-to-many relationship with the student class. The session class has a one-to-many relationship with the placement class with the relationship between the session and teacher classes. On the website platform, admins and teachers can access this system. On the mobile platform, students can only access this system.

5. *Implementation*

This implementation section explains the implementation results and explains of the functions of each feature in the SIAKAD application.

1) Feature 1

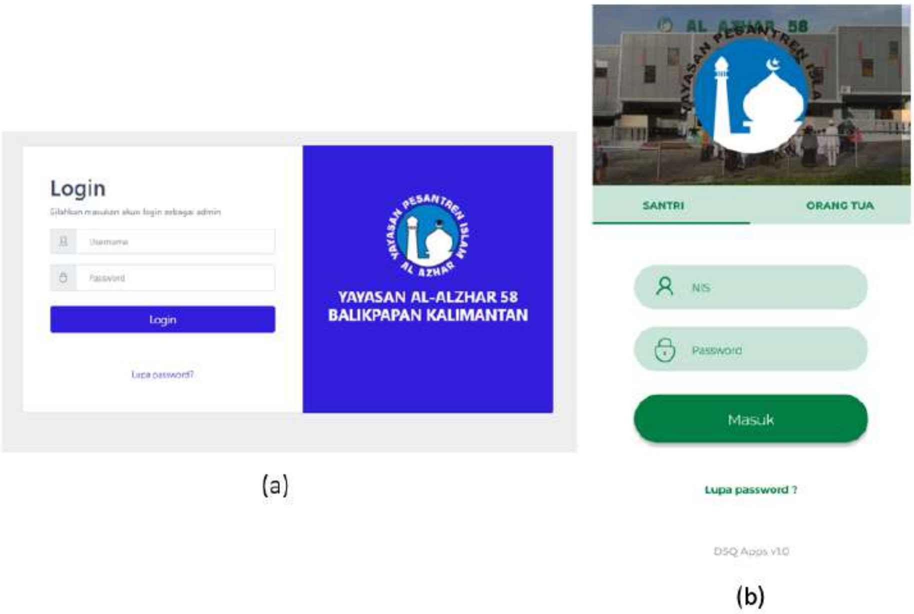
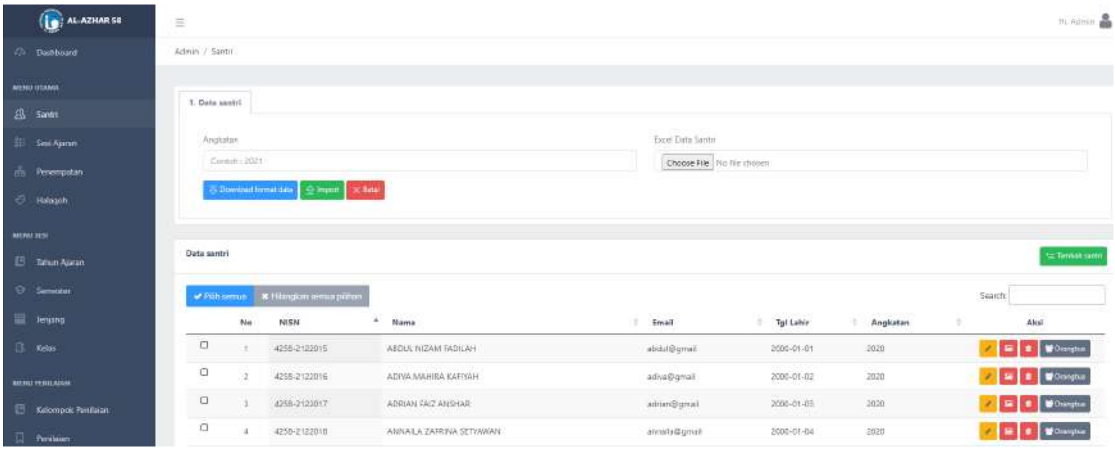


Figure 5. (a) Login Page in Website (b) Login Page in Android

The login page has different functions for web and mobile platforms. In figure 5(a), the login function will allow a user to log in when the user is an admin and a teacher. If the system detects that the user wants to log in as a student, then the system will refuse and be directed to log in to Android in Figure 5(b). In the mobile application, the login function only allows user login if the user is a student. If the user is an admin or teacher, the system will refuse and be directed to log in to the website in Figure 5(a). If the user data entered is not recognized by the system, the system will send an error message informing the user that the Username or NIS and password entered are incorrect. This login page solves the problem of filtering that can enter the application and can be done by the user.

