International Journal of Informatics and Computation (IJICOM)
Vol. 7, No.1, 2025
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Prototype of Web-Based Inventory System for Ringgit Village Office using ADDIE Model

Ilham Afrianto[1], Sri Mulyati[2]

Abstract
This study presents the development of a web-based Inventory Management Information System for the Ringgit Village Office using the ADDIE (Analysis, Design, Development, Implementation, Evaluation) model as a methodological framework. The system was designed to address existing inefficiencies in asset tracking and management, leveraging Google Apps Script to ensure accessibility, cost-effectiveness, and ease of use. Each phase of the ADDIE model contributed systematically to the prototype's success, from identifying user needs to validating the system's functionality and usability. The implementation demonstrated that the ADDIE model supports structured and iterative development, enabling alignment between system features and institutional requirements. The resulting prototype improved asset transparency, accountability, and operational efficiency within the village administration. This study highlights the potential of combining instructional design models with lightweight web technologies to produce scalable and sustainable digital solutions for local governance.

Keywords:
Web-Based System, Inventory Management, ADDIE Model, Village Office Automation

1. Introduction

Inventory management plays a critical role in supporting the transparency and efficiency of village governance. Many village offices still rely on manual recording methods such as ledgers and spreadsheets, which are prone to human error, data duplication, and poor traceability. This inefficiency can lead to asset loss, mismanagement, and limited accountability. As digital transformation reaches even rural government institutions, there is a pressing need for web-based systems to digitize asset management to improve service delivery and support local development [4], [10].

Manual inventory systems often suffer from delayed updates, limited access, and a lack of real-time monitoring. Additionally, most existing systems are not scalable and do not support multi-user collaboration, which is essential in administrative environments where multiple staff members access inventory records. This creates inconsistencies and increases the risk of data manipulation or asset misuse [1],[6]. Thus, a shift toward automated, centralized inventory systems is necessary to improve both data integrity and operational control.

Web-based systems offer several advantages over standalone software, including accessibility from multiple devices, real-time synchronization, and easier data backup. Tools like Google Workspace and Google Apps Script allow the development of lightweight, cloud-based systems without the need for complex infrastructure. These technologies have been used effectively in small-scale information systems due to their low cost and ease of deployment, making them highly suitable for local government applications in villages with limited IT resources [12], [2].

Studies have shown that custom-developed inventory systems tailored to village needs

Corresponding Author: Sri Mulyati, Universitas Islam Indonesia, Yogyakarta (email: mulya@uii.ac.id)
1 Ilham Afrianto, Universitas Islam Indonesia, Yogyakarta
2 Sri Mulyati, Universitas Islam Indonesia, Yogyakarta

can significantly improve the accuracy and timeliness of data collection and reporting. For example, web-based applications have been successfully implemented to manage population data, administrative records, and local asset tracking in similar rural contexts. These systems are also more adaptable to the workflow of local offices, which often differ from centralized government procedures [8], [11].

The integration of digital tools in village-level governance is part of a broader movement to modernize public service through low-cost innovation. Projects using platforms like Google Sites and Google Apps Script demonstrate that even non-developers or semi-technical staff can maintain and adapt systems independently. This reduces the dependency on external vendors and supports knowledge transfer within the community [12], [5]. As a result, village offices can build capacity for long-term system sustainability and adaptability.

Despite the growing use of web-based systems, very few prototypes have focused specifically on using the Google Script environment for inventory management at the village level. Most research has explored commercial platforms or traditional web development stacks. This study addresses that gap by designing and evaluating a prototype using Google Apps Script, which combines simplicity, accessibility, and cloud integration. It targets the operational inefficiencies in Ringgit Village Office and proposes a scalable and maintainable digital solution [3], [7], [9].

2. Literature Review

In conducting this research, we conducted a literature review on similar studies. An article developed a web-based inventory information system at the Faculty of Information Technology, Universitas Merdeka. The study aimed to reduce issues caused by manual inventory tracking, such as data loss and input errors. By utilizing web technology, the system allowed centralized access, reduced duplication, and improved item monitoring across departments. This study demonstrates the effectiveness of simple web-based systems in managing institutional inventory, laying a foundational approach for government-level applications like those at the Ringgit Village Office [1].

