MJI
Media Jurnal Informatika
Vol. 17, No. 1a, September 2025
p-issn : 2088-2114, e-issn : 2477-2542

Designing an Android-Based Employee Individual Assistant Application (Asika) (Case Study: At Madtive Studio)

Agie Nugraha Pratama [a,1,*], Mohamad Sidiq Permana [a,2], Wildan Muhammad [a,3]

[a]Putera Indonesia University, Jl. Doktor Muwardi Gang Perjuangan No.66 Muka Bypass, Kec. Cianjur, Cianjur 43215, Indonesia.
[1]agie.nugraha@unpi.ac.id*, [2] mohamadsidiqpermana@unpi-cianjur.ac.id, [3] wildanathoillah@gemail.com

ARTICLE INFO

Article history
Received 2025/08/09
Revised 2025/09/26
Accepted 2025/09/27

Keywords
HR Management
Software House
Extreme Programming
Agile Methodology

ABSTRACT

This research is based on the process and analysis conducted at one of the software houses in Cianjur district named Madtive Studio. Madtive Studio is a software house engaged in Information Technology, Systems and Branding which was established in 2015. The main focus of this company is the development of application systems for all types of companies ranging from retail, services, distributors and manufactures. This is done in order to become a benchmark for the author in carrying out research. The author found 3 factors as follows: (1) Input and reporting of data still use conventional methods. (2) Confusion when controlling work. (3) Difficulty in reporting work. For this reason, a HR management program is needed so that the ongoing procedures are well-systematic and will make it easier for owners and employees to input and report data. Making the system usually begins with an analysis process, but the process of gathering system requirements is not always found in the analysis process. In this study, the system design that the author uses is the extreme programming (XP) method. Method (XP) is needed to design a system that will be made by taking into account the needs and desires of users. In (XP) feedback from users becomes one of the important values. The result of the research is a system design with only one display, which is based on Android with a firebase database. This will make the user more simple but useful.

1. Introduction

The rapid advancement of technology in the present era, particularly within the field of Information Technology, has profoundly reshaped the landscape of daily human activities and professional workflows. This transformation is evident in how modern society integrates digital tools into routine tasks. However, despite this widespread adoption, there remains a notable gap in the full utilization of technology to streamline professional work and meet the specific needs of an organization's internal processes. Many businesses still rely on conventional, manual systems for tasks that could be efficiently managed through technological solutions, especially concerning data collection and reporting (Setiawan, P. R. 2020:63). This reliance on outdated methods often leads to inefficiencies, miscommunication, and difficulties in managerial oversight.

This study addresses this critical gap by focusing on the ineffectiveness and inefficiency of a conventional manual system currently in use at Madtive Studio. A thorough analysis of the company's workflow identified several key issues, including frequent miscommunication and a distinct lack of control over employee tasks from the perspective of the company owner. The primary challenges observed were the difficulty in monitoring work progress and the cumbersome process of generating accurate work reports. From a systemic perspective, these problems highlight the need for a unified platform that can effectively handle three core aspects of workforce management: data input, reporting, and control.

Furthermore, a significant issue observed at Madtive Studio is the traditional method of attendance tracking, which serves as a crucial component of employee performance evaluation and discipline scoring. The manual system is susceptible to fraud, most notably "attendance swapping," a common practice among employees seeking to meet minimum attendance requirements. This form of dishonesty is not only detrimental to the company and its fair-minded employees but also affects the individual engaging in the practice. The physical attendance sheets, vital for official work reports, are also prone to damage, which can disrupt the subsequent document verification and auditing processes (Setiawan, P. R. 2020:64). The research, therefore, is also motivated by the need to mitigate these risks and ensure the integrity of employee data.

In response to the problems outlined above, the objective of this research is to design and develop an automated, real-time, and user-friendly Android-based application platform. This application, named Employee Individual Assistant (ASIKA), aims to provide a comprehensive solution for managing human resources processes at Madtive Studio. The application will simplify work processes for employees and facilitate the generation of timely work reports for management, thereby marking a transition from conventional to digital practices. The methodology for this project will be executed in three distinct stages: first, gathering fundamental requirements from both the company and its employees; second, the development and implementation of the application based on the collected requirements; and third, rigorous testing and a trial period of the application within the workplace environment to ensure its functionality and effectiveness. This structured approach ensures that the final solution directly addresses the identified challenges and contributes to a more efficient and transparent workflow.

2. Method

The Employee Individual Assistant (ASIKA) application aims to serve as a supporting system at Madtive Studio, facilitating data input and task control between employees and HR management or the company owner. The design of this application requires a planning phase for needs analysis and scheduling to estimate the project's completion time.

The next stage after planning is design, where system analysis and design are determined. This is followed by the coding stage, which involves application programming and testing. The final stage is testing, which concerns the distribution of the application to users and gathering user feedback on the created application.

2.1 Type and Approach of Research

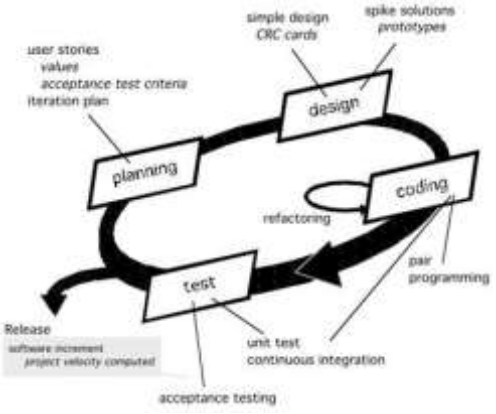


Fig. 1. Extreme Programming Method

This research uses the Extreme Programming (XP) method, known for its iterative and collaborative software development approach, to ensure a rapid response to changing needs and high product quality.

In applying this method, the initial step is the planning phase. This involves a needs analysis with stakeholders and determining a development schedule to more accurately estimate the project's completion time.

2.2 Object and Scope of Research

This research focuses on the design and development of an application named the Employee Individual Assistant Application (ASIKA). The application aims to be a supporting system at Madtive Studio, specifically to assist with data input and work control between employees and the Human Resources management or the company owner.

