Jurnal Computech & Bisnis, Vol. 17, No. 1, June 2023, 9-19
ISSN (print): 1978-9629, ISSN (online): 2442-4943

Website Based Student Absence Application Design Using Extreme Programming Method (Study at A University in Bandung)

Mochamad Jejen Jaelani[1], Soecipto[2], Yenni Fatman[3], Siti Nur[4]
Universitas Islam Nusantara, Bandung[1,2,3,4]
jejenmochamad@gmail.com[1], cipto.bdg@gmail.com[2], yennifatman@gmail.com[3], sitinur.uninus@gmail.com[4]

Abstract

Absenteeism application is one of the systems where attendance is critical data in every activity, be it lecture activities. Nevertheless, often, the data - this absence of data is less noticed. The proof is that many educational institutions still use conventional absenteeism. Conditions like this are certainly very vulnerable to fraud. Therefore, the Informatics Engineering Study Program of the Faculty of Engineering needs a system that can create an absence process that can produce information more efficiently and by its standards, of course, using methods that can solve the problems in the Informatics Engineering Study Program of the Faculty of Engineering—using the Extreme Programming Method. The extreme Programming method was chosen because the software to be created is simple and classified as small-scale software, and development takes little time. This system development method is one of the agile methods. Which has several stages, namely: 1. Planning, 2. Design, 3. Coding, and 4. Testing. Of course, with some design processes that use UML, such as Use Case Diagram, Activity Diagram, Flowchart &, and DFD, the student attendance website application built has been tested for eligibility using Blackbox testing. The results of this study create a system that can help informatics engineering programs to help in terms of student retirement. Design a system that can produce reports that suit needs effectively and efficiently.

Keywords: *Student Absenteeism, Extreme Programming Methods, UML, Blackbox*

INTRODUCTION

The swift advancement of information technology in contemporary times has significantly influenced the efficacy and proficiency of all undertakings. Computing technology plays a pivotal role in the continuous progress of various fields. (Firliana & Rhohman, 2019) In contemporary times, items once considered luxurious or scarce have become ubiquitous and essential, especially in governmental and private establishments. The employment of computers has progressed into the swiftest method of communication and learning.

Technology integration in education has become widespread, including at Universitas Islam Nusantara. Most educational institutions have adopted computer technology for various academic purposes, such as processing student data and recording inventory, including attendance and other academic activities. Efficient compilation of student data is crucial for recapitulation.

Being present in a given activity is essential in diverse contexts, including academic and professional environments. This condition is because attendance data plays a critical role in various aspects. Nevertheless, attendance data should be more frequently noticed. Empirical data indicates that a significant number of educational establishments continue to utilize traditional methods of monitoring attendance.

These circumstances are susceptible to academic dishonesty since attendance holds a significant role in the overall management of the course. Attending classes holds significant importance as it encompasses crucial data, including students' attendance rates, which can influence their academic performance, thereby raising concerns regarding the precision of the resultant data. (Colclasure et al., 2020; Zhang et al., 2021)

An approach to tackle this matter involves digitizing student attendance and catering to students and lecturers at a university in Bandung. This initiative targets the Faculty of Engineering, Department of Informatics. The attendance procedure remains traditional, whereby students provide their signatures on attendance sheets corresponding to their ongoing courses. (Yusuf et al., 2016) The prescribed protocol for commencing a class entails the act of recording the presence of students and the retrieval of pertinent documentation. The manual attendance procedures also undergo rotation during class sessions. (Ariyanti et al., 2020)

At a university in Bandung, there is a prevalent occurrence of cheating in attendance records. Consequently, the Department of Informatics at a university in Bandung necessitates a system that can optimize the attendance process, leading to enhanced efficiency and uniformity of data, utilizing the Extreme Programming Methodology. (Setiawansyah et al., 2021; Shameem et al., 2020) In general terms, Extreme Programming (XP) is a methodology that facilitates swift system development and testing while upholding high-quality standards. (Hakak et al., 2019; Perkusich et al., 2020)