2) Feature 2



Angkatan: Contoh : 2021

Excel Data Santri: Choose File No file chosen

Data santri

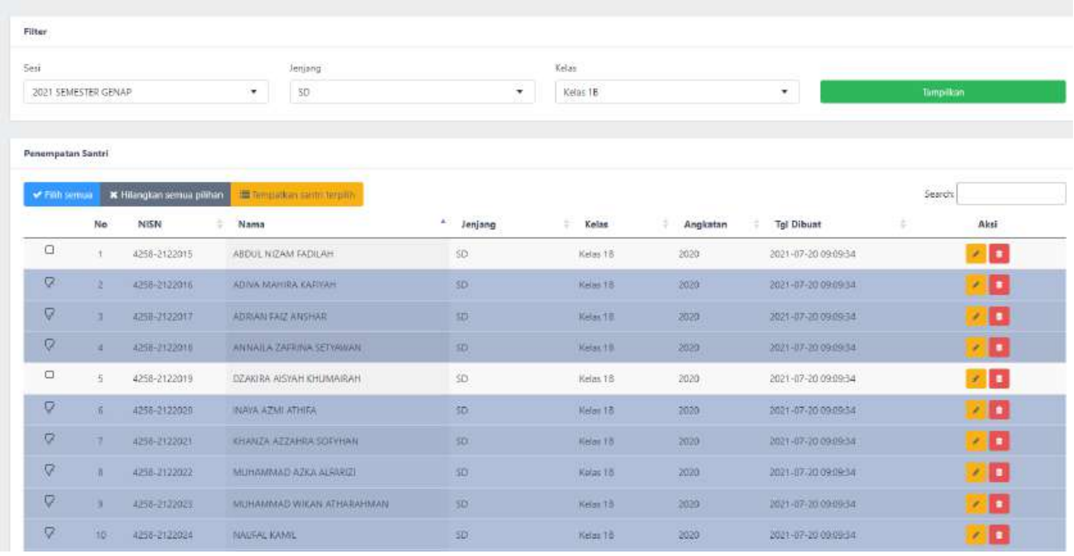
Search:

	No	NISN	Nama	Email	Tgl Lahir	Angkatan	Aksi
☐	1	4258-2122015	ABDUL NIZAM FADILAH	abdul@gmail	2000-01-01	2020	Orangtua
☐	2	4258-2122016	ADIVA MAHIRA KAFIYAH	adiva@gmail	2000-01-02	2020	Orangtua
☐	3	4258-2122017	ADRIAN FAIZ ANSHAR	adrian@gmail	2000-01-03	2020	Orangtua
☐	4	4258-2122018	ANNAILA ZAFRINA SETYAWAN	annaila@gmail	2000-01-04	2020	Orangtua

Figure 6. Feature of Add, Edit, View, and Delete Student Data in Website

Figure 6 can be accessed by admins and teachers on the website because users can add, edit, view, and delete student data features. Not only adding student data but also adding parental data for each student on the list. Then we added a student data import feature. Therefore, we do not add student data one by one.