2.3 Data Collection Techniques

Data was collected through three main techniques: interviews, observation, and a literature review. Interviews were conducted with key individuals to understand the functional and non-functional requirements of the system. Observation involved directly observing the existing workflow at Madtive Studio, while a literature review was carried out to gather supporting information and theories relevant to the system's development.

2.4 Tools and Materials Used

2.4.1. Software:

a. Programming Language: Kotlin
b. Database: Google Firebase
c. Development Environment: Android Studio, JDK, and JRE
d. Libraries: Glide, OneSignal, Code Scanner, Image Cropper, Firebase Crashlytics, and Firebase Performance Monitoring.

2.4.2. Hardware:

a. Laptop: MacBook Pro (Retina, 13-inch, Early 2015)
b. Test Smartphone: Samsung A70

3. Result and Conclussion

3.1 Research Procedures or Stages

3.1.1. Planning

a. Functional Requirements

1) General Requirements

Users can register and log in to the system using their email address and password, and they can receive notifications from the application. Other general requirements for the application are that it must be real-time, easy to use, and automated.

2) Employee
   a) Can view information on the company, branches, positions, employees, including work contracts and supporting employee information.
   b) Can create tasks, categorize tasks, and report on task completion. c. Can perform online attendance, view attendance history, see attendance schedules, and view work schedules.
   c) Can submit online leave and sick leave requests, and also perform reimbursement for expenses.
3) Admin or Company Leader
   a) Can approve new users who have registered on the application.
   b) Can monitor all employee activities, such as viewing attendance history data and seeing all existing tasks.
   c) Can approve or reject leave and reimbursement requests.
   d) Can send notifications to all registered users in the application.

b.Non-Functional Requirements

1) User Friendliness

When using the Android system, users (employees and company leaders) can easily understand the available features because the system is designed with a simple and appealing interface.

2) Ease of Implementation

This system is developed with modern technology. The programming language used is Kotlin with a Firebase database, which means the application does not require a lot of time to build a back-end server system, as it is already covered by Firebase, which also minimizes costs.

c.Time Estimation

The estimated time for the Employee Individual Assistant (ASIKA) application is as follows:

Table 1. Time Estimation

No	Task	March				April				May				June			
		1	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*	*1*	*2*	*3*	*4*
1.	Data Collection																
	a. Observation																
	b. Literature Review																
	c. Interviews																
2.	Building a Prototype																
	a. Studying the Data																
	b. Building the Prototype																
3.	Prototype Evaluation																
	a. User Design Testing																
4.	Coding / System Encoding																
	a. Coding Phase Setup																
	b. Implementing the Application																
5.	System Testing																
	a. Application Testing																
	b. Design Testing																
6.	System Evaluation																

3.1.2. Design

The results of the previous functional requirements analysis were then designed using UML (Unified Modeling Language). The design developed in this writing is the first iteration, which is the alpha version.

a. Use Case

1) Use Case Diagram Employee

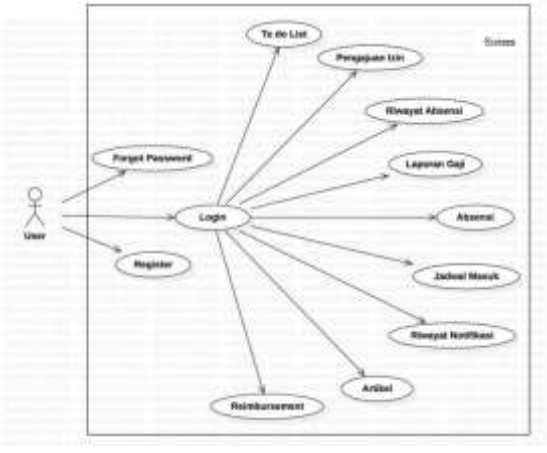


Fig. 2. Use Case Diagram Employee

2) Use Case Diagram Admin

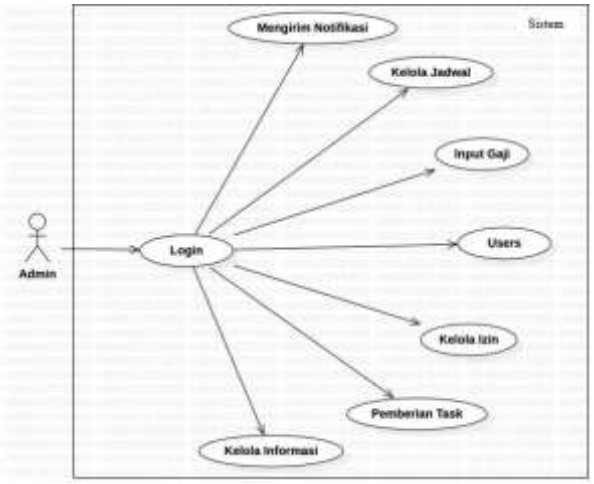


Fig. 3. Use Case Diagram Admin

b. Activity Diagram

1) Activity Diagram Register

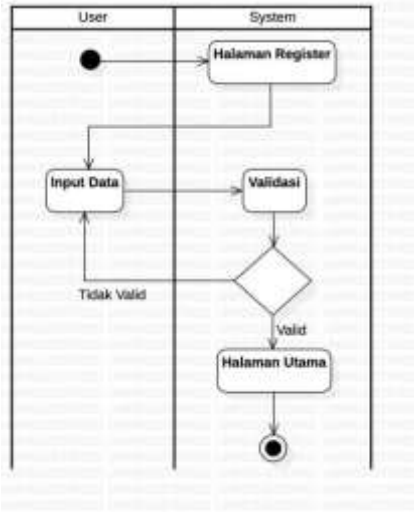


Fig. 4. Activity Diagram Register

2) Activity Diagram Login

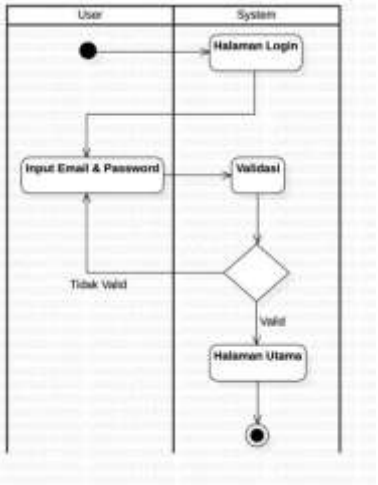


Fig. 5. Activity Diagram Login

Fig. 6.

