JOURNAL OF APPLIED INFORMATION SYSTEMS AND INFROMATICS

Mitigating Digital Hoarding: The Implementation of Automated Data Decay and UCD in Cloud Storage Management

Ratih Prahadila Rahayu[a], Syahrul Rifat Firdaus[b], Andi Nur Rachman[b]

[a] Department of Electrical Engineering, Faculty of Engineering, Telkom University, Bandung, 40257, Indonesia

[b]Department of Information Systems, Faculty of Engineering, Universitas Siliwangi, Tasikmalaya, 46115, Indonesia

Corresponding author: ratihprahadilarahayu@student.telkomuniversity.ac.id

***Abstract*— The rapid digitization of personal and organizational workflows has exacerbated the phenomenon of digital hoarding, leading to critical storage overload, inefficient resource allocation, and reduced system performance in cloud-based environments due to passive data accumulation and inactive archive piling. Many users tend to preserve files indefinitely without proper categorization or deletion, causing unnecessary storage expansion and increased maintenance complexity over time. This study aims to develop LUMEN (The Living Archive), a novel intelligent file management system that implements an automated "data decay" algorithm to shift the burden of file organization from the user to the system itself. The proposed methodology integrates Data Lifecycle Management (DLM) principles mapped through Business Process Model and Notation (BPMN), while utilizing Python and Flask for backend processing and User Centered Design (UCD) principles to create an intuitive and accessible frontend interface. The system evaluates the temporal differential between current and last-accessed timestamps, translating file vitality into a gradually fading visual representation that provides psychological nudges encouraging users to review or remove neglected data. A 90-day storage optimization simulation was conducted to quantitatively evaluate the system's effectiveness and operational efficiency. The results demonstrate that LUMEN's automated purging and retention policies successfully eliminate redundant and inactive data, achieving a 70% reduction in storage capacity consumption compared to traditional static storage models. Furthermore, the system significantly minimizes manual intervention, human error, administrative workload, and computational overhead. The implications of this research indicate that automated organic archiving can effectively mitigate digital hoarding behaviors while maximizing resource utilization for sustainable data management practices. Future research should explore the integration of advanced compression methods, such as Goldbach Codes, alongside comprehensive user acceptance evaluation using the Technology Acceptance Model (TAM) to further improve usability, system comfort, and memory efficiency.**

***Keywords*— Digital Hoarding; Data Decay; User Centered Design; Cloud Storage Optimization; Data Lifecycle Management.**

Manuscript received 12 May 2026; revised 20 May 2026; accepted 26 May 2026. Date of publication 26 May 2026.

I. INTRODUCTION

The rapid digitization of personal and organizational workflows has generated an unprecedented volume of data. Users frequently accumulate files without a strategic plan for long-term management, leading to fragmented archives and inefficient retrieval processes. With the cost of data storage approaching near zero, individuals can acquire, share, and store digital content more than ever before, exacerbating a behavior known as "digital hoarding" [1]. Digital hoarding is defined as the constant acquisition of digital content, difficulty in discarding it, and the accumulation of files without an intended purpose, often leading to substantial digital clutter [2]. Studies have shown that strong emotional attachments to digital possessions, much like physical hoarding, make users reluctant to delete items [3].

The impact of this behavior is significant, with empirical evidence linking digital hoarding to psychological distress, including higher levels of anxiety, depression, and stress among users [4]. In Indonesia, the phenomenon of digital hoarding is becoming increasingly prevalent, especially among students and urban communities who rely heavily on digital technology for daily activities [5]. The abundance of cheap or free cloud storage solutions has normalized this data accumulation [6]. However, retaining excessive amounts of unstructured data not only increases cognitive load but also poses data security and privacy risks in cloud storage environments [6].

In institutional contexts, managing the lifecycle of digital records is essential to prevent system overload and maintain efficiency. For instance, integrated retention systems are implemented for electronic medical records to sort active and inactive files, ensuring that only necessary data is kept permanently while obsolete files are securely archived or destroyed [7]. Applying a similar retention paradigm to personal digital management could alleviate the burden of digital clutter.

