DESIGN OF AN ANDROID-BASED AIRSIDE FACILITY INSPECTION APPLICATION AT H.ASAN SAMPIT AIRPORT IN CENTRAL KALIMANTAN

Bagus Dayana Mikdam[1,*], Linda Winiasri[2], Yudhis Tiro Kabul Yunior[3]

[1,] Politeknik Penerbangan Surabaya, Jemur Andayani I/73 Wonocolo Surabaya, Jawa Timur 60236
Email: [1,]bagusdayana1@gmail.com [2] winiasri@gmail.com [3]yudhis.Kabul@poltekbangsby.co.id*
**Corresponding Author*

ABSTRACT
H.Asan Airport in Sampit, Central Kalimantan, plays a strategic role in supporting regional air transportation. The current manual inspection system for airside facilities has several limitations, including vulnerability to human error and inefficiency in data recording. This study aims to design an Android-based inspection application to improve the accuracy, reliability, and efficiency of daily airside facility inspections. The research uses the waterfall development method and employs MIT App Inventor, integrated with Google Sheets and Firebase, to create a user-friendly and real-time data processing system. The resulting application enables faster data recording, supports decision-making, and helps maintain airside safety and cleanliness. The application has undergone functional testing using black-box methods and proven successful in all intended functions.

Keywords: *Android application, airside inspection, airport maintenance, mobile development, MIT App Inventor*.

1. INTRODUCTION

H.Asan Airport, located in Sampit, Central Kalimantan, serves as a vital transportation hub, facilitating the mobility of people and goods and supporting the local economy. In line with modernization efforts, the airport continuously upgrades its infrastructure and services, including maintenance and inspection operations for both landside and airside facilities.

The Unit of Buildings and Runways (Bangunan & Landasan) is responsible for daily inspections, conducted in accordance with MOS 139 standards. These inspections are recorded manually using paper-based forms, which are vulnerable to damage, loss, and inefficiencies during data processing. This traditional approach also hampers the compilation of long-term trend analyses and delays the identification of critical safety issues.

Through interviews with inspection personnel, it was found that manual methods were highly prone to human error, especially in wet or unpredictable field conditions. To address these challenges, digital transformation through an Android-based inspection application offers a promising solution. Such technology allows real-time data input, minimizes paperwork, and provides better integration with digital documentation tools.

This paper presents the design and development of an airside inspection application tailored for H.Asan Airport. The application focuses on key aspects of the airside, including runway, taxiway, apron, wildlife hazards, surface conditions, and foreign object debris (FOD). Its development follows existing regulations such as KP 220/2017 and KP 94/2015, ensuring compliance with aviation safety standards.

By transitioning to a mobile platform, the inspection process becomes faster, more accurate, and less susceptible to errors. Moreover, the centralized digital storage allows easier tracking, evaluation, and improvement of facility maintenance. Ultimately, this system aims to enhance airside safety and operational efficiency at H.Asan Airport.

2. METHODOLOGY

This research utilized a descriptive qualitative method integrated with a waterfall software development model to design an Android-based airside inspection application. The process emphasized real-world data collection, structured system development, and iterative testing to ensure the application met field requirements at H.Asan Airport, Sampit.

2.1 Research Approach

The research approach was divided into three main components:

1. **Problem Identification**
2. **System Design and Development**
3. **Testing and Evaluation**

All phases were carried out between September 2024 and March 2025 during the author's On-the-Job Training (OJT) at H.Asan Airport. This allowed direct involvement with airside inspectors, real inspection routines, and live system validation.

2.2 Problem Identification and Data Collection

2.2.1 Field Observation

The author accompanied airside personnel daily and observed the real process of inspecting the runway, taxiway, apron, and aircraft stands. All observations were documented in a logbook, including how staff:

- Inspected surface conditions and markings
- Checked for water puddles, rubber build-up, and FOD
- Recorded results on paper checklists, which were vulnerable to weather, smudging, or loss

2.2.2 Interviews

Semi-structured interviews were conducted with five members of the Bangunan & Landasan Unit. Each participant answered questions regarding:

- Pain points in the manual inspection workflow
- Frequency of misrecorded or delayed data
- Needs for a digital solution that could work both online and offline

Interview results showed 3 key issues:

1. Paper checklists were often damaged by rain or wind during inspections.
2. Manual reports took too long to process, delaying follow-ups.
3. There was no centralized digital archive for long-term data analysis.