3) Activity Diagram Forgot Password

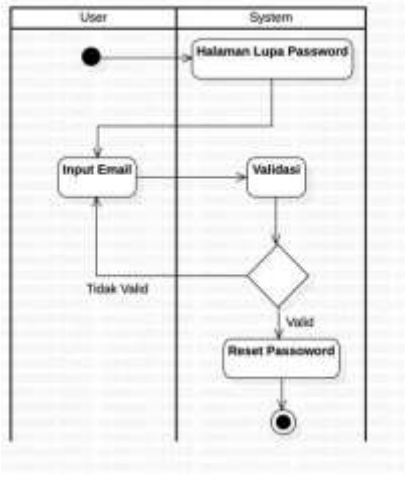


Fig. 7. Activity Diagram Forgot Password

4) Activity Diagram Scan QR Code

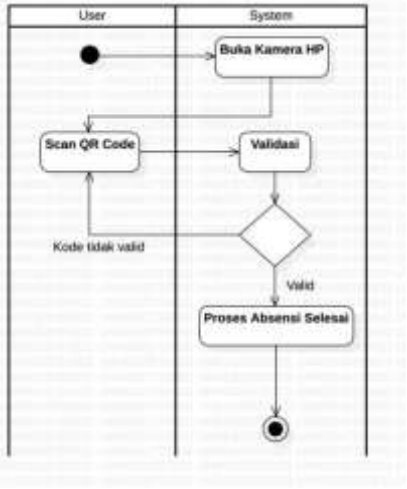


Fig. 8. Activity Diagram Scan QR Code

5) Activity Diagram Information List

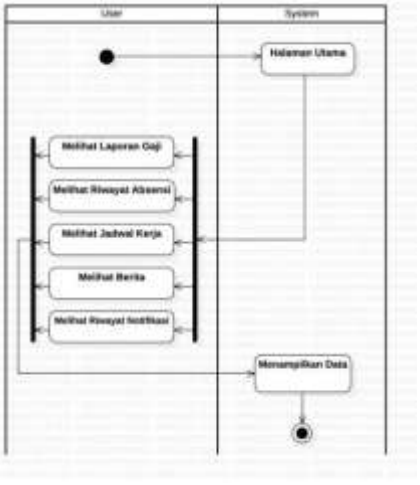


Fig. 9. Activity Diagram Information List

6) Activity Diagram Edit Profile

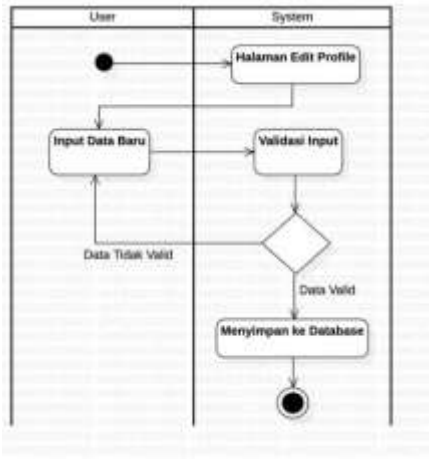


Fig. 10. Activity Diagram Edit Profil

7) Activity Diagram Leave and Permission Request

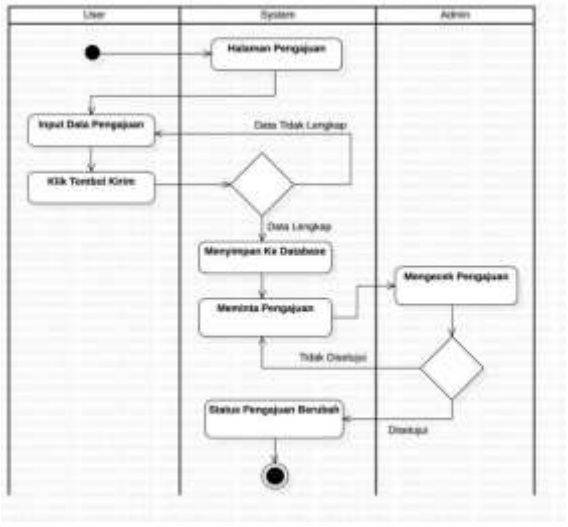


Fig. 11. Activity Diagram Leave and Permission Request

8) Activity Diagram To do List

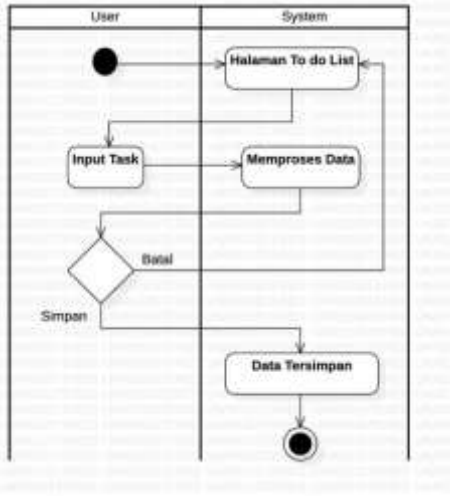
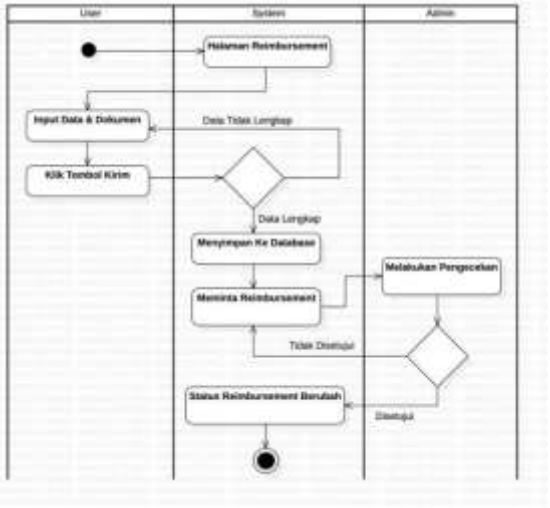