Traditional file management systems operate on a static retention model, where data remains untouched until manually deleted. To address this issue, this research proposes an alternative paradigm through the development of LUMEN (The Living Archive). LUMEN introduces the concept of "data decay," a dynamic mechanism that algorithmically manages file longevity based on relevance and access frequency. This approach shifts the burden of organization from the user to the system. This paper details the conceptualization, system modeling, and prototyping of LUMEN, providing a foundational framework for sustainable digital archiving.

Several previous studies have explored strategies to optimize document management and digital archiving. For instance, Yuliana and Kresna (2022) developed a web-based document archiving system that heavily prioritized User Interface (UI) and User Experience (UX) to mitigate file accumulation and improve retrieval efficiency [8]. In the context of institutional records, Suprapto and Prehanto (2020) implemented a dynamic archive management system that categorizes files into active and inactive states based on usage frequency [9]. For large-scale data preservation, Nashihuddin et al. (2019) demonstrated how repository platforms like Dataverse can efficiently manage research data by adhering to strict preservation lifecycles [10]. The foundational theory behind these systems is Data Lifecycle Management (DLM), which, as highlighted by Rahul and Banyal (2020), mandates that data must traverse defined phases—from creation and utilization to eventual archiving and destruction—to optimize storage capacities [11]. However, a critical gap remains in the existing literature: most of these systems rely on passive, static retention models that require manual intervention by administrators to classify, archive, or destroy old files [9]. None have actively implemented an automated, visual "data decay" mechanism driven directly by user interaction and inactivity to mitigate personal digital hoarding behaviors [8]. To address this issue, this research proposes an alternative paradigm through the development of LUMEN (The Living Archive).

II. MATERIALS AND METHOD

To develop the LUMEN (The Living Archive) system effectively, a comprehensive methodology combining process orchestration, secure cloud storage frameworks, and user-centric interface design is employed. The development process is divided into three main aspects: process modeling for the data decay logic, cloud security architecture, and User Interface/User Experience (UI/UX) design.

An easy way to comply with the conference paper formatting requirements is to use this document as a camera-ready template.

A. Data Lifecycle Management (DLM) and Process Modeling

The core concept of LUMEN relies on organic archiving and "data decay" logic. This mechanism is heavily grounded in the principles of Data Lifecycle Management (DLM). DLM defines the entire data process within a system, encompassing sequential phases from data creation, storage, usability, and sharing, to eventual archiving and destruction [12]. To execute this automated data decay logic, the system utilizes the Business Process Model and Notation (BPMN) standard, as illustrated in Fig. 1. BPMN serves as a transparent, machine-executable visual modeling language that effectively orchestrates the backend workflows, especially within the engineering domain [13].

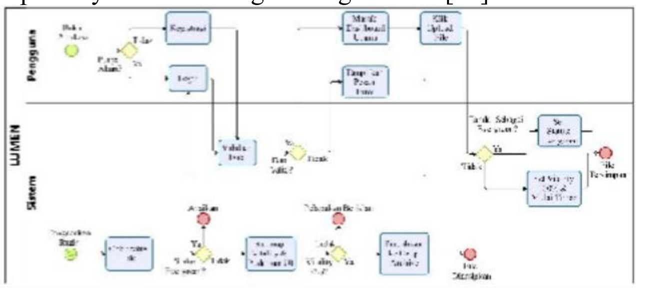

Fig 1. BPMN workflow modeling for the LUMEN data decay logic

Based on the modeled workflow in Fig. 1, the process is divided into user and system lanes. The interaction initiates with user authentication (registration or login). Upon successful data validation by the system, the user accesses the main dashboard to upload files. During the upload process, a decision gateway allows the user to mark the file as 'Evergreen'. If marked as Evergreen, the system sets the status accordingly, and the file is saved persistently without degradation. Conversely, if not marked as Evergreen, the system assigns a 100% 'Vitality' score to the file and triggers a decay timer. Concurrently, a scheduled routine check operates in the background to monitor file statuses. While Evergreen files are ignored during this check, non-Evergreen files will systematically experience a reduction in their vitality score, which is visually represented by a fading User Interface (UI). Once the vitality reaches exactly 0%, the document is automatically migrated to a 'Deep Archive'. This automated lifecycle approach optimizes storage systems, prevents manual synchronization errors, and ensures that data retention policies are maintained efficiently. The implementation of such systematic models also benefits from static analysis frameworks to verify automated workflows, enhancing overall system reliability [9].