2.2.3 Document Analysis

Regulatory documents were reviewed to ensure the application would comply with civil aviation standards:

- KP 220/2017 – Staff Instruction 139-01 on Aerodrome Operations
- KP 94/2015 – Advisory Circular CASR 139-23 on Airside Facility Inspection
- PR 21/2023 – Technical and Operational Standards

The checklist items in the app were directly modeled on mandatory inspection points outlined in these documents, such as rubber deposit presence, wildlife hazard indicators, pavement distress, and surface markings.

2.3 System Development Process

The application was built using the Waterfall Model, which consists of five sequential stages:

2.3.1 Requirements Analysis

At this stage, the following needs were finalized:

- Application must work on Android devices
- Must include dropdown options for condition status
- Must save all inspection data digitally
- Must be accessible even without internet (offline data caching)

2.3.2 Design Phase

The User Interface (UI) was drafted on paper and translated into digital mockups using MIT App Inventor's visual blocks.

The application flow was structured as:

1. Login screen
2. Home menu
3. Category selection (Runway, Taxiway, Apron, etc.)
4. Form with pre-filled elements (e.g., Surface Condition, Markings)
5. Submission to Google Sheets via Firebase API
6. Report viewing screen

UI design focused on one-handed usability, large touch targets (for field use), and dropdowns instead of manual text entry to minimize error.

2.3.3 Implementation Phase

- MIT App Inventor was used to build the front-end using block programming.
- Firebase Realtime Database was connected to handle authentication and sync.
- Google Sheets integration was scripted via Google Apps Script to create a dynamic log of inspection entries.

Each element of the app (e.g., FOD detection form) was mapped to the real inspection checklist used at H.Asan.

2.3.4 Testing Phase

Once the implementation stage was completed, the application entered the testing phase to ensure that all functions worked according to expectations. The testing process used the black-box testing approach, where the emphasis was placed on the functionality of each feature without considering the underlying code. Several scenarios were carried out, such as attempting to log in with valid and invalid credentials, submitting inspection forms with complete and incomplete data, and using the application in airplane mode to test offline submission and synchronization once the device reconnected to the internet.

Additional checks included verifying whether the inspection history page displayed data correctly and ensuring that photo attachments could be added to the inspection forms without error. Each of these tests confirmed that the application was able to handle the intended input and produce the expected output, meaning that the login system, form validation, offline data caching, and automatic synchronization with Google Sheets all worked properly. In cases where minor errors appeared during testing, such as incomplete form submission or interface readability issues, the system was revised and retested until the functions operated smoothly. Overall, the testing phase demonstrated that the application was stable, reliable, and ready to be used in field conditions.

2.3.5 Deployment and Review

Following the testing phase, the application was deployed on several Android devices used by airside inspection staff at H.Asan Airport for a trial period of two weeks. During this trial, inspectors carried out daily inspections of the runway, taxiway, and apron using the application instead of the conventional paper-based forms. This real-world deployment provided valuable feedback on how the system performed under actual operational conditions, including exposure to weather and time pressure during inspections. Inspectors reported that the application significantly reduced the time needed to record inspection results and minimized the risk of errors that commonly occurred with handwritten reports. However, some users also highlighted areas for improvement, such as the need for a confirmation dialog before final submission and larger font sizes for better visibility under direct sunlight. These suggestions were incorporated into the final revision of the system, after which the application was considered ready for operational use. The deployment stage not only validated the technical functionality of the system but also confirmed its practicality and usability for the personnel directly responsible for airside inspections.

2.4 Tools and Technologies Used

Tool/Platform	Purpose
MIT App Inventor	Visual block-based app development
Firebase	Real-time database + authentication
Google Sheets	Dynamic storage of inspection reports
Google Apps Script	Script to transfer data from app to spreadsheet

3. RESULT AND DISCUSSION

This section describes the outcomes of designing the Android-based airside inspection application using MIT App Inventor with Google Sheets and Firebase as supporting platforms. The development process followed a structured flow based on the waterfall method. Each part of the system — from UI flow to actual field testing — was aligned with real operational practices at H.Asan Airport in Sampit.

3.1 Application Workflow and Interface

The application was designed with a focus on simplicity and accessibility, especially for airside personnel who often work under time pressure and outdoor conditions. The app flow is structured as follows:

1. **Splash Screen**
    - Displays application logo and loading animation.
2. **Login & Sign-In Page**
    - New users can register with email and password.
    - Existing users can log in securely using Firebase Authentication.