Fig. 12. Activity Diagram To do List



9) Activity Diagram Reimbursement

Fig. 13. Activity Diagram Reimbursement

10) Activity Diagram Approval User

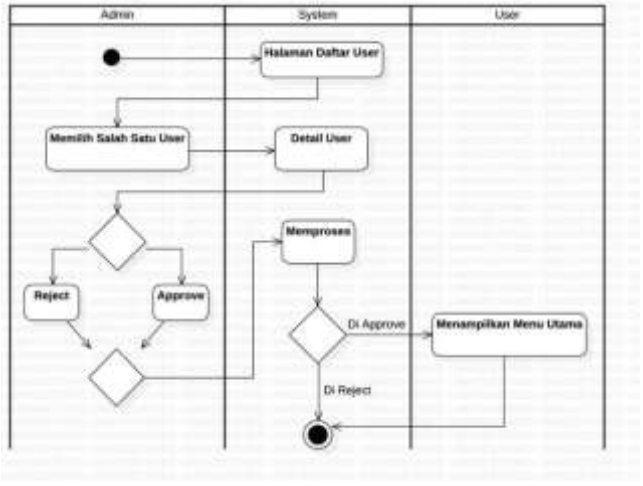


Fig. 14. Activity Diagram Approval User

11) Activity Diagram Employee Salary Input

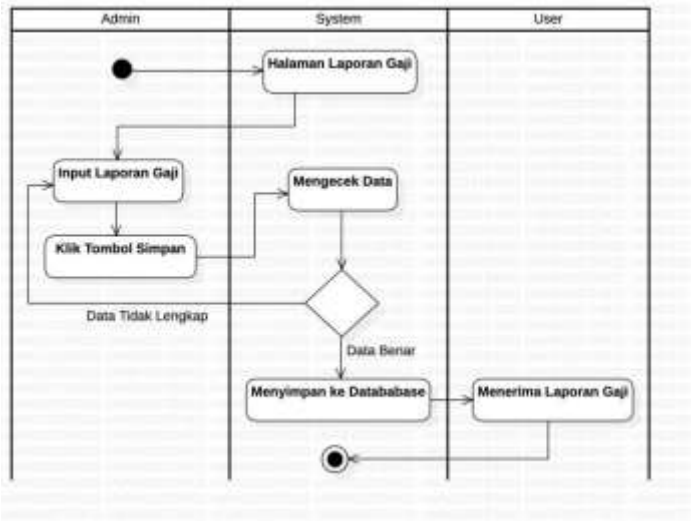


Fig. 15. Activity Diagram Employee Salary Input

c. Class DIagram

1) Class Diagram

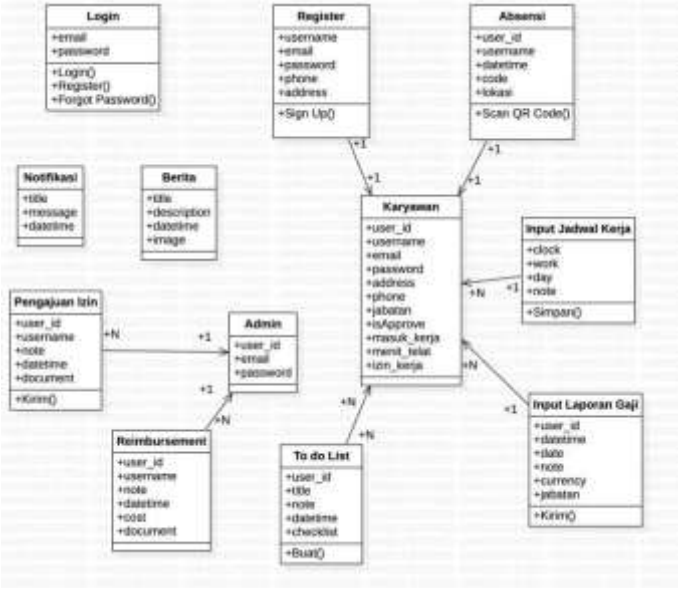


Fig. 16. Class Diagram

d. Sequence Diagram

1) Sequence Diagram Register

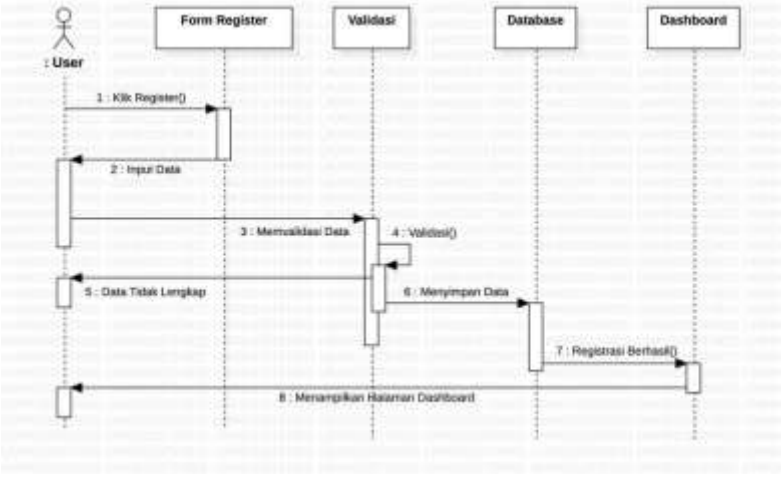


Fig. 17. Sequence Diagram Register

2) Sequence Diagram Login