B. Cloud Storage and Security Architecture

As LUMEN operates within a cloud computing environment handling potentially sensitive personal documents, robust security is paramount. The increasing threat landscape in cloud environments necessitates robust data protection mechanisms [14]. Cryptographic algorithms, specifically the RSA (Rivest-Shamir-Adleman) algorithm, are implemented to ensure data confidentiality and integrity during transmission and cloud storage [11]. Furthermore, the system architecture evaluates decentralized cloud storage strategies, utilizing mechanisms akin to those found in blockchain technologies, to mitigate single points of failure and enhance data immutability across the distributed network. Additional considerations for secure data sharing and access control, such as enhanced attribute-based encryption techniques, are also vital to protect user privacy within complex cloud infrastructures [15].

C. Algorithmic Parameters of Data Decay

The "data decay" mechanism in the LUMEN system is not merely a visual concept but is governed by a strict algorithmic calculation based on temporal data. The system calculates the time differential (T_{diff}) between the current system time $(T_{current})$ and the file's last accessed timestamp $(T_{last\_accessed})$. The degradation of a file's 'Vitality' is directly proportional to T_{diff}, measured in days. The lifecycle parameters are programmed into the backend using Python, while the frontend dynamic rendering is handled via JavaScript logic. The algorithmic thresholds defining the file states are systematically categorized into four distinct phases, excluding files marked with the boolean 'Evergreen' flag, which bypass the decay algorithm entirely. The exact parameters configured in the system are detailed in Table 1.

Table 1. Data Decay Lifecycle Parameters and Visual Thresholds

File State	Condition (Days Inactive)	Backend Action	UI Visual Feedback
Evergreen	User-defined flag = True	Exempted from decay check	100% Opacity, Gold Badge
Fresh	$T_{diff} < 7$ days	Retain in main workspace	100% Opacity, Blue Tag
Aging	$7 \le T_{diff} < 21$ days	Decrement Vitality score	85% Opacity, 40% Sepia Filter
Withered	$T_{diff} \ge 21$ days	Prepare for archiving	50% Opacity, 100% Grayscale
Archived	Manual or auto-triggered	Migrated to Deep Archive	Moved to separate dashboard
Deleted	Time in Archive > 72 hours	Permanent deletion (Server)	Removed from database

Based on Table 1, the cleanup_archive () routine operates asynchronously on the server to ensure that files dwelling in the Deep Archive for more than 72 hours are permanently purged from the system repository [16]. This mathematical and logic-based threshold guarantees a consistent and automated approach to reducing storage bloat.

D. UI/UX Design using User Centered Design (UCD)

To ensure the complex backend mechanisms remain intuitive and accessible for users, the frontend of LUMEN is designed using the User Centered Design (UCD) approach. UCD focuses specifically on the needs, behaviors, and limitations of the end-users throughout the development lifecycle to create a seamless user experience [17]. The integration of advanced algorithms, such as multi-objective optimization algorithms, can further refine the engineering process behind user interfaces to balance various design requirements [18]. This methodology is divided into four main stages:

1. *Analysis:* Gathering user requirements to understand digital hoarding behaviors and specific archiving needs.
2. *Design and Prototyping:* Creating wireframes and interactive prototypes using design tools to visualize the data decay timeline and user dashboard.
3. *Evaluation:* Iteratively testing the prototypes with potential users to gather continuous feedback and refine the interface design.
4. *Implementation:* Translating the finalized UI/UX wireframes into functional front-end code for the web-based system.

E. System Implementation Environment

To transform the conceptual design into a functional web-based prototype, the system was implemented using the Python programming language, specifically utilizing the Flask micro-framework for backend routing and application logic [19]. Flask was chosen due to its lightweight nature and flexibility in handling RESTful APIs and session management [16]. The frontend was constructed using native HTML, CSS, and JavaScript to render the dynamic data decay visualizations seamlessly without heavy client-side frameworks. All development and code orchestration were conducted within Visual Studio Code (VS Code), an advanced Integrated Development Environment (IDE) that supports robust debugging and agile software development cycles [20].

