Document Tracking System Software Engineering with Rapid Application Development (RAD) Methodology

Muhammad Noor[1], Rezki Kurniati[2], Lidya Wati[3], Sri Mawarni[4], Ricky Riotaro[5]
[1]*School of Computing, Universiti Utara Malaysia, Sintok, Kedah, Malaysia*
[2,3,4,5]*Informatics Engineering Department, Politeknik Negeri Bengkalis, Indonesia*
m_noor@ahsgs.uum.edu.my[1] , rezki@polbeng.ac.id[2], lidyawati@polbeng.ac.id[3], rickyriotaro288@gmail.com[4]

Abstract

In the era of digital transformation, efficient document management is crucial for organizational effectiveness. Manual document tracking often leads to delays, traceability issues, and loss risks. This study focuses on the development of a Document Tracking System (DTS) using a systematic Software Engineering (SE) approach and the Rapid Application Development (RAD) methodology. RAD is chosen for its rapid prototyping and intensive user involvement, ensuring the final product meets dynamic user needs quickly. This article comprehensively outlines the SE phases for building the DTS, from requirements analysis, through iterative prototyping, to implementation and validation. The resulting system is expected to automate workflows, enhance accountability, and provide realtime document status monitoring, thereby demonstrating the practical and academic significance of applying modern, agile methodologies to concrete business problems. The methodology phase details the four core stages of RAD: Requirements Planning, User Design, Construction, and Cutover, setting the foundation for the implementation and discussion of the system's key functional and nonfunctional features.

Keywords: Document Tracking System, Software Engineering, Rapid Application Development, Prototyping, Document Management.

1. INTRODUCTION

In the midst of digital transformation, the speed and transparency of information flows have become decisive factors for organizational effectiveness. Document management, the backbone of business processes and bureaucracy often encounters serious obstacles when handled manually. Common issues such as distribution delays, difficulty tracking document location and status in real time, and the risk of file loss can hinder decision making and reduce productivity (Laudon & Laudon, 2020). To address these challenges, developing a Document Tracking System (DTS) offers a strategic solution that automates workflows, strengthens accountability, and enables end to end monitoring of each document's lifecycle.

Building such a complex information system requires a structured discipline to ensure the quality and success of the final product. Accordingly, we adopt a Software Engineering approach, which defines a systematic set of processes from requirements analysis and design to implementation and testing (Pressman & Maxim, 2020). This framework explicitly positions system development not merely as programming, but as a measurable, scientifically accountable engineering process, ensuring that the resulting solution truly satisfies users' functional and non functional requirements.

Several previous researchers have discussed document management systems using system development methods, such as the Agile-SSDLC method, which emphasizes system security (Setiawan, et al, 2024) and (Mendonca & Sulianta, 2025), which uses the Object Oriented Analysis and Design (OOAD) approach and the waterfall method in its design. Meanwhile, several other researchers used the Rapid Application Development (RAD) method in building applications, such as (Putra, et al, 2025), who built a Document Management System application, (Sari, et al, 2024), who built a monitoring information system, and (Hidayatullah & Patyani, 2024), who developed a web-based library information system. From the results of the literature review, recently researchers tend to use the RAD method in developing

applications and information systems because this method, when used to develop information

systems, requires a fairly efficient time due to the few stages in this method that need to be passed (Subhiyanto, 2023).

Selecting an appropriate software development methodology is critical to implementation speed and accuracy. Traditional approaches often feature long, rigid life cycles and are less responsive to dynamic user needs. In this context, Rapid Application Development (RAD) is chosen for its iterative, fast prototyping orientation. As Martin (1991) notes, RAD emphasizes active user involvement throughout short development cycles, enabling continuous validation and feedback. This significantly reduces the risk of a final product that diverges from expectations and accelerates the delivery of a functional solution to users.

Building on these three pillars, the urgency of this research lies in synthesizing an immediate business need for an efficient Document Tracking System with a systematic software engineering discipline and the use of RAD as a development accelerator. This combination is not only practically relevant for improving organizational performance, but also academically significant in demonstrating the application of modern development methods to a concrete case study. Accordingly, this article aims to comprehensively outline the endtoend software engineering stages for constructing a Document Tracking System using the RAD methodology from planning through implementation.