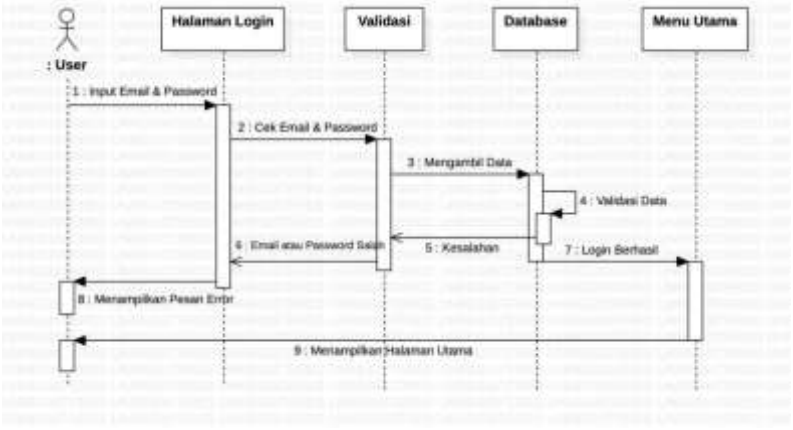


Fig. 18. Sequence Diagram Login

3) Sequence Diagram Forgot Password

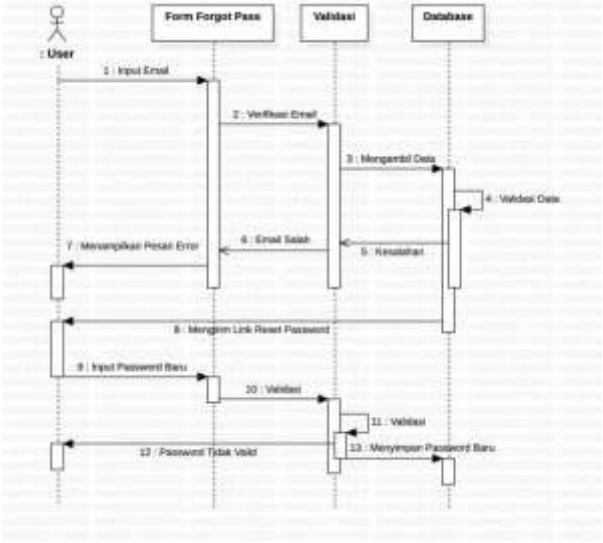


Fig. 19. Sequence Diagram Forgot Password

4) Sequence Diagram Scan QR Code

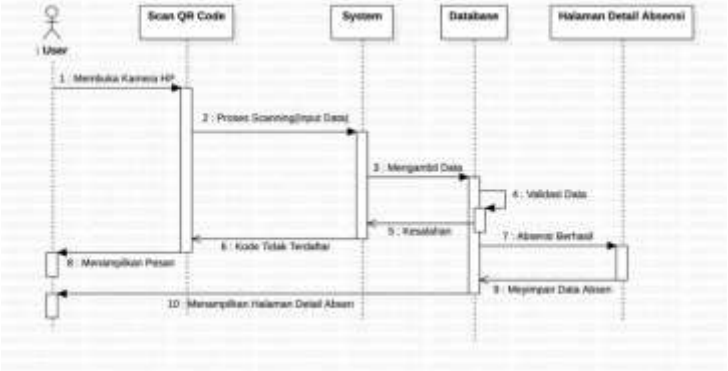


Fig. 20. Sequence Diagram Scan QR Code

5) Sequence Diagram Information List

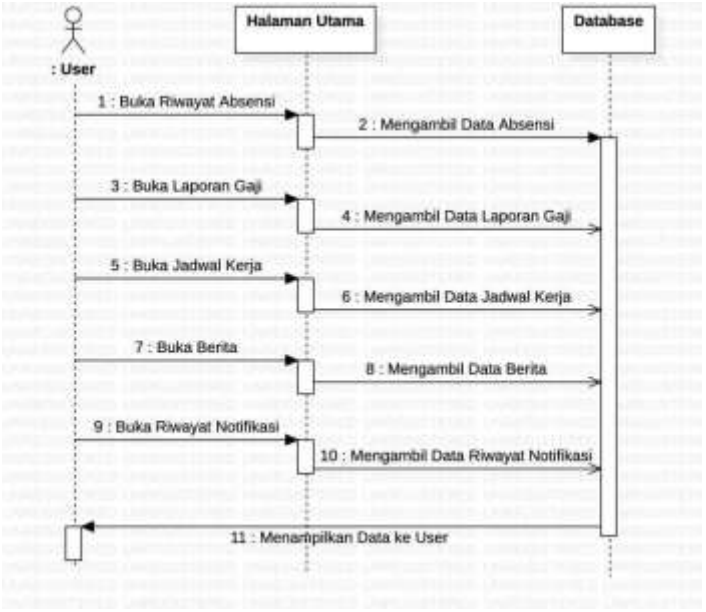


Fig. 21. Sequence Diagram Information List

6) Sequence Diagram Edit Profile

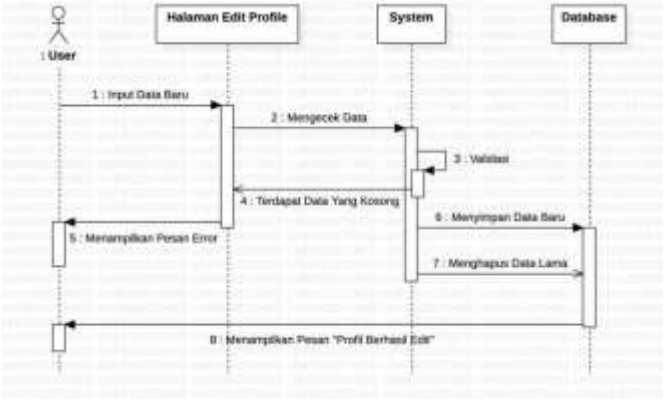


Fig. 22. Sequence Diagram Edit Profil

7) Sequence Diagram Leave and Permission Request

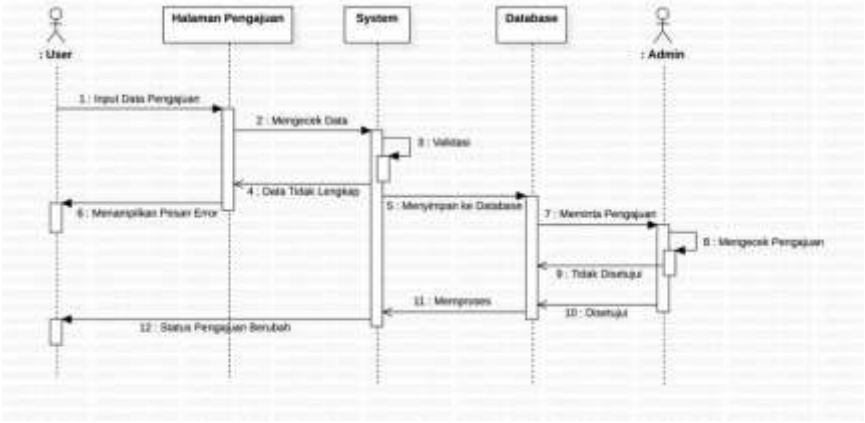


