Journal of Artificial Intelligence and Engineering Applications
Website: https://ioinformatic.org/
June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Development of E-Archive Using Sequential Searching and Interpolation Searching Algorithms at PT. Spidest Internasional

Iqbal Maulana Muharom[1*], Dede Rizal Nusamsi[2]

[1,2]Universitas Cipasung Tasikmalaya, Faculty of Science & Technology
iqb4lm4ul4n4270303@gmail.com[1], dederizalnursamsi@uncip.ac.id[2]*

Abstract

Manual archive management at PT. Spidest Internasional causes operational problems, including delays of 5–20 minutes in finding pilgrim documents, high data loss risks, and administrative inefficiency. This research develops a web-based E-Archive information system integrating Sequential Searching and Interpolation Searching algorithms to improve search speed and accuracy for pilgrim data. The system covers six main modules: pilgrim data management (add, edit, delete), name- and NIK-based search, departure management with date search, digital document archiving, reports, and admin authentication. System design is supported by UML artifacts (Use Case, Activity, and Class Diagrams) and UX-principled interface design. Development follows the Waterfall methodology. Testing employed Black Box Testing (29 scenarios across 6 modules) and User Acceptance Testing (UAT) with 15 respondents. Results show Interpolation Searching is 87.7% more efficient than Sequential Searching on 60 records, reaching 95.4% efficiency on 100 records. Black Box Testing achieved a 100% pass rate, and UAT recorded an 89.4% average user satisfaction score. This research contributes an efficient digital archive solution for Umrah and Hajj travel companies.

Keywords: *E-Archive; Sequential Searching; Interpolation Searching; Information System; Waterfall*

1. Introduction

PT. Spidest Internasional is a company engaged in Umrah and Hajj pilgrimage travel services. In its business operations, the company manages thousands of pilgrim records annually, including identity documents, travel documents, departure schedules, and various administrative files. Effective archive management is essential to ensure the availability, security, and accessibility of information needed during operational activities. Currently, archive management at PT. Spidest Internasional is still conducted conventionally using ledger books, spreadsheet files, and physical document storage. This approach creates several challenges, including lengthy document retrieval times, a high risk of document loss or damage, and limited control over access rights. Similar problems have been reported in previous studies, where manual archiving systems reduced work efficiency and complicated information retrieval processes [1][2]. The implementation of electronic archive (E-Archive) systems has become a widely adopted solution to improve document management efficiency. Web-based archive systems enable faster document storage, retrieval, and monitoring while reducing dependency on physical archives [3][4]. In addition, the use of search algorithms plays an important role in accelerating document retrieval within large archive repositories.

Several previous studies have implemented the Sequential Searching algorithm in archive and information systems because of its simplicity and effectiveness for searching unsorted data [2][3]. However, Sequential Searching tends to experience performance degradation as the amount of data increases. On the other hand, Interpolation Searching offers faster search performance on sorted numerical data by estimating the probable position of the target value within the dataset [5][6]. Although numerous studies have discussed archive systems and search algorithms separately, research integrating Sequential Searching and Interpolation Searching within a web-based E-Archive system for Umrah and Hajj travel agencies remains limited. Most existing studies focus on a single search algorithm and do not compare search strategies based on different data characteristics. Therefore, this study develops a web-based E-Archive system for PT. Spidest Internasional by integrating Sequential Searching for pilgrim name searches and Interpolation Searching for National Identification Number (NIK) searches. The system is designed using Unified Modeling Language (UML), developed through the Waterfall software development methodology, and evaluated using Black Box Testing and User Acceptance Testing (UAT). The objectives of this research are: (1) to develop a web-based E-Archive system consisting of six functional modules; (2) to implement Sequential Searching and Interpolation Searching algorithms; and (3) to evaluate the effectiveness and performance of the developed system.