3) Feature 3



Filter

Sesi: 2021 SEMESTER GENAP
Jenjang: SD
Kelas: Kelas 1B

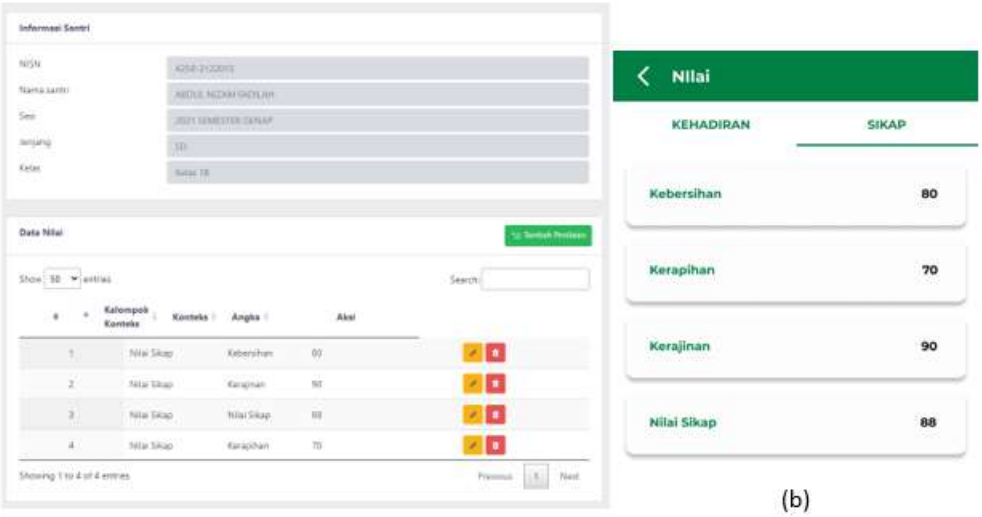
Penempatan Santri

	No	NISN	Nama	Jenjang	Kelas	Angkatan	Tgl Dibuat	Aksi
☐	1	4258-2122015	ABDUL NIZAM FADILAH	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	2	4258-2122016	ADIVA MAHIRA KAFIYAH	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	3	4258-2122017	ADRIAN FAIZ ANSHAR	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	4	4258-2122018	ANNAILA ZAFRINA SETYAWAN	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☐	5	4258-2122019	DZAKIRA AISYAH KHUMAIRAH	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	6	4258-2122020	INAYA AZMI ATHIFA	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	7	4258-2122021	KHANZA AZZAHRA SOFYHAN	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	8	4258-2122022	MUHAMMAD AZKA ALFARIZI	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	9	4258-2122023	MUHAMMAD WIKAN ATHARAHMAN	SD	Kelas 1B	2020	2021-07-20 09:09:34	
☑	10	4258-2122024	NAUFAL KAMIL	SD	Kelas 1B	2020	2021-07-20 09:09:34	

Figure 7. Manage Student Grades in Website

Figure 7 can increase the level of each class. The teacher can select a student if the student gets a low grade or does not advance to a grade. Figure 7 can also show several classes for each different session. One example is the semester of 2020/2021 academic year event. Then the teacher can choose a new school year with odd semesters from the choices of all students.

4) Feature 4



Informasi Santri

NISN: 4258-2122015
Nama santri: ABDUL NIZAM FADILAH
Sesi: 2021 SEMESTER GENAP
Jenjang: SD
Kelas: Kelas 1B

Data Nilai

#	Kelompok Konteks	Konteks	Angka	Aksi
1	Nilai Sikap	Kebersihan	80	
2	Nilai Sikap	Kerajinan	90	
3	Nilai Sikap	Nilai Sikap	88	
4	Nilai Sikap	Kerapihan	70	

Showing 1 to 4 of 4 entries

Nilai

Kebersihan	80
Kerapihan	70
Kerajinan	90
Nilai Sikap	88

(a)

(b)

Figure 8. (a) View Student Grades in Website (b) View Student Grades in Android

Figure 8 is an example of a display to see the assessment of each student with a different class. Not only the teacher sees the assessment, but also the teacher can give a different value in each assessment group. Figure 8 (a) displays the website where there is an assessment for students given by the teacher. Figure 8 (b) is a display from Android that contains student grade results that student can only access. In that section, the student can only see the student grades that the teacher has given.

6. System Testing

This stage will notify the progress of application work in iterations one to eleven to headmaster and teachers. The results of system testing for the eleven iterations can be seen in table 2.

TABLE 2
RESULT SYSTEM TESTING

No	Iteration Application	Expectation	Result	Valid
1	Login (Feature 1)	Login to the Homepage	Login to the Homepage	Yes
2	Logout (Feature 1)	Logout to the Login Page	Logout to the Login Page	Yes
3	Edit Profile (Feature 2)	Update Profile	Update Profile	Yes
4	Add Student Data (Feature 2)	Add Student Data	Add Student Data	Yes
5	Edit Student Data (Feature 2)	Update Student Data	Update Student Data	Yes
6	Delete Student Data (Feature 2)	Delete Student Data	Delete Student Data	Yes
7	View Student Data (Feature 2)	View Student Data	View Student Data	Yes
8	Manage Student Grades (Feature 3)	Set Student Grades	Set Student Grades	Yes
9	Manage Student Payments (Feature 3)	Set Student Payments	Set Student Payments	Yes
10	View Student Grades (Feature 4)	View Student Grades	View Student Grades	Yes
11	View Student Payments (Feature 4)	View Student Grades	View Student Grades	Yes

In Table 2, every iteration of login, logout, edit profile, add student, edit student, delete student, view student, manage student grades, manage student payments, view student grades and payments show Valid 'Yes.' means running according to expectations.

7. Retrospective

In the retrospective stages of iterations 1 to 11, we carried out the development process faster than the estimated time set by the school. We also encountered obstacles, namely the customization of the login system in the CodeIgniter framework and the process of displaying student assessments and finances on Android. The details of the obstacles are in feature one and feature four are shown in table 3.