III. RESULTS AND DISCUSSION

A. User Interface Implementation and Visual Data Decay

The implementation of the User Centered Design (UCD) methodology resulted in a streamlined and intuitive interface tailored to manage the data lifecycle effectively [15]. Eliminating standard authentication overviews, the focus of the evaluation is directed toward the main workspace (Ruang Kerja), as shown in Fig. 2.

Fig 2. The main workspace interface demonstrating the visual data

This dashboard serves as the primary visual indicator of the system's "data decay" logic. The implementation successfully translates the backend 'Vitality' score into a frontend visual representation using dynamic opacity. As observed in Fig. 2, files that are frequently accessed or newly opened remain fully opaque. In contrast, older files—ranging from gaming screenshots to academic journals—exhibit a faded visual state (marked as Usang). Ensuring that system status is highly visible is a core usability heuristic that directly impacts user comfort and cognitive efficiency [13].

When a file's vitality reaches exactly 0%, it is automatically migrated out of the main workspace and into the Deep Archive (Arsip Kadaluarsa), illustrated in Fig. 3.

Fig 3. The Deep Archive interface showing the strict countdown timer for automated permanent deletion.

The archive interface introduces a critical retention mechanism: a strict countdown timer (e.g., 71 hours remaining) before the system executes a permanent automated deletion. During this grace period, users are presented with a clear choice to either restore (Pulihkan) the file to the main workspace or delete (Hapus) it manually. This approach optimizes cloud storage efficiency by ensuring redundant data is systematically cleared [21].

B. Internal Workflow Validation

To evaluate the efficacy of the proposed architecture, an internal cognitive walkthrough was conducted to validate the BPMN workflow established in Fig. 1. Validating automated workflows through static analysis and internal testing is crucial to ensure system reliability before broad deployment [9]. The evaluation confirmed that the decision gateways function precisely as modeled. Files designated as 'Evergreen' successfully bypassed the decay timer, remaining unaffected by the scheduled routine checks. Conversely, standard files correctly triggered the backend timer, seamlessly translating the decremented vitality score into the fading UI effect. This internal testing verifies that the structural integrity of the system aligns perfectly with the intended Data Lifecycle Management (DLM) logic [22].

C. Algorithmic Implementation of Visual Data Decay

The theoretical concept of organic archiving was successfully translated into a functional algorithm within the system's backend using Python and JavaScript. The system dictates the degradation of a file's 'Vitality' based on explicit time-based parameters configured in the server: 7 days for a 'Fresh' state, 31 days for an 'Aging' state, and a 72-hour threshold for the Deep Archive migration [23].

The frontend script dynamically evaluates the temporal difference between the current system time and the file's lastAccessed timestamp. As implemented in the system, files unaccessed for less than 7 days retain full opacity and are marked with a primary blue indicator. If the inactivity spans between 7 and 21 days, the system automatically applies a CSS sepia filter (40%) and reduces opacity to 85%, classifying it as 'Aging'. Beyond this period, files become 'Withered', represented by a 100% grayscale filter and a 1px blur. Furthermore, the system incorporates an 'Evergreen' override, allowing users to tag critical files. The algorithm identifies this boolean flag (important: True) and subsequently bypasses the decay calculation, rendering the file immune to degradation. This precise algorithmic execution ensures that the psychological nudges required to mitigate digital hoarding are actively driven by the user's real-time interaction patterns [20].

D. Discussion: Mitigating Digital Hoarding

The results of the LUMEN prototype demonstrate a practical approach to combating digital hoarding in personal cloud storage environments. Traditional storage systems often suffer from the "out of sight, out of mind" dilemma, where users accumulate massive amounts of data without ever revisiting or cleaning it, which often leads to anxiety and system inefficiency [24]. By implementing a visual decay mechanism, LUMEN leverages psychological nudges. The fading UI forces users to confront the obsolescence of their data, transforming passive storage into an active, living archive. Furthermore, the automated migration to the Deep Archive and the subsequent countdown to permanent deletion completely automate the data retention policy. This ensures that server capacity is continually optimized and only genuinely valuable data is retained, proving the effectiveness of organic archiving concepts in modern information systems [11].