Figure 1 Login Page

3. **Main Menu**
   - Provides options to:
     - Create a new inspection report
     - View previous inspection history
     - Log out

Figure 2 Main Menu

4. **Inspection Menu**
   - Offers inspection categories:
     - Runway
     - Taxiway
     - Apron
     - Parking Stand
     - Service Road
     - Markings
     - FOD
     - Wildlife

Figure 3 Inspection Menu

5. **Inspection Form Page**
   - Contains pre-defined elements based on the selected category.
   - Each element includes a dropdown menu (spinner) to choose the condition (e.g., Safe, Not Safe).
   - Includes text fields for notes and optional photo uploads.

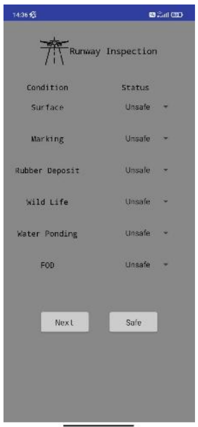

Figure 4 Inspection Form Page

6. **History Page**
   - Displays completed inspection data.
   - Data is stored in Google Sheets and retrievable by category and date.

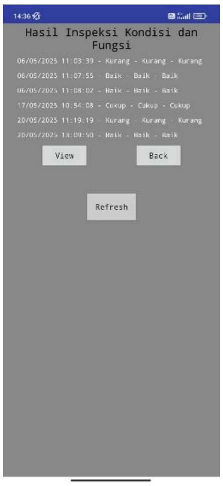

Figure 5 History Page

3.2 Features and Functionality

The inspection application was developed using MIT App Inventor as the primary platform. This choice was based on its ability to simplify application development through visual block programming, which is highly suitable for rapid prototyping. To support the system's functionality, Firebase was integrated as the backend service, providing authentication and secure user login, while Google Sheets served as the database for storing inspection records in a structured and easily retrievable format. The integration between these three components ensures that the application is lightweight, accessible on standard Android devices, and capable of synchronizing inspection data in real time.

In terms of its key features, the application offers several functions that are designed to streamline the inspection process and minimize human error. The login and registration system, supported by Firebase, ensures that each report can be attributed to a specific inspector, thereby increasing accountability. Inspection forms are designed with dropdown menus (spinners) for condition input, reducing typing requirements and avoiding inconsistent terminology. Submitted reports are automatically sent to Google Sheets, eliminating delays in data archiving and improving the efficiency of reporting. Furthermore, the history page allows users to revisit and review past inspection records, which supports better monitoring and follow-up on recurring issues. Together, these features transform what was previously a paper-based process into a more reliable, digital workflow.

Each form within the application is also structured to reflect mandatory checklist items as outlined in national aviation regulations. These include the Civil Aviation Safety Regulation (CASR) Part 139-23, the Staff Instruction KP 220/2017, and PR 21/2023 on technical and operational standards. For example, inspection elements for the runway cover surface condition, markings, rubber deposits, and potential water pooling, while apron and taxiway inspections include pavement quality, cleanliness, and the presence of foreign object debris (FOD). Wildlife observations are also included as a separate category in line with regulatory requirements to ensure airside safety. By aligning the application forms directly with these regulations, the system not only digitizes the inspection process but also ensures compliance with the standards that govern aerodrome operations in Indonesia.

3.3 Application Display Results

The application was tested and demonstrated in real conditions. Screenshots of the working pages include:

- **Login Page** – shows Firebase-linked authentication
- **Main Menu** – minimal interface with three core options
- **Inspection Categories** – tiles representing each inspection type
- **Inspection Form** – dropdown and notes fields for each element
- **History Page** – logged records displayed in list format

Each section works based on the inspection flow done by Unit Bangunan & Landasan, ensuring familiarity and relevance to their daily operations.

3.4 Testing Results

The app was tested using black-box testing, which focused on verifying that:

- Buttons function correctly
- Data flows from the form to Google Sheets
- History display updates as expected

Example Test Results:

Function Tested	**Expected Outcome**	**Result**
Login Function	User logs in with correct email	Success
Registration Function	New account created and stored	Success
Form Input (Dropdowns)	Spinner values are selectable	Success
Submit Report	Data appears in Google Sheets	Success

Function Tested	Expected Outcome	Result
View History	Completed inspections displayed	Success

Testing confirmed that all core features of the app worked correctly and matched the intended inspection process.

3.5 Relevance to Field Needs

By implementing this application, the system now provides a faster and more reliable method for inspectors to:

- Record conditions in real time
- Reduce paper usage
- Eliminate risk of losing manual records
- Improve data accuracy and traceability

Previously, the inspection process relied on handwritten reports that were manually archived, often taking more time and prone to errors or loss.