2. REVIEW OF LITERATURE

2.1 Software Engineering

Software Engineering (SE) is a discipline that applies systematic, measurable, and structured approaches to the development, operation, and maintenance of software. According to Pressman & Maxim (2020), SE provides the technical foundation that enables developers to build high quality software efficiently and effectively. The process comprises a set of fundamental phases: requirements analysis, design, coding, testing, and maintenance. Applying SE ensures that the resulting software not only meets users' desired functional specifications, but also achieves strong non functional qualities such as reliability, scalability, and maintainability.

2.2 Document Tracking System

A Document Tracking System is an information system specifically designed to monitor and manage document workflows and lifecycles within an organization. Its primary goal is to enhance efficiency, transparency, and accountability in document management (Laudon & Laudon, 2020). Essential features typically include logging every document activity, realtime monitoring of document status and location, automated notifications to relevant parties, and accesscontrol management to ensure information security. By automating tracking processes, organizations can minimize the risk of document loss, accelerate approvals, and simplify audit trails for all circulating documents.

2.3 Rapid Application Development (RAD) Methodology

Rapid Application Development (RAD) is an iterative software development process model that emphasizes very short development cycles. Introduced by James Martin (1991), this methodology is designed to produce high quality systems in less time than traditional approaches such as Waterfall. RAD relies heavily on prototype driven development and active end user involvement to continuously validate system design and functionality.

RAD model four core phases:

1. Requirements Planning
   Stakeholdersend users, managers, and the IT teamcollaborate to define the system's goals, constraints, and highlevel requirements.
2. User Design
   The development team builds functional prototypes (e.g., interfaces and reports) that are iteratively reviewed and refined based on direct user feedback.

3. Construction
   Approved prototypes are evolved into a fully working applicationoften accelerated with CASE toolsandreusable components to shorten build time.
4. Cutover
   Final activities: comprehensive testing, user training, data migration (if needed), and deployment to the operational environment.

Key advantage RAD excels at adapting to changing requirements and significantly reducing development time, making it ideal for projects that demand rapid delivery while closely aligning with user expectations.

3. METHOD

This study adopts the RAD methodology, emphasizing rapid, iterative development cycles. The approach was chosen to ensure intensive user involvement, which is crucial for validating complex document workflows

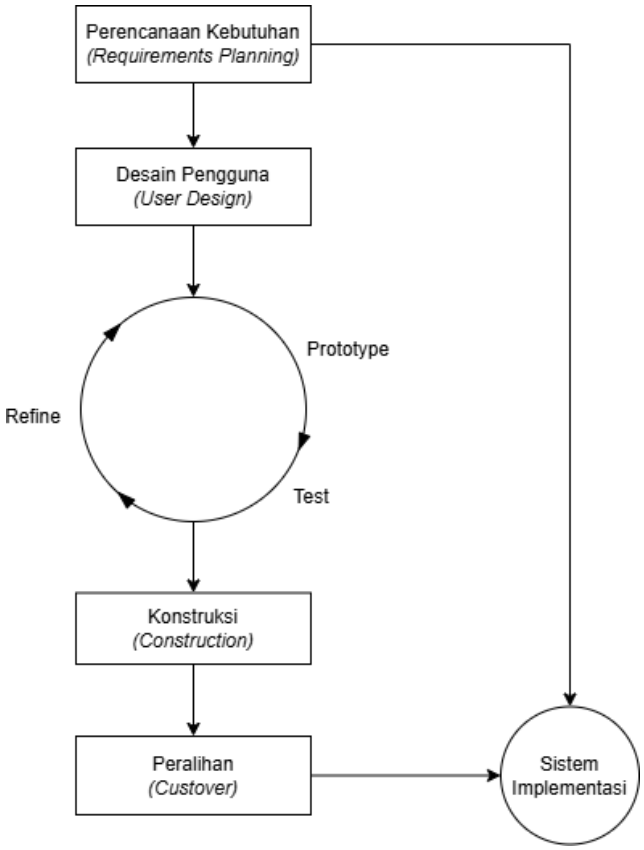

Figure 1. Flowchart of the RAD Methodology

3.1 Type and Location of the Study

This study is an applied research project in the field of Software Engineering. The research was conducted within the operational environment of Politeknik Negeri Bengkalis, with participants encompassing the key roles involved in administrative and correspondence workflows, including Administrator, Head of Study Program (Kaprodi), Department Secretary (Sekjur), Department Head (Kajur), Executive Secretary (Sekdir), Director, Vice Director (Wadir), and Deputy Executive Secretary (Sekwadir).