METHOD

Extreme Programming (XP) has been selected as the methodology for developing the Attendance system. (Jay et al., 2021; Melegati et al., 2019) The decision to adopt Extreme Programming as the software development methodology is based on the software's relatively low complexity and its classification as small-scale software, facilitating a more expedited development process. (Nurkholis et al., 2021; Zhou et al., 2021) As mentioned earlier, the approach to system development is classified as an agile methodology intended to facilitate the iterative and continuously evolving phases of business system development. (Han & Ghadimi, 2022; Ibrahem Ahmed Osman et al., 2021) The Extreme Programming methodology comprises a series of stages, which are as follows:

1. Planning

   The planning stage of software development entails collecting preliminary user requirements, commonly called user stories. This condition facilitates the developers' comprehension of the business context, system output requirements, and essential software features. The initial stage of software development involves identifying and analyzing application requirements. This condition involves identifying issues, creating user stories as the fundamental aspect of Extreme Programming (XP) planning, establishing software objectives, identifying users,

determining necessary information, and analyzing application requirements.

2. Design

   The present study employs UML models, including use case and activity diagrams, to illustrate the system design. The design functions as a visual or conceptual model of the system, aiding in streamlining the development process. The software development methodology known as Extreme Programming employs Class Responsibility Collaborator (CRC) cards to identify and organize the classes involved in the process. The application design process encompasses the development of introductory classes utilizing the Class Responsibility Collaborator (CRC) technique, the design of a graphical user interface (GUI), and the establishment of a database design.

3. Coding

   During the coding stage, the development process entails the direct implementation of the system design that was previously established. The activities encompassed in the project involve the development of the database and the execution of the user interface. The study's system development employs programming languages, namely PHP, HTML, and MySQL, with the support of XAMPP and the Laravel Framework.

4. Testing

   The testing phase is intended to assess the functionality of the developed website application and ascertain its proper operation. Iterative testing is conducted in order to detect possible errors or issues. The evaluation procedure involves assessing the web application software's design, functionality, and performance.

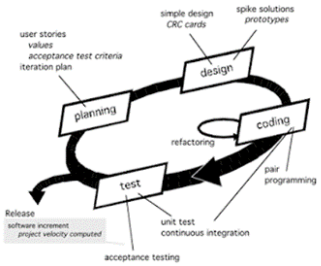

Figure 1. Depicts the Extreme Programming Method

The utilization of the Extreme Programming approach in software development presents various benefits, such as:

1. Enhanced customer contentment.
2. Expedited duration of development.
3. The topic of interest is the efficacy of client communication.
4. Decreased expenses associated with creating a product or service.
5. The methodology employed in this study is semi-formal.

Nonetheless, the Extreme Programming (XP) development approach has its drawbacks:

1. The principle of simplicity advocates for addressing immediate needs, which may result in an inability to create detailed code from the outset.
2. The development process relies primarily on the initial documentation provided by users, with minimal formal documentation.

RESULTS AND DISCUSSION

Attending is a means of evaluating the degree of work ethic and compliance with relevant

policies. The objective is to enhance the quality and service of an institution by improving discipline. The utilization of attendance serves as an indicator of discipline within a given setting and assesses the efficacy of its operational framework. Therefore, attending can also contribute to positively evaluating institutions that have implemented it.

Following the requirements analysis, the subsequent phase of system development entails the generation of system and software design. The system and software design phase marks the primary stage in the development of the Student Attendance System.

1. Context Diagram

   The Context Diagram provides a comprehensive overview of the system, depicting the system's inbound and outbound data flow.

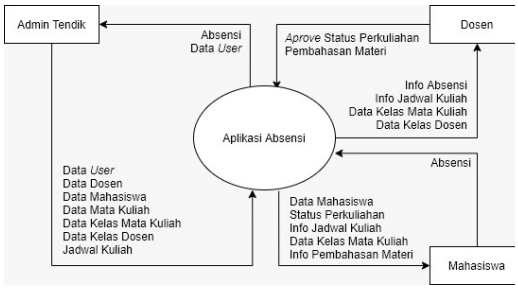

Figure 2. Context Diagram

2. Data Flow Diagram (DFD)

   Data Flow Diagrams are utilized to depict the progression of data and procedures within a system, as well as the entities implicated in it.