Another work focused on developing a web-based inventory system for a rural village (Desa Matang Danau). The research emphasized user accessibility and ease of use for non-technical staff. The interface allowed real-time updates and item tracking. Results showed increased efficiency in documentation and reduced data inconsistencies. This aligns closely with the Ringgit Village case, where local staff require lightweight, cloud-accessible tools to track village assets [2].

A paper was designed for a school inventory system for SMP Katolik Padadita using basic web technologies. The study showed that a simple, browser-accessible system could fulfill inventory recording, searching, and reporting needs. Despite limited infrastructure, the school staff successfully operated and maintained the system. This supports the feasibility of implementing a Google Script-based prototype in Ringgit Village, where similar infrastructure limitations may exist [3]. Another work proposed a web-based asset inventory system using the Waterfall method. The system tracked incoming and outgoing assets, offered data export options, and had user access roles. Their evaluation showed improvements in accuracy and access control. This structured development methodology and successful implementation reinforce the importance of planning and iterative development for similar systems using Google Apps Script [4].

An article developed a web-based information system for managing village assets. The result was a user-friendly system that supported item input, report generation, and accountability. Their findings showed increased public trust and transparency due to better

information flow. This is relevant to the Ringgit Village study, where digital inventory systems are expected to support governance and community accountability [5]. Another paper explored the use of Google Sites and Google Apps Script in educational environments. The results revealed that Google's tools offered flexibility, ease of integration, and accessibility even to users with minimal programming skills. These findings validate the use of Google Script as a development platform for low-cost public service systems, such as the proposed prototype for Ringgit Village Office [6].

Table 1: Literature Review Findings

No.	Reference	Context / Scope	Key Features / Tools Used	Study Result / Relevance
[1]	Abdi (2018)	Faculty inventory system	Web-based PHP/MySQL	Reduced errors; centralized tracking
[2]	Kesuma (2023)	Village inventory management	Web system with CRUD features	Efficient local data access
[3]	Emu (2023)	School asset system	Simple web forms and reports	Suitable for low infrastructure
[4]	Usnaini et al. (2021)	Asset inventory using the waterfall	Structured module-based system	Role access: clear record flow
[5]	Syari & Supriyono (2020)	Village asset management	Web-based with report modules	Improved transparency
[6]	Sosa et al. (2023)	Google Sites and Apps Script usage	Google-based tools in education	Proven accessibility and low cost

3. Proposed Method

This research adopts the ADDIE development model used an instructional design framework. The development of a web-based inventory management system follows a structured methodology grounded in the ADDIE model, beginning with the analysis phase. This phase involves collecting detailed information on user requirements, system limitations, and organizational needs to identify inefficiencies in the current inventory process. By conducting interviews, observations, and document reviews, developers can define functional requirements that align with user expectations and business objectives. Following this, the design phase focuses on translating these requirements into a technical blueprint. System architecture, user interfaces, database schema, and interaction flows are carefully outlined to ensure that the solution is technically feasible, scalable, and user-friendly, providing a clear foundation for the upcoming development process.

In the development phase, the design specifications are translated into a working application using appropriate programming languages and web technologies. This stage includes coding, system integration, and preliminary testing to ensure that the features function as intended. Once the system is operational, the implementation phase introduces the platform into the real working environment. Training sessions and deployment procedures are conducted to support user adoption and minimize disruption. Finally, the evaluation phase is conducted both during and after implementation to measure system performance, user satisfaction, and overall effectiveness. Feedback collected during this stage informs future improvements, ensuring the system remains aligned with organizational goals, as Fig. 1.

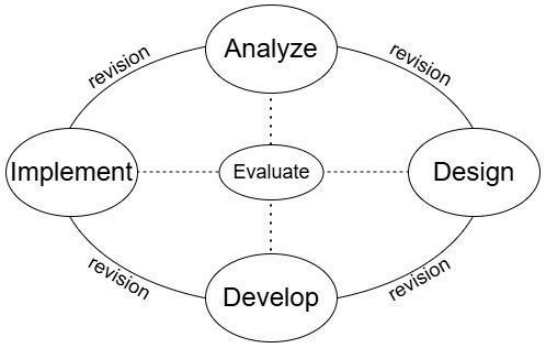

Fig. 1 ADDIE development model

In this study, we express the ADDIE development model in a mathematical form by representing it as a discrete sequential function, and optionally model iterative refinement.