3.2 System Development Methodology (RAD)

The procedures follow the four core stages of the RAD methodology, with the user design and Construction phases executed iteratively, as illustrated in Figure 1. The steps are:

1. Requirements Planning

- Data Collection. Observation of current manual document workflows and informal interviews with representatives of the user roles listed above.

- Requirements Definition. Specification of functional needs (e.g., creating, sending, validating, and tracking document status) and nonfunctional needs (e.g., security, performance, and usability).
- Security Focus. Integration of the five security pillarsConfidentiality, Availability, Authentication, Authorization, and Accountabilityfrom the planning stage, referencing securedevelopment guidelines (Cyber Security Malaysia, 2020).

2. User Design

- Iterative Prototyping**.** UI prototypes designed with visual tools, emphasizing intuitive and secure interfaces.
- Iteration & Feedback. Multiple design iterations; prototypes are tested by user representatives and refined based on direct feedback to ensure usability and workflow fit (Tisna et al., 2024).

3. Construction

- System Development. Approved prototypes are built into a functional application.
- Technology Stack. The system is implemented using PHP and JavaScript.
- Security Implementation**.** Core mechanisms are appliedsecure login (authentication), session management, and RBAC.

4. Cutover

- Functional Testing**.** Verification that each role can access only the menus and features authorized by the defined RBAC.
- User Evaluation. Collecting feedback via short surveys on usability and perceived security.
- Deployment**.** Installing and operating the system in the production environment after final testing is complete.

4. RESULT & DISCUSSION

This section presents the implementation outcomes and an in-depth discussion of how the RAD methodology and Software Engineering practices were applied to deliver a functional and secure Document Tracking System (DTS).

A. Letter Management Procedure (Current)

The Head of Study Program submits a letter to the Department. Upon receipt, the Department Head (Kajur) assigns a disposition to the Department Secretary (Sekjur) for processing. The Department Administrator then drafts and issues the departmental letter addressed to the target unit (e.g., Director, Vice Director I/II/III, or Head of Center). The departmental letter is forwarded to the Sekdir of the destination unit to be conveyed to the Leadership: once received, the Leadership provides a disposition. Based on this disposition, the Sekdir processes the letter according to its intended purpose. If the expected output is a Decree (SK) or an Invitation Letter, the Sekdir returns the finalized document to the Department; the Department then receives and archives the document. The overall flow is illustrated in Figure 2.

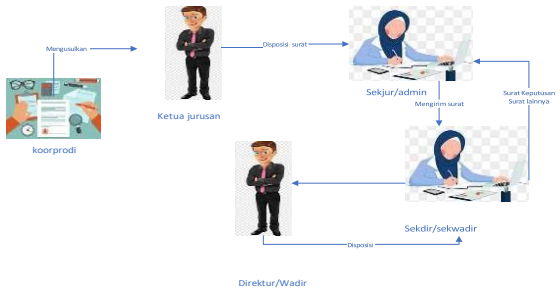

Figure 2. Rich Picture Of The Letter Management Process

B. Proposed System Design

1. Requirements Planning

The diagram depicts the system boundary of the Letter Management System with three external actors Kaprodi, Leadership (Director/Vice Director/Department Head), and Admin/Sekdir/Sekjur. Within the boundary, six core use cases are defined submit letter, Letter Management Procedure (Current) and disposition. The flow illustrates how a proposal from the Head of Study Program is processed by the department/secretariat unit and decided by leadership, with a validation/check step recorded prior to disposition (figure 3)

Actors & Roles.

- Kaprodi initiates the process via the submit letter use case (submitting a draft/request to the system).
- Admin/Sekdir/Sekjur handles day-to-day operations, compose letter (prepare official letters/memos), receive letter (log and forward), and view letter (review status/content for follow-up). These actors also participate in administrative validation.
- Leadership (Director/Vice Director/Department Head): authorized to perform validation (substance/authority checks) and disposition (decision/direction for follow-up) on incoming letters.

Relationships among Use Cases.