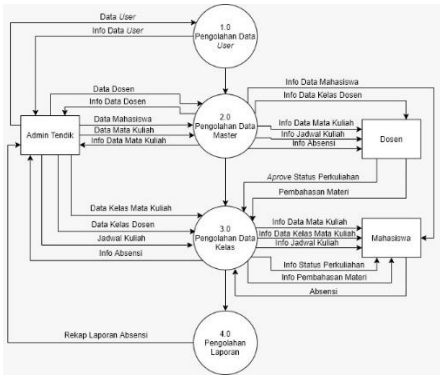

Figure 3. DFD Level 1

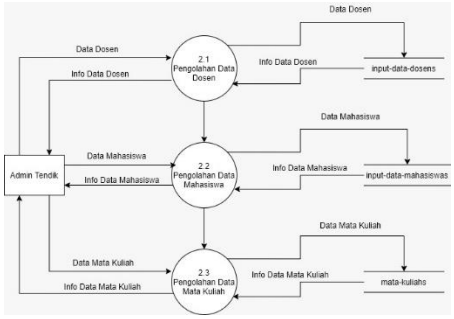

Figure 4. DFD Level 2

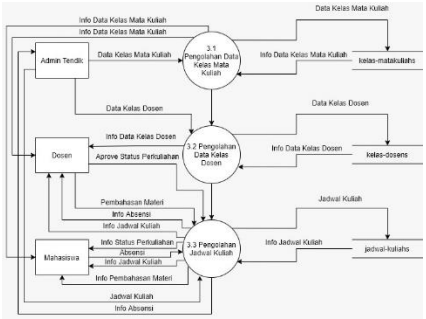

Figure 5. DFD Level 3

The design of this attendance system can improve the old system by providing a clear picture or view of the system design process from start to finish of the study.

1. Use Case Diagram

   Use Case Diagrams are used to model the interaction process based on the perspective of system users. Use Case This diagram consists of diagrams for use cases and actors.

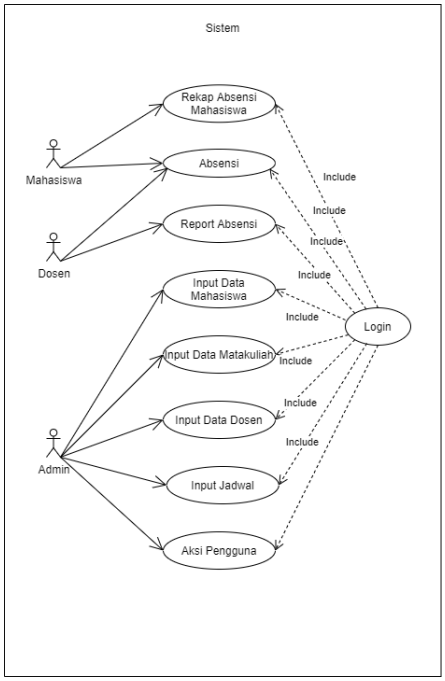

Figure 6. Use Case Diagram

2. Flowchart Diagram

   Flowchart diagrams that describe the sequence of one process with another process so that it is easy to understand and understand.