TABLE 3
THE DETAIL OF THE OBSTACLES

No	Difficulty	Simplicity

1	Customize the CodeIgniter framework login system provided by default	Search for on the CodeIgniter framework program code and read the CodeIgniter documentation on customizing the login system
2	The process of view student assessments and finances on android	Merging existing databases on the website and android carefully and correctly

8. *Deployment*

After the iteration process is complete, the next step is to carry out the application deployment process. In this deployment process, we use the Blue-Green Deployment Strategy. As explained in the research methodology, it is a fast way to change applications prioritized when in production and roll back if an issue occurs.

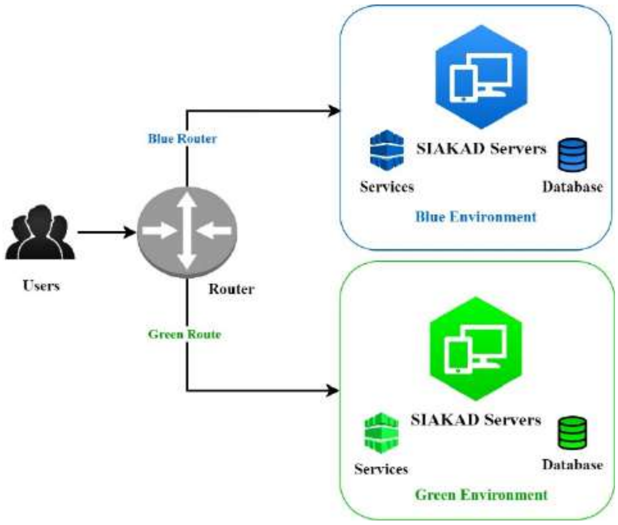


Figure 9. Blue-Green Deployment Strategy (Initial State)

Figure 9 explains that we have two identical environments (infrastructure). Blue and Green environment means hosting environments currently in production (app1 version1). When we want to upgrade app1 to version2, it will be done in the blue environment in figure 10. We will deploy the latest version (app1 v2) and other tests in that environment. When everything looks good, we can change our router point to a green environment.

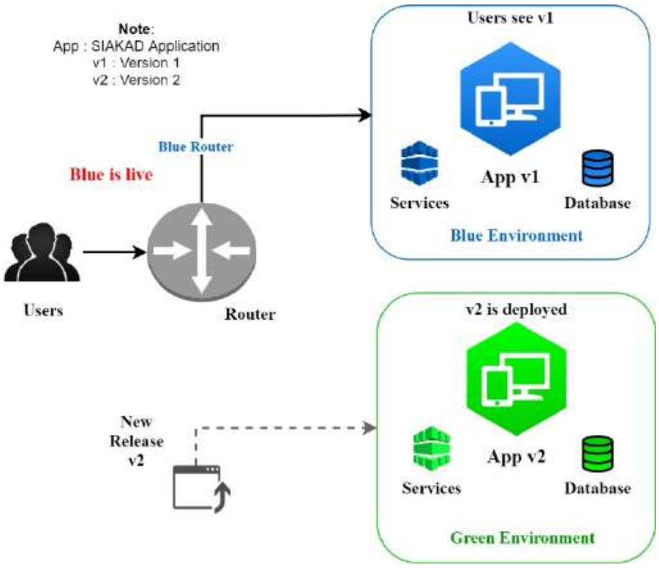


Figure 10. Blue-Green Deployment Strategy (Running State)

We can shut down the blue environment and reuse it for the latest releases in figure 11. If not, we can quickly roll back to a blue environment by changing the router point back. The deployment process can change very quickly once the switch service changes, and the deployment process has almost no issues with different versions of the service.