- Validation is a prerequisite for disposition (conceptually modeled as `<<include>>` from *disposition* to *validation*), ensuring decisions are issued only for letters that have passed the checks.
- View letter supports both validation and disposition by providing access to content and metadata for decision-making.
- Receive letter serves as the entry point for documents into the system before they can be validated/disposed.

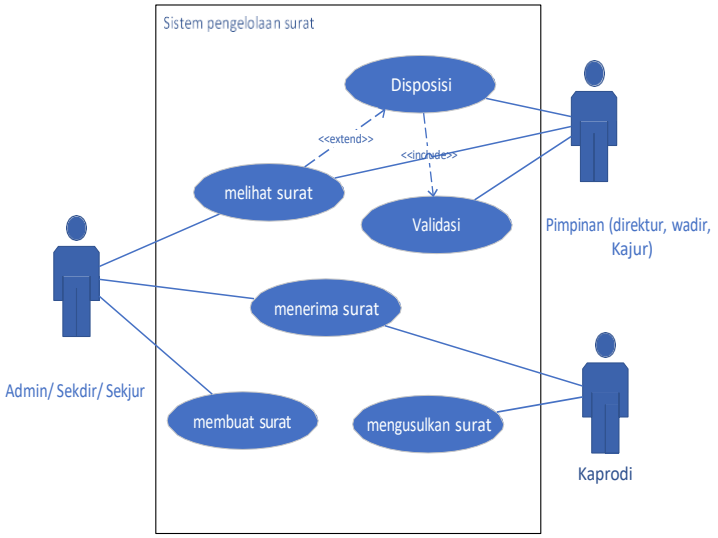

Figure 3. Use Case Diagram Of The Letter Management System.

2. User Design

a) System Login Page UI

Confidentiality and authentication are realized through a securely designed login page. This page serves as the primary gateway that protects data from unauthorized access. Figure 4 presents the security-focused login interface design, while Figure 5 Brute-Force Attack Mitigation Mechanism.

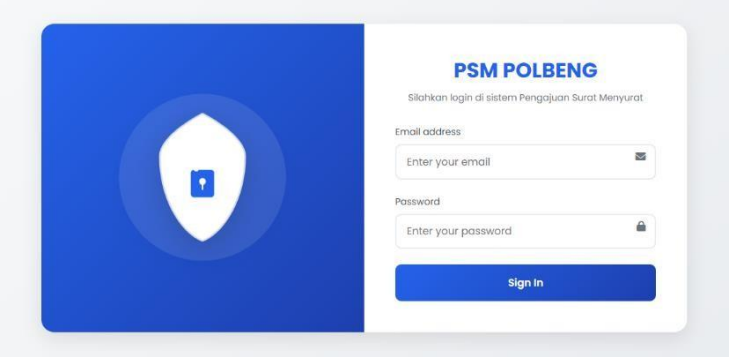

Figure 4. DTS System Login Page Interface.

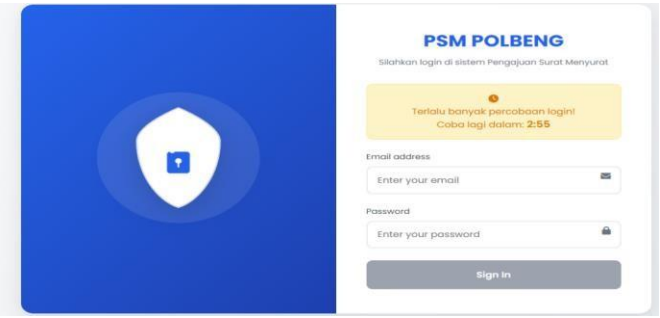

Figure 5. Brute-Force Attack Mitigation Mechanism (Account Lockout)

b) Login Failure Warning Interface

Secure Login Implementation: The login check is performed by comparing the user-entered email and password with the records stored in the database. If the authentication succeeds, the system creates a session ($_SESSION) that stores the user's essential data, including their role ($_SESSION['role']).