   a. Flowchart of Student Attendance Diagram

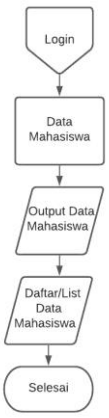

Figure 7. Flowchart of Student Attendance Diagram

   b. Lecturer Absenteeism Flowchart Diagram

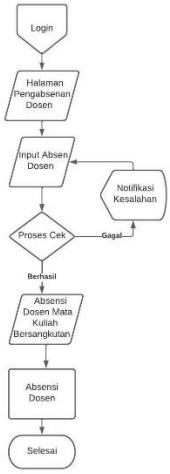

Figure 8. Flowchart of Lecturer Attendance Diagram

   c. Schedule Input Diagram Flowchart

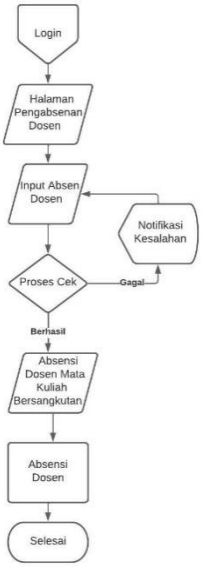

Figure 9. Schedule Input Diagram Flowchart

   d. Flowchart Diagram of Student Data Input

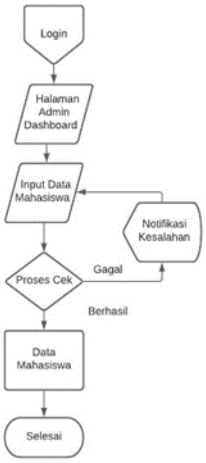

Figure 10. Flowchart of Student Data Input Diagram

e. Lecturer Data Input Flowchart Diagram

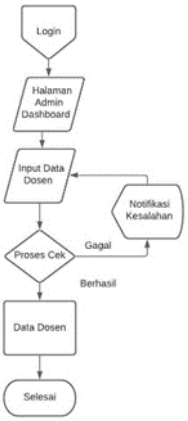

Figure 11. Lecturer Data Input Flowchart Diagram

f. Flowchart of Course Data Input Diagram

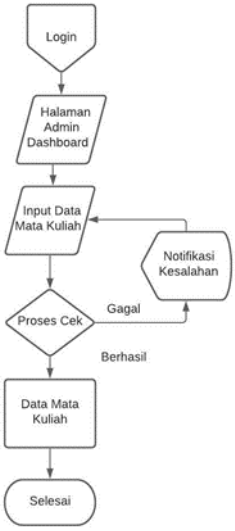

Figure 12. Course Data Input Flowchart Diagram

The coding implementation stage, also known as programming, is the subsequent phase of software development following system design activities. Once the system is prepared for operation, it comprises a description of the implementation of the database and the interface.

1. Database Implementation

   The attendance application utilized by the student population employs a MySQL database, which bears the nomenclature "absent online." The table structure, which has been devised, is executed on the MySQL server using the XAMPP web server utilizing the Apache network. Utilizing the programming language PHP version 7.0. The following table is present in the absence line database, encompassing:

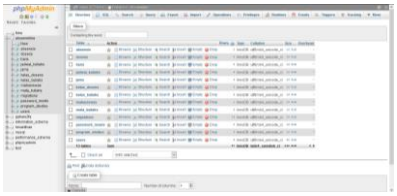

Figure 13. Database Implementation

2. Implementation of the Interface (Interface)

   Implementation of the interface (interface) is the tangible result of the user interface previously created at the design stage. In the implementation of this interface, the page designs that have been created through the program will be displayed.

   a. Login Page Implementation

Figure 14 Implementation of the Login Page

   b. Master Dashboard Admin Page Implementation

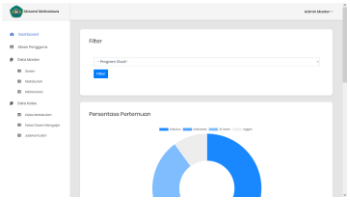

Figure 15 Implementation of the Master Dashboard Admin Page

The primary objective of testing this application is to perform Black Box Testing, which aims to evaluate the system's functional capabilities and analyze the outcomes of the application's execution. This testing aims to verify that the input will trigger the intended process and generate the expected output per the design specifications. The categorization of system testing will be based on user roles, specifically the Admin master (Tendik), Lecturers, and Students.