Mathematical Description of the ADDIE Model

Let:

- A: Analysis phase output
- D_1 : Design phase output
- D_2 : Development phase output
- I: Implementation phase output
- E: Evaluation result
- P: Final product or system

Then, the ADDIE model can be described as a composition of functions:

$$P = E \circ I \circ D_2 \circ D_1 \circ A$$

Where each phase is a transformation function:

- $A = f_1(R)$: Analyze user requirements R
- $D_1 = f_2(A)$: Design learning or system structure based on analysis
- $D_2 = f_3(D_1)$: Develop content or software components
- $I = f_4(D_2)$: Implement or deploy in real environment
- $E = f_5(I)$: Evaluate effectiveness and usability

So:

$$P = f_5(f_4(f_3(f_2(f_1(R)))))$$

Iterative Model with Feedback Loop

The ADDIE model is **not strictly linear**; it incorporates feedback. Let iteration index be $t \in \mathbb{N}$. Then:

$$A^{(t)} \to D_1^{(t)} \to D_2^{(t)} \to I^{(t)} \to E^{(t)}$$

If $E^{(t)}$ does **not** meet the acceptance threshold θ, then feedback is used:

$$\text{If } E^{(t)} < \theta, \text{ then } A^{(t+1)} = \text{revise}(E^{(t)})$$

The system iterates until:

$$E^{(t)} \geq \theta$$

Alternative Notation (Process Pipeline)

The ADDIE process can be written in vectorized or pipeline form:

$$P = \text{ADDIE}(R) = (A, D_1, D_2, I, E)$$

Where:

- Each stage is a function of the previous output
- P is the deployed and validated software system

In this study, the ADDIE model used in instructional design and software development, can be mathematically described as a sequential composition of functions that transform user requirements into a validated final product. Represented as $P = f_5(f_4(f_3(f_2(f_1(R)))))$, each function f_i corresponds to a phase in the ADDIE process: Analysis f_1, Design f_2, Development f_3, Implementation f_4, and Evaluation f_5. The input R represents initial user or system requirements, which are refined through each stage to produce P, the final software or instructional output. This formulation not only captures the linear flow of ADDIE but also allows for iterative refinement, where evaluation results feed back into earlier stages if the performance threshold θ is not met. Such a structured model supports systematic development while enabling adaptability and continuous improvement.

4. System Design

The Web-Based Village Asset Inventory Management Information System is developed based on a use case diagram that illustrates the interaction between the administrator and the system as Fig. 2.

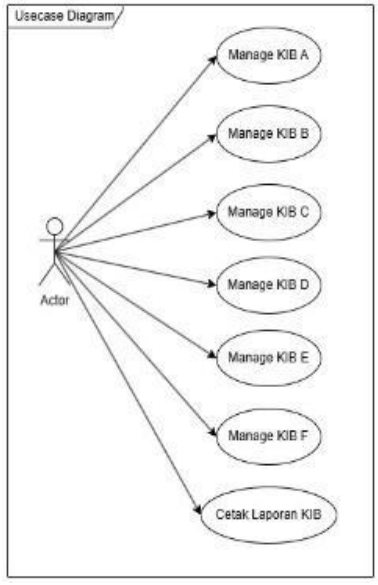

Fig. 2 Use Case Diagram

In the system design, the administrator can manage various types of Inventory Cards (KIB-A to KIB-F), including adding, modifying, and deleting data. We can represent the use case diagram in a mathematical representation by translating its elements into sets, relations, and functions as follows:
At the first stage, we define the ADDIE system as a tuple:

$$\text{UC} = (A, U, R, I, E)$$

Where:

- A: Set of actors
- U: Set of use cases
- R: Actor-to-use case mapping (participation)
- I: Include the relationship between use cases
- E: Extend the relationship between use cases