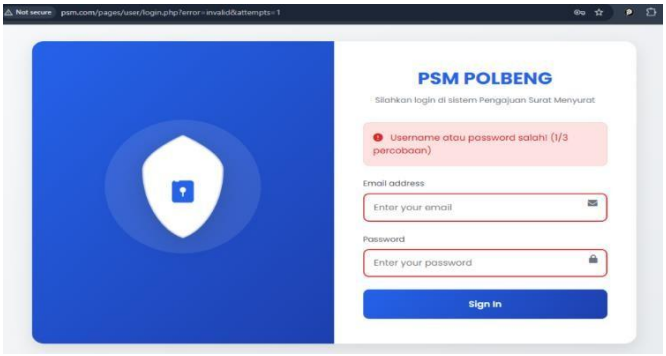

Figure 6. Login Failure Warning Interface.

c) Letter Creation and Submission

The workflow begins when a user with the appropriate role (e.g., Head of Study Program/Kaprodi) creates a new document. Figure 7 shows the letter creation form interface. After completing the letter, the user submits it to the next authorized party (the Department Secretary/Sekjur), as shown in Figure 8.

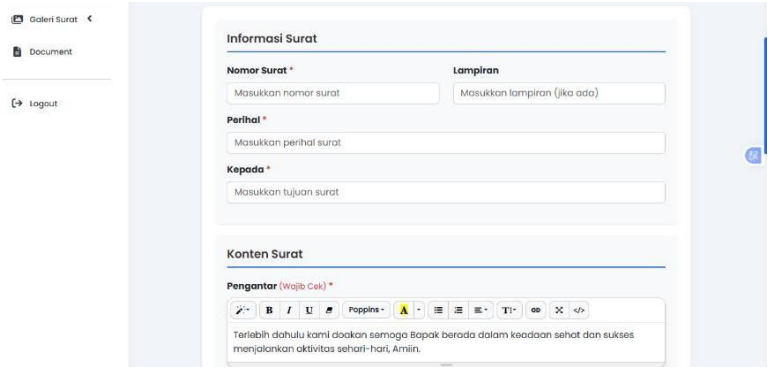

Figure 7. User Interface for Creating a New Letter.

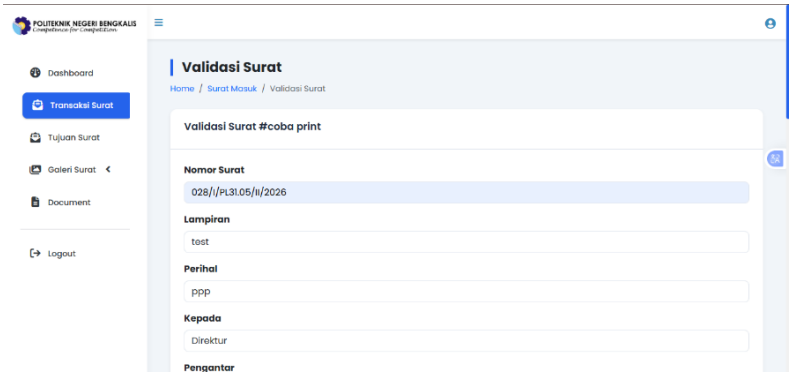

Figure 8. Interface for Submitting the Letter to the Next Role.

Receipt and Verification of Letters

After a letter is submitted, it appears in the "Incoming Letters" menu for the receiving role (Sekjur or Admin). This stage includes the Receive and Verify functions.

- Receive: Figure 9 shows the screen where the receiving role confirms receipt of the document; the system then records an audit log (timestamp and recipient).
- Verify: Before a letter can be forwarded to higher leadership, a designated role (e.g., Sekjur or Kajur) must perform Verification. Figure 10 shows the verification interface, which often includes options for Digital Signature or formal approval.

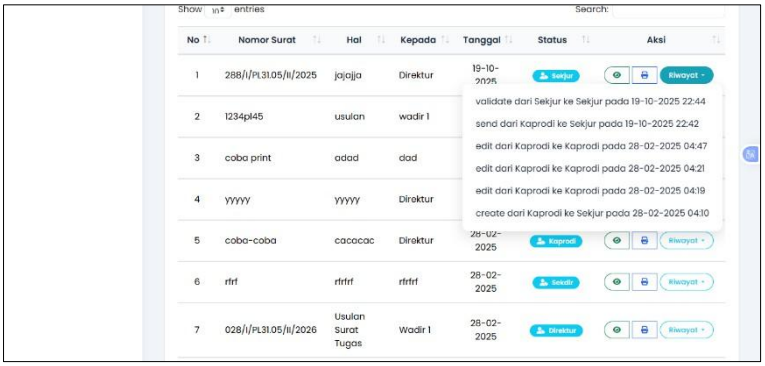

Figure 10. Activity Log and Letter Receipt View (Receiving Role).