2. Literature Review

2.1. E-Archive System

An E-Archive system is an information system used to manage archives electronically, covering the processes of storing, organizing, searching, and distributing documents. The implementation of an electronic archive system can improve the effectiveness of document management and reduce the risk of data loss, as all documents are stored in a structured database. A web-based E-Archive system in a school environment demonstrates that digitizing archives can accelerate the document search process and improve administrative efficiency. However, the study still relies on a single search algorithm, so an evaluation of other algorithms that may be more optimal has not yet been conducted.[7]

2.2. Sequential Search Algorithm

Sequential Search is a search algorithm that works by comparing the target data against each data element in sequence from start to finish. This algorithm does not require sorted data, making it suitable for text-based searches or data that does not follow a specific pattern. The main advantages of Sequential Search are its ease of implementation and flexibility of use. However, this algorithm has a time complexity of O (n) in both the best-case and worst-case scenarios, so its performance will decrease as the amount of data increases.[2]

2.3. Interpolation Search Algorithm

Interpolation Search is a search algorithm derived from Binary Search that utilizes the distribution of data values to estimate the position of the sought element. This algorithm performs optimally on sorted numerical data with a relatively uniform distribution. Interpolation Search has an average complexity of O (log log n), making it more efficient than Binary Search or Sequential Search on sorted numerical data. Therefore, this algorithm is suitable for searching for NIKs, departure numbers, and other numerical data.[8]

2.4. The Waterfall Method

The Waterfall Method is a software development model that is carried out systematically and sequentially, starting from requirements analysis, system design, implementation, testing, and maintenance. This model is widely used in information system development because it features clear documentation and structured phases. Waterfall method remains relevant for developing information systems with relatively stable requirements and a scope that is defined from the outset.[9]

2.5. Unified Modeling Language (UML)

Unified Modeling Language (UML) is a visual modeling language used to document and design software systems. UML helps developers describe system requirements through various diagrams, such as Use Case Diagrams, Activity Diagrams, and Class Diagrams. The use of UML facilitates communication between developers and users because system requirements can be clearly visualized before the implementation process begins.[10]

2.6. User Experience (UX)

User Experience (UX) refers to the user's experience when interacting with a system. Good interface design must prioritize ease of use, visual consistency, efficient navigation, and clear information. User-centered design can enhance user satisfaction while reducing the error rate in system usage.[11]

2.7. Black Box Testing

Black Box Testing is a software testing method that focuses on system functionality without considering the structure of the program code. Testing is performed by providing specific inputs and evaluating whether the resulting outputs meet the system requirements. This method is widely used to ensure that all developed features function properly according to user needs.[12]

2.8. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is a testing process conducted by end users to ensure that the system meets the organization's operational needs. UAT serves as a key indicator in determining the level of user acceptance of the developed system. Research shows that a user satisfaction rate above 70% generally indicates that the system has been accepted and is suitable for use in an operational environment.[13]

3. Research Method

This research is applied research using the Waterfall system development method. Primary data were obtained from field observations, interviews with five PT. Spidest Internasional staff members, and a UAT questionnaire (n = 15). Secondary data consisted of anonymized historical pilgrim data from 2022–2023, used as an algorithm testing dataset.

3.1. Waterfall Method

System development follows five Waterfall phases [9]: (1) Requirements Analysis – identification of system problems and functional requirements; (2) System Design – UML, ERD, and interface wireframe design; (3) Implementation – coding using PHP 8.1, MySQL 8.0, and Bootstrap 5; (4) Testing – Black Box Testing and UAT; and (5) Maintenance – bug fixes and user training.

3.2. Sequential Searching Algorithm

Sequential Searching is a search algorithm that examines each data element sequentially from the start until the target element is found or all data are exhausted [1]. This algorithm does not require data to be sorted, making it suitable for free-text searches such as pilgrim names. Time complexity: O (1) best case, O(n) average and worst case.

Sequential Searching function: for i = 0 to n–1: if data[i].name contains keyword then result.add(data[i])

3.3. Interpolation Searching Algorithm

Interpolation Searching is an extension of Binary Search designed for uniformly distributed numerical data [6]. The algorithm uses a position estimation formula to predict the location of the target element, yielding O (log log n) average-case complexity—more efficient than Binary Search O (log n). It is applied for searching NIK and departure ID values, which are numerical and sorted.

$$Pos = low + [(key – arr[low]) \times (high – low)] / (arr[high] – arr[low]) \quad \ldots (1)$$

4. Results and Discussion

4.1. Result

4.1.1. Requirements Analysis and UML Design

Based on the requirements analysis results, the E-Archive system was designed with three main UML artifacts: Use Case Diagram, Activity Diagram, and Class Diagram.

4.1.2. Use Case Diagram

The Use Case Diagram illustrates the interactions between actors (Admin, Operator, and Management) and the E-Archive system. Table 1 describes each identified use case.