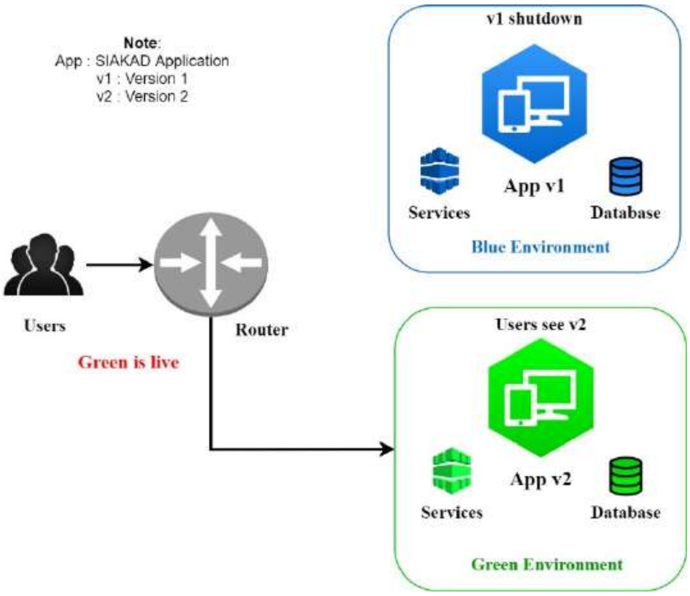


Figure 11. Blue-Green Deployment Strategy (Final State)

IV. CONCLUSION

This conclusion is an academic information system application (SIAKAD) based on Website and Android, which is made using the Advanced Personal Extreme Programming (APXP) method. We can see that the PXP method does not use the deployment process after the retrospective process. Therefore, one day it will become a problem because it can make the application unable to run. In the APXP method, we add Deployment methods such as Blue-Green Deployment Strategy. This deployment process can change very quickly once the switch service changes immediately. Moreover, the deployment process has almost no issues with different versions of the service. Each process from 1-11 iterations (login, logout, edit profile, add, edit, delete, view, manage grades, manage payments, view grades, and payments) or four features produce valid results in the system testing stage.

REFERENCES

[1] H.D. Benington, “Production of large computer programs,” *ONR Symposium on Advanced Programming Methods for Digital Computers*. 1956; pp. 15-27.

[2] W. W. Royce, “Managing the development of large software systems,” *IEEE Wescon*. 1970; pp 1-9,

[3] N. B. Ruparelia, “Software development lifecycle models,” *SIGSOFT Softw. Eng*. Notes (35:3). 2010; pp. 8-13,

[4] T. Dyba and T. Dingsoyr, "What Do We Know about Agile Software Development?" *IEEE Software* (26:5). 2009; pp. 6-9,

[5] R. P. Pawar, “A comparative study of agile software development methodology and traditional waterfall model,” *IOSR Journal of Computer Engineering* (2:2). 2015; pp. 1-

8.

[6] I. G. Marthasari, W. Suharso and F. A. Ardiansyah, “Personal Extreme Programming with MoSCoW Prioritization for Developing Library Information System,” *Proceeding of the Electrical Engineering Computer Science and Informatics*, *5*(1). 2018; pp.537-541.

[7] R. Agarwal and D. Umphress, “Extreme programming for a single person team,” *Proceedings of the 46th Annual Southeast Regional Conference on XX*. 2008; (pp. 82-87).

[8] Y. Dzhurov, I. Krasteva and S. Ilieva, “Personal Extreme Programming–An Agile Process for Autonomous Developers.”, 2009.

[9] S. A. Asri, I.G.A.M. Sunaya, E. Rudiastari and W. Setiawan, “Web Based Information System for Job Training Activities Using Personal Extreme Programming (PXP),” *Journal of Physics: Conference Series* (953:1), 2018.

[10] F. Gobbo and M. Vaccari, “The pomodoro technique for sustainable pace in extreme programming teams.” *International Conference on Agile Processes and Extreme Programming in Software Engineering*. 2008; pp. 180-184.

[11] A. Buzachis, A. Galletta, A. Celesti, L. Carnevale and M. Villari, “Towards osmotic computing: a blue-green strategy for the fast re-deployment of microservices.” *IEEE Symposium on Computers and Communications*. 2019; pp. 1-6.

[12] A. Herdiansah, “System Development for Learning Process Monitoring in Private Lesson Institution Using Codeigniter Framework.” *JISA*. (4:1), 2021.

[13] T. Tamrin and S. Ma’arif, “Android BAse Rapid Application Development for Learning Yanbu’a.” *Journal of Applied Intelligent System* (5:2). 2020; pp.91-97.

[14] A. Buzachis, A. Galletta, A. Celesti, L. Carnevale and M. Villari, "Towards osmotic computing: a blue-green strategy for the fast re-deployment of microservices." *2019 IEEE Symposium on Computers and Communications (ISCC). IEEE*, 2019.

[15] B. Yang, A. Sailer, S. Jain, A. E. Tomala-Reyes, M. Singh and A. Ramnath, "Service discovery based blue-green deployment technique in cloud native environments." *IEEE International Conference on Services Computing (SCC). IEEE*, 2018.