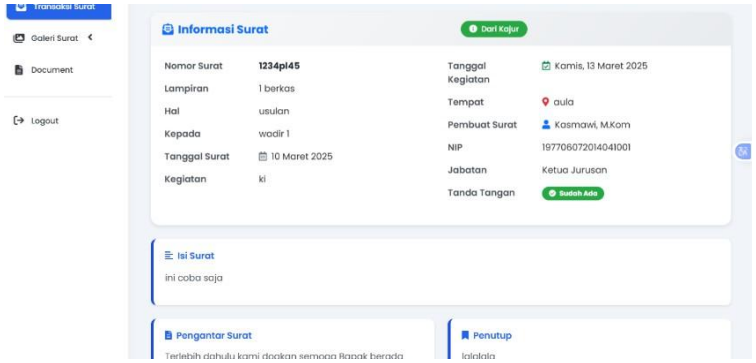

Figure 11. Letter Verification Interface.

d) Letter Disposition

After a letter is verified and/or validated by the authorized party (Director/Vice Director), the final step is Disposition. This stage ensures the letter is routed to the responsible unit or staff for follow-up. Figure 12 displays the interface for recording dispositions, which is crucial for guaranteeing accountability within the document workflow.

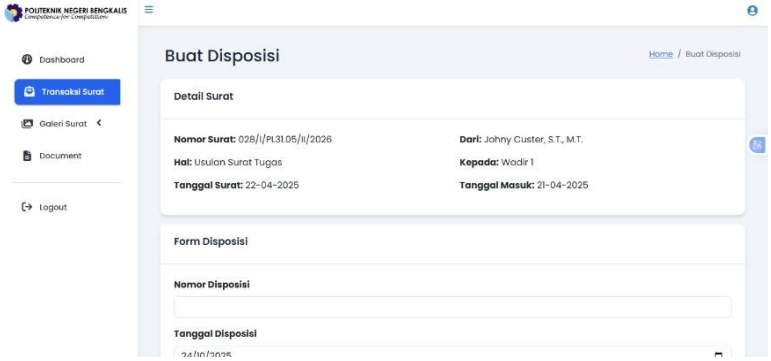

Figure 12. Letter Disposition Recording Interface

3. Construction

In the Construction phase, the approved prototype is transformed into a fully functional application. Development leverages PHP and JavaScript to implement business logic, user interfaces, and integrations with supporting components. Core security mechanisms are embedded systematically at this stage, including login authentication to ensure only authorized users can access the system, secure session management to maintain identity consistency and prevent misuse (e.g., session expiration/timeout), and role-based access control (RBAC) so that each function and dataset is accessible strictly according to user authorization. These controls are woven directly into the application flow, ranging from protected endpoints and server-side validation to comprehensive audit logging, so the resulting system meets not only functional requirements but also the principles of confidentiality, integrity, and availability.

4. Cutover

A comprehensive system test is conducted using a black-box testing approach, which evaluates the application's external behavior against specifications without considering internal code structure. The tests focus on the correctness of the input, process, output relationship, the clarity and consistency of the user interface, and overall ease of use for end users. Test scenarios cover normal, boundary, and error cases, ensuring that every user, facing function operates as required and is ready for production deployment.

5. CONCLUSION

This study reaffirms that information technology plays a vital and far-reaching role in day to day activities, particularly in administrative letter management within higher education institutions. The conclusions are as follows:

a. The web based DTS serves as an online service platform for managing document workflows (create, send, receive, verify, dispose), enabling required information to be delivered faster**,** more accurately, and with status visibility**.**
b. DTS supports centralized processing and archiving of documents—covering unit/official data, users and their roles, letters/memos and attachments, and disposition history—thereby facilitating traceability and accountability.
c. Implementing DTS streamlines previously manual processes (Excel files/email) into a structured flow with RBAC and validation, reducing errors, accelerating service delivery, and strengthening information security.
d. The letter tracking feature provides real time statuses (e.g., Sent, Received, Verified, Disposed), automatic cross role notifications, a unique document ID for rapid lookup, and a complete audit trail (timestamp, actor, action). These tracking data enable SLA monitoring and turnaround-time analysis, allowing administrators to identify bottlenecks and improve process efficiency.