1. Login User Admin (Tendik)

Table 1. Login User Admin (Tendik)

No	Objective	Input	Expected output
1. Login and Reset Password section			
1	Know the response of the login display if the user username and password are correct	Usernames and passwords	Displays the dashboard page of the system
2	Know the response of the login display if the password is wrong and to change the password or reset	Passwords	Displays the display when changing the password or resetting
3	Know the respone when the email has not been registered	E-mail or users	View where the email error occurred
4	Know the response when you want to reset the password again	Reset Passwords	Display when resetting password and setting link to reset password to email page

No	Objective	Input	Expected output
1. Login and Reset Password section			
5	Know the response when you have entered the email intended to reset the password	Reset passwords	Display for reset password
6	Know the response when to change the password	Reset Passwords	Display to change the password
2. Lecturer Data Section			
7	Knowing the response when importing lecturer data	Import lecturer data	Display for importing lecturer data
8	Knowing the response when importing lecturer data is successful	Import lecturer data	Display when successfully adding lecturer data
9	Knowing the response when successfully editing lecturer data	Lecturer data	Displays information when the lecturer edits data successfully changes data
10	Knowing the response when editing lecturer data	Edit lecturer data	Displays data for editing lecturer data

2. Login User Lecturer

This function logs in for lecturers who have a system test scenario like the following:

Table 2. Login User Lecturer

No	Objective	Input	Expected output
1	Knowing the response of the lecture schedule display in the lecturer user section	Lecture schedule	Displays the class schedule section
2	Knowing the response of the display to change the status of the lecture where the red color indicates the lecture is closed and the green color indicates the lecture is opened by the lecturer concerned	Change schedule status	Displays information when changing class schedule status
3	Knowing the response when opening lecture classes	Open lecture	Displays a display where successfully opened lectures in the lecturer user section
4	Know the response when	Close lecture	Displays information when closing

No	Objective	Input	Expected output
	closing lectures		lectures where the lecturer will see students who are present and add discussion in lectures
5	Knowing the response when closing lectures	Close lecture	Displays information that the lecture is closed by the lecturer concerned
6	Knowing the respone when showing the attendance details	Absence details	Displays the attendance detail display in the lecturer user section

3. Login Student User

   The login function as a student has a system testing scenario as follows:

Table 3. Student User Login

No	Objective	Input	Expected output
1	Knowing the appearance of the student attendance section on the student user	Lecture schedule in student user	Displays the lecture schedule display of student data
2	Know the student attendance section	Student absences	Displays the view to do lecture attendance
3	Knowing the response when successful attendance	Student absences	Displays the display when successful lecture attendance
4	Know the respone of the detail attendance section	Absence details	Displays the attendance detail display when it is successful

CONCLUSION

Based on the conducted research, the following conclusions were drawn. The Extreme Programming methodology was effectively employed to develop a student attendance website, which involved a comprehensive process comprising four key stages: planning, design, coding, and testing. The feasibility of the Student Attendance Website application assesses via Blackbox testing methodology. The feasibility testing outcomes suggest that the application exhibits satisfactory performance, with all features of the student attendance website operating correctly. Hence, the website can serve as a platform for students to mark their attendance during lectures in the Informatics Engineering Program offered by the Faculty of Engineering. This system aims to aid the Informatics Engineering Program manage student attendance and devise a streamlined system that produces reports tailored to particular requirements.

REFERENCES

Ariyanti, L., Satria, M. N. D., & Alita, D. (2020). SISTEM INFORMASI AKADEMIK DAN ADMINISTRASI DENGAN METODE EXTREME PROGRAMMING PADA LEMBAGA KURSUS DAN PELATIHAN. *Jurnal Teknologi Dan Sistem Informasi*, *1*(1), 90–96. https://doi.org/10.33365/jtsi.v1i1.214

Colclasure, A. M., Tanim, T. R., Jansen, A. N., Trask, S. E., Dunlop, A. R., Polzin, B. J., Bloom, I., Robertson, D., Flores, L., Evans, M., Dufek, E. J., & Smith, K. (2020). Electrode scale and electrolyte transport effects on extreme fast charging of lithium-ion cells. *Electrochimica Acta*, *337*, 135854. https://doi.org/10.1016/j.electacta.2020.135854

Firliana, R., & Rhohman, F. (2019). Aplikasi Sistem Informasi Absensi Mahasiswa dan Dosen. *Journal of Computer and Information Technology*, *2*(2), 70–74.