1. Actors Set
Let:

$$A = a_1, a_2, \ldots, a_n$$

be the set of **actors**, e.g.,

$$A = a_1, a_2, \ldots, a_n$$

2. Use Case Set

Let:

$$U = u_1, u_2, \ldots, u_m$$

be the set of **use cases**, e.g.,

$U =$ Login,Register,Manage Users,View Data,Generate Report

3. Participation Relation

Define a **relation** between actors and use cases:

$$R \subseteq A \times U$$

Where $(a_i, u_j) \in R$ means actor a_i participates in use case u_j

For example:

$R =$ (User,Login), (User,Register), (Admin,Login), (Admin,Manage Users), (Admin,Generate Report)

4. Include and Extend Relations

Let:

- $I \subseteq U \times U$ be the **include relation**: if $(u_i, u_j) \in I$, then use case u_i includes u_j
- $E \subseteq U \times U$ be the **extend relation**: if $(u_k, u_l) \in E$, then u_k optionally extends u_l

So:

$I = \{$(Generate Report,Login)$\}, E = \{$(Reset Password,Login)$\}$

5. Results and Analysis

In this study, we develop a web-based inventory model using Google Apps Script to enhance the efficiency of inventory management at the Ringgit Village Office with Google Sheets as the primary database. In this prototype, the system enables automatic recording and updating of inventory data while ensuring accessibility at any time. App Script serves as a data input tool, allowing users to easily add or update item information. The following outlines the implementation of the village inventory management system as Fig. 3.

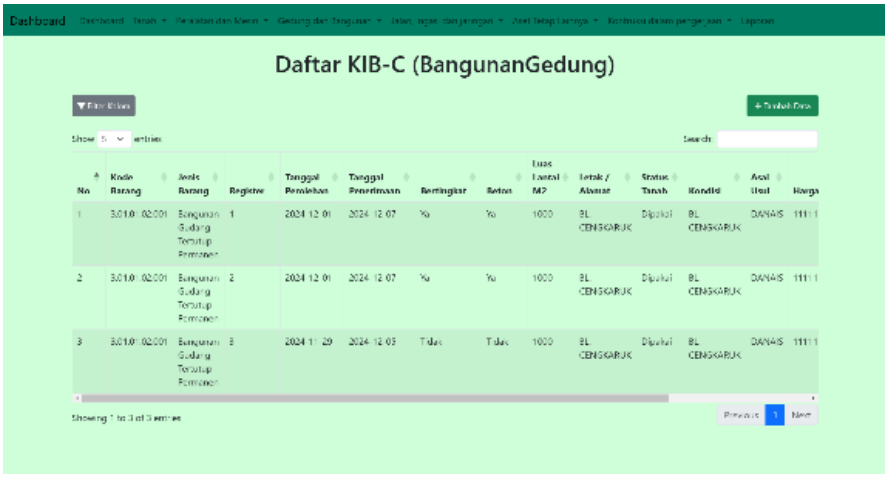

Fig. 3 Implementation of Item Data Page

Fig. 3 presents the implementation of the item data management interface categorized under KIB (Kartu Inventaris Barang), a standard classification system for government-owned assets in Indonesia. This web-based interface serves as the core component of the inventory management system, enabling a structured and categorized representation of item data. The system organizes assets according to KIB categories (such as KIB A, KIB B, etc.), ensuring compliance with official asset documentation frameworks. Each row in the displayed table represents an inventory item, showing critical details such as item name, asset code, acquisition year, quantity, condition, and location to provide users with a clear overview of asset records in a centralized format.

6. Conclusion

The results of this study demonstrate that the application of the ADDIE development model provides a structured and effective framework for constructing a web-based Inventory Management Information System tailored to the needs of the Ringgit Village Office. Each phase of ADDIE contributed systematically to the system's success: the analysis phase identified critical asset management challenges; the design phase translated these findings into functional system architecture; the development phase operationalized the system using Google Apps Script; implementation ensured smooth deployment; and the evaluation phase validated its performance and usability. This structured model allowed for iterative refinement, user-centered development, and ensured alignment between system functionality and organizational requirements.