3.6 System Limitations

Although the Android-based inspection application offers significant improvements over the manual system, several limitations remain in its current version. At this stage, the application is only compatible with Android devices, which restricts accessibility for users operating on iOS or other platforms. In addition, the system does not yet include a scheduling or reminder feature to notify inspectors of routine inspection times, meaning that inspections must still be initiated manually. Another limitation lies in the reporting function, as the data stored in Google Sheets is presented in raw form without visualization or analytical summaries that could support higher-level decision-making by supervisors. Furthermore, the application does not integrate asset management functions such as tracking of equipment, runway lights, or maintenance logs, which could provide a more comprehensive digital tool for airport operations. These limitations indicate that while the prototype effectively digitizes the inspection workflow, further development is necessary to enhance its functionality and adaptability in the future.

4. Conclusion

The design and development of an Android-based airside inspection application for H.Asan Airport in Sampit, Central Kalimantan, have resulted in a functional prototype that addresses key limitations of the previous manual inspection system. By utilizing MIT App Inventor for development and integrating Firebase for authentication and Google Sheets for data storage, the application streamlines the inspection workflow into a more efficient, consistent, and secure process.

The application's features — including category-based checklists, dropdown condition status, and automatic data synchronization — allow inspectors to carry out daily tasks with reduced error risk and improved data accuracy. The history log feature provides accessible and organized inspection archives, supporting better operational monitoring.

Black-box testing confirms that all major functions — login, form completion, data submission, and history view — operate correctly. Field testing also demonstrated that the application can support real-world airport operations, particularly for personnel from the Unit Bangunan dan Landasan.

While the application is limited to the Android platform and lacks advanced features like inspection scheduling or automated analytics, it effectively fulfills its primary goal of digitizing and improving the airside inspection process. Future improvements could focus on expanding platform compatibility, integrating dashboards, and enhancing reporting functions to further increase operational efficiency and decision-making support at the airport.

AUTHORS' CONTRIBUTIONS

The author carried out all stages of this research, including:

- Identifying the problem and setting research objectives
- Designing the system and choosing the development method
- Creating the Android-based inspection application
- Collecting data from field observations and interviews
- Testing the application and making improvements based on feedback
- Writing and preparing the journal manuscript

ACKNOWLEDGMENTS

The author would like to express sincere gratitude to the staff of H.Asan Airport's Building & Runway Unit for their cooperation and insights during field research, as well as to the faculty of Aviation Polytechnic of Surabaya for their guidance and support.

REFERENCES

[1] Z. Mulyadi, Z. Zulkarnain, and M. Laugu, "Teknologi Informasi Dalam Pengembangan Organisasi," *Jurnal Teknologi dan Sistem Informasi*, vol. 9, no. 2, pp. 45–52, 2019.

[2] M. Atoillah, T. Gonzales, and A. Amirulloh, "Pengaruh Versi Android Terhadap Kinerja Aplikasi," *Journal of Mobile Systems*, vol. 7, no. 1, pp. 30–36, 2024.

[3] A. A. Wahid, *Metodologi Pengembangan Perangkat Lunak*, Yogyakarta: Deepublish, 2020.

[4] A. Irawan, N. Desryanto, and I. H. Wibowo, "Manajemen Pemeliharaan Fasilitas Transportasi," *Jurnal Infrastruktur dan Logistik*, vol. 6, no. 3, pp. 112–118, 2018.

[5] Directorate General of Civil Aviation, "Staff Instruction 139-01," Ministry of Transportation, Indonesia, 2017.

[6] Directorate General of Civil Aviation, "Advisory Circular CASR Part 139-23," Ministry of Transportation, Indonesia, 2015.

[7] Directorate General of Civil Aviation, "PR 21 Tahun 2023 – Standar Teknis dan Operasional Bandara," Ministry of Transportation, 2023.

[8] T. Yanto, M. Lamada, and F. Ramadhani, "Introduction to MIT App Inventor," *Journal of Educational Technology*, vol. 10, no. 1, pp. 50–56, 2022.

[9] R. Hartanto, B. Angga, and M. Ramdan, "Pemanfaatan Firebase Dalam Sistem Informasi Mobile," *Jurnal Teknologi Informasi dan Aplikasi*, vol. 12, no. 2, pp. 67–74, 2023.

[10] P. Gonsior et al., "Spreadsheet Systems in Data Management," *International Journal of Data Applications*, vol. 5, no. 2, pp. 91–99, 2020.