Hakak, S., Noor, N. F. M., Ayub, M. N., Affal, H., Hussin, N., ahmed, E., & Imran, M. (2019). Cloud-assisted gamification for education and learning – Recent advances and challenges. *Computers & Electrical Engineering*, *74*, 22–34. https://doi.org/10.1016/j.compeleceng.2019.01.002

Han, E., & Ghadimi, N. (2022). Model identification of proton-exchange membrane fuel cells based on a hybrid convolutional neural network and extreme learning machine optimized by improved honey badger algorithm. *Sustainable Energy Technologies and Assessments*, *52*, 102005. https://doi.org/10.1016/j.seta.2022.102005

Ibrahem Ahmed Osman, A., Najah Ahmed, A., Chow, M. F., Feng Huang, Y., & El-Shafie, A. (2021). Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia. *Ain Shams Engineering Journal*, *12*(2), 1545–1556. https://doi.org/10.1016/j.asej.2020.11.011

Jay, O., Capon, A., Berry, P., Broderick, C., de Dear, R., Havenith, G., Honda, Y., Kovats, R. S., Ma, W., Malik, A., Morris, N. B., Nybo, L., Seneviratne, S. I., Vanos, J., & Ebi, K. L. (2021). Reducing the health effects of hot weather and heat extremes: from personal cooling strategies to green cities. *The Lancet*, *398*(10301), 709–724. https://doi.org/10.1016/S0140-6736(21)01209-5

Melegati, J., Goldman, A., Kon, F., & Wang, X. (2019). A model of requirements engineering in software startups. *Information and Software Technology*, *109*, 92–107. https://doi.org/10.1016/j.infsof.2019.02.001

Nurkholis, A., Susanto, E. R., & Wijaya, S. (2021). Penerapan Extreme Programming dalam Pengembangan Sistem Informasi Manajemen Pelayanan Publik. *J-SAKTI (Jurnal Sains Komputer & Informatika)*, *5*(1), 124–134.

Perkusich, M., Chaves e Silva, L., Costa, A., Ramos, F., Saraiva, R., Freire, A., Dilorenzo, E., Dantas, E., Santos, D.,

Gorgônio, K., Almeida, H., & Perkusich, A. (2020). Intelligent software engineering in the context of agile software development: A systematic literature review. *Information and Software Technology*, *119*, 106241. https://doi.org/10.1016/j.infsof.2019.106241

Setiawansyah, Sulistiani, H., Yuliani, A., & Hamidy, F. (2021). Perancangan Sistem Informasi Akuntansi Upah Lembur Karyawan Menggunakan Extreme Programming. *Technomedia Journal*, *6*(1). https://doi.org/10.33050/tmj.v6i1.1421

Shameem, M., Kumar, R. R., Nadeem, M., & Khan, A. A. (2020). Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. *Applied Soft Computing*, *90*, 106122. https://doi.org/10.1016/j.asoc.2020.106122

Yusuf, M., Ginardi, R. V. H., & Ahmadiyah, A. S. (2016). Rancang Bangun Aplikasi Absensi Perkuliahan
Mahasiswa dengan Pengenalan Wajah. *Jurnal Teknik ITS*, *5*(2). https://doi.org/10.12962/j23373539.v5i2.17518

Zhang, W., Wu, C., Zhong, H., Li, Y., & Wang, L. (2021). Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization. *Geoscience Frontiers*, *12*(1), 469–477. https://doi.org/10.1016/j.gsf.2020.03.007

Zhou, J., Qiu, Y., Khandelwal, M., Zhu, S., & Zhang, X. (2021). Developing a hybrid model of Jaya algorithm-based extreme gradient boosting machine to estimate blast-induced ground vibrations. *International Journal of Rock Mechanics and Mining Sciences*, *145*, 104856. https://doi.org/10.1016/j.ijrmms.2021.104856