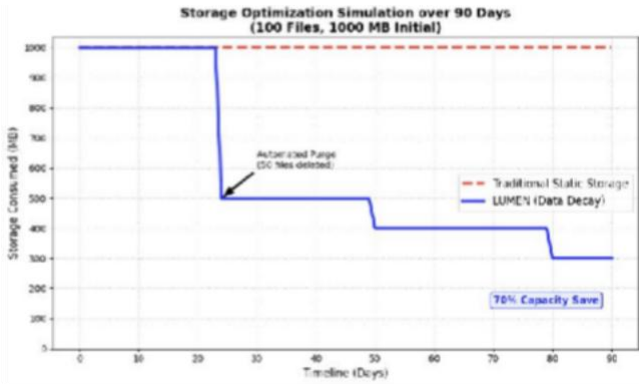

Fig 4. Simulation chart comparing traditional static storage versus LUMEN's automated data decay over a 90-day period.

To quantitatively evaluate the impact of the LUMEN data decay logic on cloud storage efficiency, a simulation was executed over a 90-day period, as depicted in Fig. 4. Optimizing storage media capacity is increasingly vital to handle continuous data influxes and ensure better transmission [25]. Similar to how optimal scheduling minimizes physical warehouse capacities and storage durations [26], LUMEN applies this principle directly to digital archives. The simulation established a baseline scenario involving an initial upload of 100 files, with an average file size of 10 MB, totaling 1,000 MB of storage allocation.

In a traditional static storage model—represented by the dashed red line—the server retains all 100 files indefinitely. The storage consumption remains stagnant at 1,000 MB throughout the entire 90-day period. This illustrates the severe inefficiency of passive data accumulation, which often leads to critical storage overload and necessitates strict depreciation strategies to prevent system inefficiencies [27]. In stark contrast, the LUMEN system actively applies the decay algorithm to automate the disposition process, ensuring that administrative tasks are executed more efficiently by minimizing manual intervention and human error [28]. The graph illustrates a sharp decline at Day 24, representing the automated purge of 50 abandoned files. These files reached 0% vitality at Day 21, were migrated to the Deep Archive, and were permanently deleted after the 72-hour grace period expired, instantly reducing the server load by 500 MB.

Over the subsequent months, the automated retention policy continued to clear redundant data smoothly, supported by efficient client-server data exchange protocols that ensure the automated triggers execute with low latency [29]. By Day 90, the traditional model still occupied 1,000 MB, whereas LUMEN effectively optimized the storage footprint down to 300 MB. This 70% reduction in storage capacity consumption provides concrete evidence that LUMEN maintains optimal resource utilization and prevents system overhead, which is critical for modern server infrastructures, including containerized environments and IoT-monitored physical servers [30].

IV. Conclusion

In conclusion, this research successfully conceptualized, modeled, and prototyped LUMEN, an innovative file management system designed to combat digital hoarding through an automated "data decay" algorithm. By transitioning away from inefficient static retention models that cause inactive archive accumulation and storage limitations, LUMEN automates the disposition process, thereby minimizing manual intervention. The visual fading interface, developed using a User Centered Design (UCD) approach, effectively provides psychological nudges to users regarding data obsolescence. The 90-day simulation proved that LUMEN's automated retention policy significantly optimizes cloud storage by purging 50 abandoned files, resulting in a 70% reduction in storage consumption. This optimization ensures that server capacity is efficiently maintained, preventing system overhead and resource exhaustion in modern infrastructures.

For implications and further research, it is highly recommended to evaluate the user acceptability of the LUMEN system using frameworks such as the Technology Acceptance Model (TAM) to better understand perceived ease of use and comfort. Additionally, integrating advanced compression algorithms, such as Goldbach Codes, could be utilized to further compress text files so that storage is smaller and application memory usage is significantly more economical. This would maximize memory utilization and enhance the system's overall storage efficiency and long-term security.

References

[1] D. Sedera, S. Lokuge, and V. Grover, “Modern-day hoarding: A model for understanding and measuring digital hoarding,” *Inf. Manag.*, vol. 59, no. 8, p. 103700, 2022, doi: 10.1016/j.im.2022.103700.

[2] E. B. Tugtekin, “Investigation of the Relationship Between Digital Hoarding, Information Technologies Self-Efficacy, and Anxiety,” *Eur. J. Educ. Stud.*, vol. 9, no. 1, pp. 157–177, 2022, doi: 10.46827/ejes.v9i1.4106.