Fig. 23. Sequence Diagram Leave and Permission Request

8) Sequence Diagram To do List

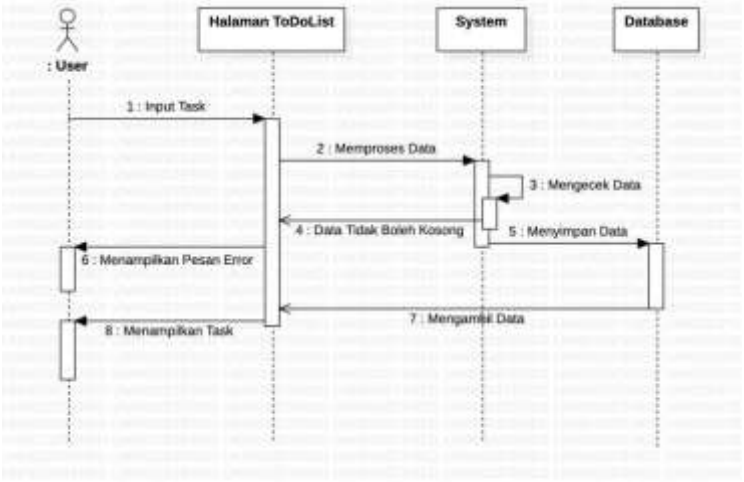


Fig. 24. Sequence Diagram To do List

9) Sequence Diagram Reimbursement

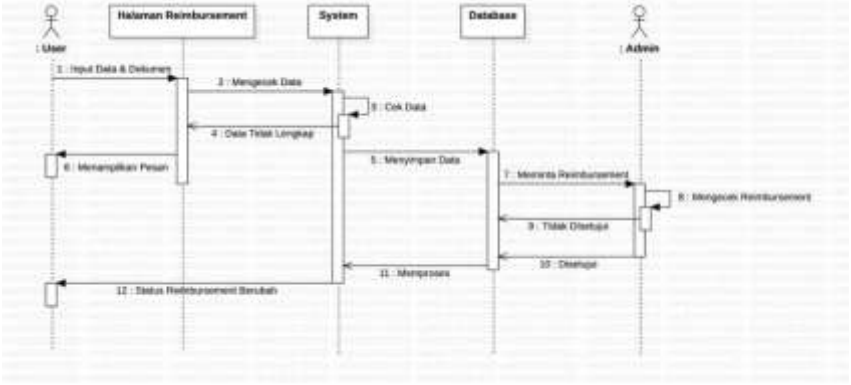


Fig. 25. Sequence Diagram Reimbursement

10) Sequence Diagram Register

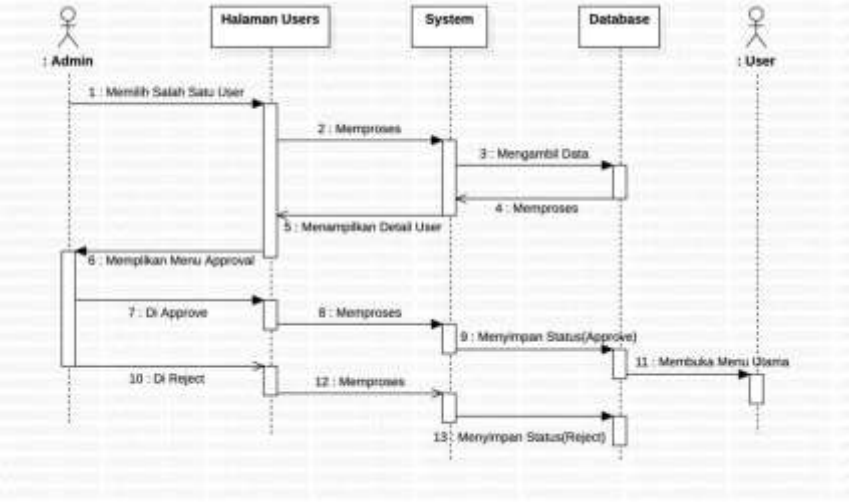


Fig. 26. Sequence Diagram Register

11) Sequence Diagram Approval User

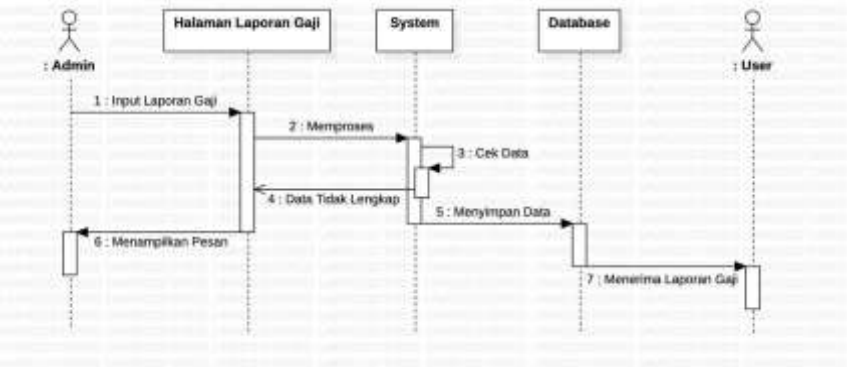


Fig. 27. Sequence Diagram Approval User

e. Database

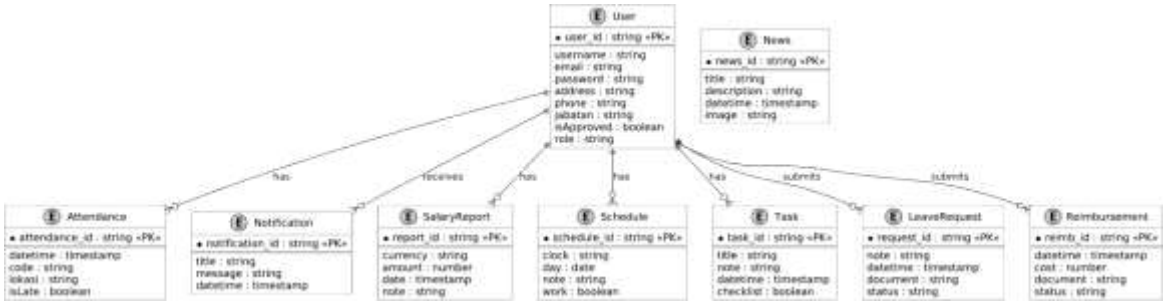


Fig. 28. Database ASIKA

f. Implementation

12) Application Display (For Employees)