.

6. REFERENCES

- Amalina, N. D., &Rachmawati, E. P. (2025). PenerapanMetode Design Thinking Dalam Perancangan Ulang UI & UX Aplikasi iPusnas. *Jurnal Informatika Polnema, 11(3).*
- Bestin, B., Pratama. M. A., Fadillah, R., &Dzakirin, D. J. (2024). Perancangan UI/UX pada Website Waroung Rai Raka Untuk Meningkatkan Pengalaman Pengguna. *Jurnal Dinamika Informatika*, 13(1).
- Chandra, A. F. M. (2022). Penerapan Metode Design Thinking dalam Rancang Prototipe Aplikasi Berbasis Web Peminjaman Dokumen Arsip di Dinas Komunikasi dan Informatika Provinsi Jawa Timur. *Jurnal Penelitian Administrasi Publik*, 2(4).
- Cyber Security Malaysia. (2020). MyVAC3GUI2SSDLCv1: Guidelines for Secure Software Development Life Cycle. Cyber Security Malaysia.
- Firdaus, M. (2024). Penerapan Metode Design Thinking Dalam Perancangan UI/UX pada dekontruksi Aplikasi Komunitas Developer Perumahan. *Jurnal Ilmiah Computing Insight*, 6(2).
- Hidayatullah, S & Patyani, E. (2024). Metode Rapid Application Development Dalam Pengembangan sistem Informasi Perpustakaan Berbasis Web, *Computer Science*, 4(2)
- Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems: Managing the Digital Firm* (16th ed.). Pearson.
- Martin, J. (1991). *Rapid Application Development*. Macmillan.
- Menduca, F. & Sulianta, F. (2025), Pengembangan Aplikasi Document Management System Lembaga Pemasyarakatan Kelas IIA Banceuy Menggunakan Pendekatan Object Oriented Analysis and Design, *Indonesian Journal of Machine Learning and Computer Science*, 5, 141-149
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach (9th ed.)*. McGraw Hill Education.
- Purwati, N., Syukron, A., Muningsih, E., Akbar, D. F., Waspada, A. R., &Syahroni, M. A. G (2024). Design UI/UX Aplikasi Safe4C&W Menggunakan Metode User Centered Design (UCD). *Jurnal Informatika Manajemen dan Teknologi*, 26(2).
- Putra, H. (2023). Sistem Informasi Manajemen Surat Masuk dan Keluar (SIMMK) Responsif Berbasis Web Menggunakan Metode Design Thinking. *Bulletin of Computer Science Research*, 3(6).
- Putra, R. R., Yasin, V., Zulhalim, Yulianto, A. B. (2025), Design of document management System Application Using Web-Baseed Rapid Application Development Method for Monitoring Verbal at Provincial Agencies in The Special Capital Region Jakarta, *Jurnal Teknologi Informasi dan Komputer MH. Thamrin*, 11(1).

- Rahman, F. F. R., Rustiawan, A., Ramadhan, D. O., Dermawan, G., Saputra, H. N., &Muis, A. (2025). Perancangan UI/UX Sistem Manajemen Persuratan Menggunakan Metode Design Thinking. *Jurnal Processor*, 20(1).
- Sari, A. P., Haromayni, M. M. A., Purnomo, R. (2024), Implementasi Metode Rapid Application Development Pada Aplikasi Sistem Monitoring Santri berbasis Web, *Jurnal Pendidikan Teknologi Informasi,* 4(1).
- Setiawan, H., Hanaputra, R. R.,Anggara, C. R, Putra, I. G. H., Wijayati, R. A., Hindami, A. L. A. (2024), Rancang Bangun Secure Document Management System (DMS) Menggunakan Metode Agile-SSDLC, *INSERT: Information System and Engineering Technology Journal,* 5(1).
- Subhiyanto. (2023), Analisis Metode pengembangan Sistem Informasi Management : A systematic Literature review, *Journal of Islamic Buisness Management Studies*, 4(2).
- Tisna, D. R., Maharani, T., Nugroho, K. T., &Julianto, B. (2024). Perancangan user inteface (UI) pada aplikasi angkut rosok berbasis mobile menggunakan figma pada TPS 3R Sido makmur. JAMI: *Jurnal Ahli Muda Indonesia*, 5(2), 101–113.