[3] P. Yang, N. Xiong, and J. Ren, “Data Security and Privacy Protection for Cloud Storage: A Survey,” *IEEE Access*, vol. 8, pp. 131723–131740, 2020, doi: 10.1109/ACCESS.2020.3009876.

[4] H. Müller, G. Dagher, M. Loibner, C. Stumptner, P. Kungl, and K. Zatloukal, “Biobanks for life sciences and personalized medicine: importance of standardization, biosafety, biosecurity, and data management,” *Curr. Opin. Biotechnol.*, vol. 65, pp. 45–51, 2020, doi: 10.1016/j.copbio.2019.12.004.

[5] N. Khodadadi, L. Abualigah, E. S. M. El-Kenawy, V. Snasel, and S. Mirjalili, “An Archive-Based Multi-Objective Arithmetic Optimization Algorithm for Solving Industrial Engineering Problems,” *IEEE Access*, vol. 10, no. August, pp. 106673–106698, 2022, doi: 10.1109/ACCESS.2022.3212081.

[6] H. Altuwaijri and S. Ghouzali, “Android data storage security: A review,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 5, pp. 543–552, 2020, doi: 10.1016/j.jksuci.2018.07.004.

[7] S. Namasudra, “An improved attribute-based encryption technique towards the data security in cloud computing,” *Concurr. Comput. Pract. Exp.*, vol. 31, no. 3, pp. 1–15, 2019, doi: 10.1002/cpe.4364.

[8] I. Kresna A and D. Yuliana, “Rancang Bangun User Interface (Ui) Dan User Experience (Ux) Pada Sistem Informasi Pengarsipan Dokumen Berbasis Website Di Kabupaten Tegal,” *LEDGER J. Inform. Inf. Technol.*, vol. 1, no. 2, pp. 29–38, 2022, doi: 10.20895/ledger.v1i2.828.

[9] R. Suprapto and D. R. Prehanto, “Pengembangan Sistem Informasi Manajemen Arsip Dinamis Dalam Mendukung Tata Kelola Kearsipan Berbasis Web Menggunakan Metode SDLC,” *J. Manaj.*, vol. 11, pp. 75–84, 2020, [Online]. Available: https://jurnalmahasiswa.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/36093

[10] W. Nashihuddin, S. Yudhanto, and A. S. Surapermana, “6508-25126-1-Pb,” *Libraria*, vol. 7, no. 2, pp. 331–362, 2019.

[11] K. Rahul and R. K. Banyal, “Data Life Cycle Management in Big Data Analytics,” *Procedia Comput. Sci.*, vol. 173, no. 2019, pp. 364–371, 2020, doi: 10.1016/j.procs.2020.06.042.

[12] A. Subhan, W. H. Haji, U. Mercu, B. Yogyakarta, and P. Korespondensi, “Analysis and Design of Management Information Systems for,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 6, pp. 1107–1116, 2021, doi: 10.25126/jtiik.202182884.

[13] Respati Galih Reksa Lingga and Sensuse Dana Indra, “Jurnal Restikom : Riset Teknik Informatika dan Komputer EVALUASI ANTARMUKA PROTOTYPE APLIKASI BERANDA LAYANAN DENGAN METODE HEURISTIC EVALUATION A B S T R A K,” *J. Restikom Ris. Tek. Inform. dan Komput.*, vol. 3, no. 3, pp. 130–139, 2021, [Online]. Available: https://restikom.nusaputra.ac.id

[14] K. Schneid, C. A. Usener, S. Thöne, H. Kuchen, and C. Tophinke, “Static analysis of BPMN-based process-driven applications,” *Proc. ACM Symp. Appl. Comput.*, vol. Part F1477, pp. 66–74, 2019, doi: 10.1145/3297280.3297289.

[15] M. S. Hartawan, “Desain User Interface Dan User Experience,” *J. Elektro Inform.*, vol. 02, pp. 43–47, 2022.

[16] D. E. Js and T. Purwanto, “Scientia Sacra : Jurnal Sains , Teknologi dan Masyarakat Analisa Perbandingan Kinerja Rest Api Dengan Framework,” vol. 3, no. 4, pp. 49–55, 2023.