a) Application Splash Screen

Fig. 29. Application Splash Screen

b) Initial Application View

Fig. 30. Initial Application View

c) Register / Sign Up Form

Fig. 31. Register / Sign Up Form

d) Login Form

Fig. 32. Login Form

e) Forgot Password Page

Fig. 33. Forgot Password Page

f) Main Application Page

Fig. 34. Main Application Page

g) QR Code Scan Form

Fig. 35. QR Code Scan Form

h) Display for Successful Attendance

Fig. 36. Display for Successful Attendance

i) Attendance History Form

Fig. 37. Attendance History Form

j) Work Schedule Form

Fig. 38. Work Schedule Form

k) Salary Report Form

Fig. 39. Salary Report Form

l) Notification History Form

Fig. 40. Notification History Form

m) Information List Form

Fig. 41. Information List Form

n) Profile Form

Fig. 42. Profile Form

o) Edit Profile Form

Fig. 43. Edit Profile Form

p) Leave and Permission Request Display

Fig. 44. Leave and Permission Request Display

q) Reimbursement Display

Fig. 45. Reimbursement Display

13) Application Display (For Admin)

a) Login Display

Fig. 46. Login Display

b) Main Menu

Fig. 47. Main Menu

c) Request List

Fig. 48. Request List

d) Salary Report Input

Fig. 49. Salary Report Input

e) Salary Report List

Fig. 50. Salary Report List

f) Employee Profile

Fig. 51. Employee Profile

14) Penggunaan OneSignal

a) First, open the OneSignal website at the link https://onesignal.com, then click the login button. You can log in using a OneSignal account or directly with your Gmail.

Fig. 52. OneSignal Display

b) Next, you will be taken to the "All Applications" page. Select the ASIKA application.

Fig. 53. OneSignal Initial Display

c) You will then be redirected to the "Messages" page. Here you will see a history of notifications that have been sent to clients. To create a new notification, click the "+New Push" button.

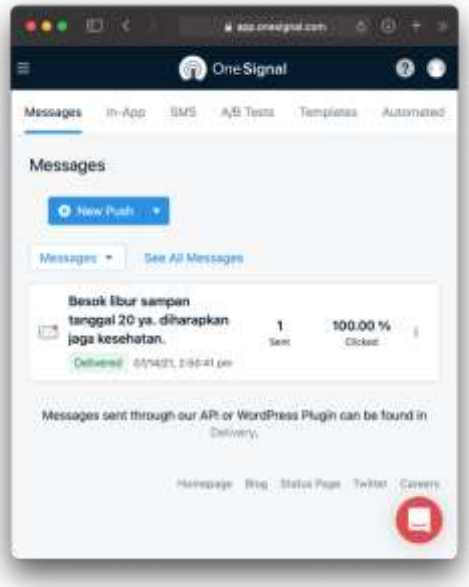


Fig. 54. OneSignal Message History

d) In the "Title" and "Message" sections, simply enter the title and content of the notification that will be displayed on the smartphone screen. Once filled out, just scroll down until you see the "Send" button.

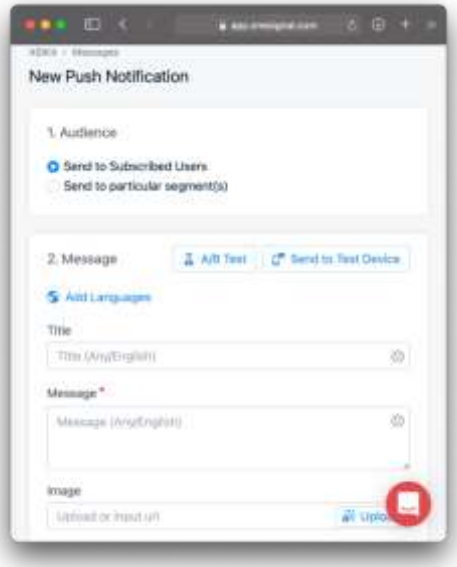


Fig. 55. Creating Notifications in OneSignal

e) Finally, just scroll down until you find the "Review and Send" button, then click that button until the sending process is complete.

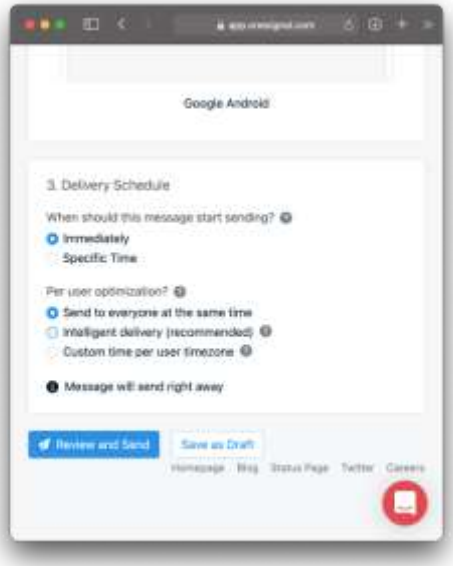


Fig. 56. Sending Notifications in OneSignal

f) The sending process will take time depending on the number of users of the ASIKA application. If successful, the notification will appear on the smartphone as shown below.

Fig. 57. Notification Successfully Sent

2.5.3. Coding

Coding is a stage of translating the application design into lines of program code, in accordance with the use case, class diagram, sequence diagram, tables, relationships between tables, and interface designs. In this application design research, the implementation uses the Kotlin programming language with Android Studio to create the ASIKA (Employee Individual Assistant) application. It also uses

Firebase as the data storage medium (database). In addition, the author used several additional components in designing this application, such as:

a. Glide A library for Android used to load images, videos, and animated GIFs.
b. OneSignal OneSignal is a service that enables push notifications, abstracting details such as the device's platform. With the OneSignal plugin, OutSystems applications can send and receive notifications.
c. Code Scanner An Android library used to scan barcodes such as QR Codes or Barcodes.
d. Image Cropper A library used to crop images to be displayed within the application.
e. Firebase Crashlytics Crashlytics is a lightweight, real-time error reporting solution that helps you track, prioritize, and fix stability issues that reduce application quality.
f. Firebase Performance Monitoring Performance Monitoring is a service that helps obtain reports on the performance characteristics of iOS, Android, and web applications.