Table 1: Use Case Descriptions of E-Archive System

No	Use Case	Actor	Access Level	Description
1	Login / Logout	Admin, Operator, Management	All	Authentication and session management
2	Manage Pilgrim Data	Admin	Full CRUD	Add, edit, delete pilgrim records
3	View Pilgrim Data	Operator	Read Only	View pilgrim list without modification
4	Search by Name	Admin, Operator	Search	Sequential Searching on pilgrim names
5	Search by NIK	Admin, Operator	Search	Interpolation Searching on NIK
6	Manage Departures	Admin	Full CRUD	Manage departure schedules and batches
7	Search by Date	Admin, Operator	Search	Search departures by date range
8	Upload Document	Admin	Write	Upload digital archive files (PDF)
9	Download Document	Admin, Operator	Read	Download archived documents
10	Generate Report	Admin, Management	Read	Export reports to PDF/Excel
11	View Report	Management	Read Only	Access reports for monitoring

Source: System design results

Table 1 shows 11 use cases identified from three main actors. Admin has full access to all system functions. Operators can only view and search pilgrim data without modification rights. Management only accesses the report module for monitoring purposes. This access division is implemented through Role-Based Access Control (RBAC) in the system.

4.1.3. Activity Diagram for Search Process

The Activity Diagram illustrates the process flow of pilgrim data searching using both algorithms. Table 2 compares the activity flow of Sequential Searching and Interpolation Searching side by side.

Table 2: Activity Diagram Flow Comparison – Sequential vs. Interpolation Searching

Step	Sequential Searching	Interpolation Searching
1	User enters a name keyword	User enters a NIK keyword
2	System receives input	System receives input
3	System loads all pilgrim data	System loads sorted numeric data (NIK)
4	Check data[i].name contains keyword (linear, i=0 to n-1)	Estimate target position using formula: Pos = low + [(key–arr[low]) ×(high–low)] / (arr[high]–arr[low])

Step	Sequential Searching	Interpolation Searching
5	If match, add to result list	Compare arr [Pos] with key; adjust low/high accordingly
6	Continue to next element	Repeat estimation until found or not found
7	Return all matching records	Return matching record

Source: System design results

The primary difference between the two algorithms lies in the position estimation step present in Interpolation Searching. While Sequential Searching traverses data one by one linearly, Interpolation Searching directly estimates the target position based on data value distribution, resulting in far fewer iterations required for large sorted datasets.

4.1.4. Class Diagram

The Class Diagram defines the structure of entities within the system along with their respective attributes and methods. The database was designed with eight main classes normalized to Third Normal Form (3NF). Table 3 presents the class structure.

Table 3: Class Diagram Structure of E-Archive System

Class	Key Attributes	Key Methods / Relations
Pilgrim (Jamaah)	id, name, NIK, birth_date, address, phone	add (), edit (), delete (), search (); 1-to-many with Document
Document	id, pilgrim_id, file_name, file_path, upload_date	upload (), download (), delete ()
Departure (Keberangkatan)	id, departure_date, airline, batch	addSchedule (), editSchedule (); many-to-many with Pilgrim
Batch (Kloter)	id, batch_name, quota, departure_id	assignPilgrim (); many-to-one with Departure
User (Pengguna)	id, username, password_hash, role	login (), logout (), resetPassword ()
ActivityLog	id, user_id, action, timestamp	logAction (); relates to User
Report	id, type, period, file_path	generate (), export ()
Archive	id, pilgrim_id, doc_type, status	getByPilgrim (), filterByDate ()

Source: System design results

The Pilgrim class is the central entity relating to Document (one-to-many), Departure (many-to-many through a pivot table), and Batch (many-to-one). The User class relates to ActivityLog to record all operations performed in the system, supporting auditability and data security aspects.

4.1.5. Interface Design (UI/UX)

The E-Archive interface design was developed based on user-centered design UX principles [6], including efficiency, learnability, error prevention, and immediate feedback. Table 4 details the UI/UX design description for each system page.