The resulting prototype offers a sustainable and accessible solution that enhances transparency, accountability, and asset control within the village administration. The use of Google Apps Script further ensures that the system remains cost-effective, cloud-integrated, and adaptable by local staff. By adopting the ADDIE model, this research not only delivered a functional inventory system but also established a replicable methodology for other rural or resource-constrained administrations. Future work may involve scaling the prototype, integrating it with other e-Government systems, and evaluating long-term user engagement, thereby further reinforcing the system's contribution to improved digital governance at the village level.

References

[1] B. B. Abdi, "Web-based inventory information system design for the Faculty of Information Technology at Universitas Merdeka," *Jurnal Sistem Informasi dan Komputerisasi*, vol. 4, no. 2, pp. 121–128, 2018.

[2] R. S. Hartono, "Implementation of village government administration in supporting community service and development in Bekasi Regency," *Jurnal Ilmu Sosial dan Pemerintahan*, vol. 6, no. 1, pp. 35–42, 2019. [Online]. Available: https://doi.org/10.32528/pi.v0i0.2503

[3] Kusnadi, "The development of political law on village governance according to Law No. 32 of 2004 and Law No. 6 of 2014," *Padjadjaran Journal of Law*, vol. 2, no. 3, pp. 564–580, 2015. [Online]. Available: https://doi.org/10.22304/pjih.v2n3.a8

[4] S. S. Kesuma, "Design of a web-based village inventory information system: A case study in Matang Danau Village, Sambas," *JURISTI (Jurnal Riset Sains dan Teknologi Informatika)*, vol. 1, no. 1, pp. 36–42, 2023.

[5] F. M. Emu, "Web-based inventory system for school asset management: A case study at SMP Katolik Padadita," *Jurnal Teknologi dan Sistem Informasi*, vol. 5, no. 1, pp. 88–95, 2023.

[6] M. S. Jannah, "Web-based asset information system (SIMaset) for managing goods and documents," *Teknosains: Media Informasi Sains dan Teknologi*, vol. 15, no. 3, pp. 367–380, 2021.

[7] M. Usnaini, V. Yasin, and A. Z. Sianipar, “Web-based asset inventory system design using the waterfall method,” *Jurnal Manajemen Informatika Jayakarta*, vol. 1, no. 1, pp. 36–42, Feb. 2021. [Online]. Available: https://doi.org/10.52362/jmijayakarta.v1i1.415

[8] C. K. Sastradipraja, R. Hidayat, H. Ramadan, and A. Subagja, “Village population data system using the ADDIE development method,” *Jurnal Rekayasa Teknologi Nusa Putra*, vol. 2, no. 1, pp. 15–22, 2015. [Online]. Available: https://doi.org/10.52005/rekayasa.v8i1.117

[9] M. Huda, S. Wiyono, M. F. Hidayatullah, and S. Bahri, “Case study: Information systems and administrative population services,” *Komputika: Jurnal Sistem Komputer*, vol. 9, no. 1, pp. 59–65, 2020. [Online]. Available: https://doi.org/10.34010/komputika.v9i1.2518

[10] E. Mardinata, T. D. Cahyono, and R. M. Rizqi, “Digital transformation of villages through the village information system (SID): Enhancing public service and community welfare,” *Parta: Jurnal Pengabdian Kepada Masyarakat*, vol. 4, no. 1, pp. 73–81, 2023. [Online]. Available: https://doi.org/10.38043/parta.v4i1.4402

[11] R. S. Syari and H. Supriyono, “Web-based information system for managing village assets,” *JPPM (Jurnal Pendidikan dan Pemberdayaan Masyarakat)*, vol. 7, no. 1, pp. 45–56, 2020. [Online]. Available: https://doi.org/10.21831/jppm.v7i1.31464

[12] A. Sosa, K. R. Coronel, and J. B. Garófalo, “Google Sites: An effective and accessible tool to enhance mathematics instruction in virtual learning environments,” *IEEE Access*, vol. 11, pp. 1–5, 2023. [Online]. Available: https://doi.org/10.1109/ETCM58927.2023.10309017