2.5.4. Test

This application testing uses the Black Box method, which aims to determine whether the ASIKA application functions as it should according to its design. The following is the application testing table for checking the menus within the application.

Table 2. System Testing Table

Test Class	Test Scenario	Expected Result	Test Result
Splash Screen	Displaying the ASIKA App launcher icon for 2 seconds, then entering the main menu.	When the user opens the app, a splash screen appears before the login form. If the user has logged in before, they will be taken directly to the main menu.	Successful
Register	Input Name, Email, Password, Address, and Phone Number.	When the register button is clicked, the inputted data will be entered into the database. After successful registration, the user will be redirected to the main menu.	Successful
Register	Leaving one or all data fields empty, including email and password.	Displaying the message "Data cannot be empty".	Successful
Register	Filling the password form with less than 6 characters.	Displaying the message "Password is too short, minimum 6 characters".	Successful
Login	Input Email and Password.	When the user enters an email and password, a "failed" message appears if unsuccessful, and a "Login Successful" message appears if successful, followed by entry into the main menu.	Successful
Login	If the user enters an unregistered email and password.	Displaying the message "Email and password are not registered".	Successful
Forgot Password	Input Email Address.	If the entered email is correct, a link to reset the password will appear in the inbox.	Successful
Forgot Password	Entering an unregistered email.	Displaying the message "Email is not yet registered".	Successful
Scan QR Code	Scanning a QR Code for attendance.	When the phone camera is pointed at a QR Code image, the app will scan it and verify that the scanned code matches the one in the database.	Successful
Scan QR Code	Scanning a QR Code with the wrong code.	Displaying the message "Code not registered" and canceling the scan process.	Successful

Attendance History	Displaying the attendance history list.	After successful attendance using the QR Code Scan, the data will be saved in the database and displayed in the attendance history.	Successful
Notifikasi	Sending notifications from the admin to the client using OneSignal.	The notification title and content will appear in the bar of smartphones that have the ASIKA app installed.	Successful
Working Hours	The admin inputs Working Hours, consisting of hours and days, for each user.	Working Hours data will be saved on the Firebase server and displayed on the Working Hours page.	Successful
Salary Report	The admin inputs a salary report, consisting of amount, date, and description, for each user.	Salary Report data will be saved on the Firebase server and displayed on the Salary Report page.	Successful
Leave Request	Inputting data consisting of the request content and document files.	Request data will be saved to the Firebase server and sent to the admin for approval or rejection.	Successful
Reimbursement	Inputting fund reimbursement data, consisting of the amount and the content of the request.	The data is saved to the Firebase server and then sent to the admin for approval or rejection.	Successful
News Page	The admin inputs the news title and content.	News or article data is saved in Firebase and can be displayed within the application.	Successful
Edit Profile	Changing and re-entering the Name, Email, Password, Address, and Phone Number.	The data will be saved back into Firebase, and the edited data will then be displayed on the profile page.	Successful
Change Profile Picture	Inputting a photo file in JPG or PNG format.	The file will be saved in Firebase Storage, and the photo will then be displayed on the profile page.	Successful
To-do List	Inputting the task name, time, and date, and inviting members.	The created task will be saved in Firebase and then displayed on the To-do List page within the app.	Successful
Logout	Clicking the logout button on the profile page.	An authenticated user will be logged out of the app, and the existing data will remain saved in the database.	Successful
Approve User	The admin clicks the approve button for the selected user.	A user who has been approved by the admin will be able to access the dashboard page.	Successful

4. Conclusion

Based on the results of the Black Box testing conducted on the application, it can be concluded that the application is running quite optimally. However, it does not rule out the possibility of errors occurring while the application is in use. If such an error occurs, the researcher will make improvements so that the application can function as it should.

5. References

[1] A. S. Rosa and M. Shalahuddin, Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek. Bandung: Informatika, 2014, pp. 133-146.
[2] B. Hariyanto, Sistem Manajemen Basis Data. Bandung: Informatika, 2004, p. 105.
[3] S. Hermawan, Mudah Membuat Aplikasi Android. Yogyakarta: Andi Offset, 2011, p. 1.
[4] K. C. Laudon and J. P. Laudon, Sistem Informasi Manajemen Mengelola Perusahaan Digital. Yogyakarta: Andi Offset, 2005, p. 15.
[5] K. Kristanto, Perancangan Sistem Informasi. Jakarta: Rinda Hesti, 2015, p. 16.
[6] K. C. Laudon and J. P. Laudon, Management Information Systems - Managing The Digital Firm, 12th ed. Pearson Prentice Hall, 2012.

[7] R. McLeod, Sistem Informasi Manajemen Jilid I. Jakarta: Prenhallindo, 1998, p. 92.
[8] N. Nazaruddin, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android. Bandung: Penerbit Informatika, 2012, p. 1.
[9] S. Nidhra and J. Dondeti, "Black Box And White Box Testing Techniques –A Literature Review," International Journal of Embedded Systems and Applications (IJESA), vol. 2, no. 2, pp. 8-9, 2012.
[10] O'Brien, Pengantar Sistem Informasi. Jakarta: Rinda Hesti, 2016, p. 16.
[11] P. Pricilia, "Analisis Produktivitas Kerja Pada PT. BERKAT ANUGRAH RAYA," AGORA, vol. 5, no. 1, pp. 1-6, 2017.
[12] R. Syahrial, "Rancang Bangun Sistem Informasi Zakat Infaq Shodaqoh Menggunakan Metodologi Extreme Programming," 2018, pp. 32-33.
[13] P. R. Setiawan, "Aplikasi Absensi Online Berbasis Android," IT Journal Research and Development, vol. 5, no. 1, pp. 63-71, 2020.
[14] T. Arifianto, Membuat Interface Aplikasi Android Lebih Keren dengan LWIT. Yogyakarta: Andi, 2011, p. 1.
[15] Teknologi.id, "Software House: Definisi, Layanan, dan Struktur Pekerjaan," https://teknologi.id/insight/software-house-definisi-layanan-dan-struktur-pekerjaan, diakses 15 April 2021.