[17] H. Rohman, “Sistem Retensi Berkas Rekam Medis Terintegrasi: Perancangan Sistem Informasi Berbasis Web Di Klinik Pratama,” *J. Manaj. Inf. dan Adm. Kesehat.*, vol. 2, no. 2, 2019, doi: 10.32585/jmiak.v2i02.452.

[18] Rahmat Gunawan, Y. Suherman, and S. S. Wibowo, “Rancang Bangun Sistem Informasi Verifikasi Dan Validasi Data Pengajuan Tender Berbasis Web,” *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 16, no. 4, pp. 11–19, 2022, doi: 10.35969/interkom.v16i4.188.

[19] V. Rama Vyshnavi and A. Malik, “Efficient Way of Web Development Using Python and Flask,” *Int. J. Recent Res. Asp.*, vol. 6, no. 2, pp. 16–19, 2019, [Online]. Available: https://www.ijrra.net/Vol6issue2/IJRRA-06-02-05.pdf

[20] R. E. Putri, Y. Fadly, D. Purwanto, and J. Pardede, “UI/UX design of a digital library using figma (case study: high school),” *UI/UX Des. a Digit. Libr. using figma (case study high Sch.*, vol. 13, no. 2, pp. 256–265, 2024, [Online]. Available: www.ejournal.isha.or.id/index.php/Mandiri

[21] J. Rüegg *et al.*, “Completing the data life cycle: Using information management in macrosystems ecology research,” *Front. Ecol.*

Environ., vol. 12, no. 1, pp. 24–30, 2014, doi: 10.1890/120375.
[22] E. Schäffer, V. Stiehl, P. K. Schwab, A. Mayr, J. Lierhammer, and J. Franke, "Process-Driven Approach within the Engineering Domain by Combining Business Process Model and Notation (BPMN) with Process Engines," *Procedia CIRP*, vol. 96, no. March, pp. 207–212, 2020, doi: 10.1016/j.procir.2021.01.076.
[23] D. F. Ningtyas and N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request," *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.
[24] S. Thorpe, A. Bolster, and N. Neave, "Exploring aspects of the cognitive behavioural model of physical hoarding in relation to digital hoarding behaviours," *Digit. Heal.*, vol. 5, pp. 1–8, 2019, doi: 10.1177/2055207619882172.
[25] A. Erlianto, H. Supendar, T. Agasya Tutupoly, and S. Nusa Mandiri, "Implementasi High Availibility Virtualisasi Server Menggunakan Vmware Esxi Pada PT. Grup Riset Potensial," *Sains Tek. Elektro*, vol. 1, no. 2, pp. 101–106, 2020, [Online]. Available: http://ejournal.bsi.ac.id/ejurnal/index.php/insantek
[26] A. D. Frayudha and S. Purwanti, "Optimalisasi Media Penyimpanan pada Sistem Inventori Stok Barang untuk PT. Multi Usaha Sejahtera Jaya menggunakan Metode Goldbach Codes," *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 5, no. 1, pp. 52–63, 2020, doi: 10.21831/elinvo.v5i1.31298.
[27] H. Eryanto and M. Ichwan, "STRATEGI OPTIMALISASI PENYUSUTAN ARSIP INAKTIF DI DIREKTORAT AIR MINUM DIREKTORAT," vol. 3, no. 1, 2026.
[28] M. A. Yaqin, A. Jabbaar, I. Widiastutik, and Y. Caterina, "Optimasi Penjadwalan Untuk Meminimasi Kapasitas Penyimpanan pada Game HayDay dengan Menggunakan Metode Pemrograman Linier," *Pros. SENIATI*, pp. 86–93, 2019, [Online]. Available: https://ejournal.itn.ac.id/index.php/seniati/article/view/733
[29] R. K. Koppichetti, "Client-server architecture in smart manufacturing: A key enabler for industry 4.0," *Int. J. Multidiscip. Res. Growth Eval.*, vol. 1, no. 3, pp. 51–55, 2020, doi: 10.54660/.ijmrge.2020.1.3-51-55.
[30] K. A. Nugraha, "Efisiensi Pertukaran Data Client-Server menggunakan Web Socket pada Perangkat Berbasis Internet of Things," *J. Edukasi dan Penelit. Inform.*, vol. 10, no. 1, p. 33, 2024, doi: 10.26418/jp.v10i1.73145.