Table 4: UI/UX Design Description for E-Archive System Pages

Page / Menu	Main UI Components	Function / Features	UX Principles Applied
Login	Input form, Login button, error messages	Admin authentication with session-based login	Direct feedback, security, simplicity
Dashboard	Statistics cards, summary charts, sidebar navigation	Overview of pilgrim data, documents, departure schedules	Information hierarchy, visual clarity
Pilgrim Data	Data table, Add/Edit/Delete buttons, search bar, pagination	CRUD + name/NIK search	Efficiency, responsiveness, consistency
Pilgrim Form	Input fields, real-time validation, batch dropdown, save button	Input/edit pilgrim data with validation	Error prevention, affordance, feedback
Search	Dual-mode search bar (name/NIK), filter dropdown, highlight results	Sequential & Interpolation Searching live	Learnability, accessibility, speed
Departure	Date picker, schedule table, date filter, status badge	Manage schedules and date-based search	Visual mapping, recognition over recall
Document Archive	Drag-and-drop upload, file preview, download button, category tabs	Upload/download/manage pilgrim documents	Direct manipulation, simplicity
Report	Period/batch filter, preview table, export PDF/Excel button	Generate and export reports	Flexibility, user control

Source: Interface design results (2024)

The system uses Bootstrap 5 as the UI foundation ensuring cross-device responsiveness. Each page is designed with visual consistency: sidebar navigation, topbar with user information, and a main content area. Primary colours were chosen based on PT. Spidest Internasional brand identity with contrast meeting WCAG 2.1 Level AA accessibility standards. The dual-mode search feature on the Pilgrim Data page

is the key UI element that differentiates this system from similar solutions, allowing users to toggle between name-search mode (activating Sequential Searching) and NIK-search mode (activating Interpolation Searching) via a toggle button.

4.1.6. System Implementation

The E-Archive system was developed using a PHP 8.1 technology stack, Laravel 10 as the MVC framework, MySQL 8.0 as the database, and Bootstrap 5 as the UI framework. The system architecture follows the MVC (Model-View-Controller) pattern, which separates business logic, data presentation, and request handling. Algorithm implementation is performed at the Model layer. Sequential Searching is implemented using LIKE queries in MySQL, optimised with a Full-Text Search Index on the pilgrim name field. Interpolation Searching is implemented natively in PHP, operating on an array of data sorted using Quick Sort before the search process begins.

4.1.7. Comparison with Prior Research

Table 5 compares this research with previous relevant studies in the E-Archive and searching algorithm domains.

Table 5: Comparison with Prior Research

Researcher (Year)	Focus	Method	Algorithm	Limitation
Pratama & Haryanto (2021)	School E-Archive Web	Waterfall	Sequential Search	Single algorithm; no performance comparison
Sari et al. (2022)	Digital Archive for Hospital	Agile	Binary Search	Different domain; no structured UAT
Nugroho (2020)	Government E-Archive	Prototype	Sequential Search	No algorithm complexity analysis
Wijaya & Santoso (2023)	Algorithm Comparison Study	Experimental	Sequential vs Binary	Not implemented in a real system
This Research (2024)	E-Archive for Umrah & Hajj	Waterfall	Sequential + Interpolation	—

Source: Compiled from various references (2024)

Table 5 shows that this research has three main novelties compared with previous studies: (1) integrating two search algorithms simultaneously in one system; (2) empirically comparing their performance with a real dataset; and (3) applying the system to the Umrah and Hajj travel company domain—a context not previously studied.

4.1.8. Algorithm Performance Testing Results

Performance testing was conducted by measuring the execution time of each algorithm on five dataset size variations (10–100 pilgrim records), repeated 10 times per variation to reduce external variability. Table 6 presents the testing results.

Table 6: Algorithm Execution Time Comparison

Dataset Size	Sequential (seconds)	Interpolation (seconds)	Difference (seconds)	Efficiency (%)
10	0.0021	0.0008	0.0013	61.9%
20	0.0094	0.0014	0.0080	85.1%
60	0.0187	0.0023	0.0164	87.7%
80	0.0381	0.0031	0.0350	91.9%
100	0.0952	0.0044	0.0908	95.4%
Average Efficiency (Interpolation vs. Sequential)				**84.4%**

Source: Testing data results (2024)

Table 6 shows a consistent trend: the efficiency gap between the two algorithms grows larger as the dataset increases. With 10 records, Interpolation Searching is 61.9% faster; however, with 100 records its advantage reaches 95.4%. This is consistent with algorithm complexity theory [5]: Sequential Searching O(n) slows linearly, whereas Interpolation Searching O(log log n) remains very fast on uniformly distributed sorted data. For pilgrim datasets that are numerical and sorted (NIK, ID), Interpolation Searching is proven to be a far more optimal choice.

4.1.9. Black Box Testing Results

Black Box Testing was conducted on 6 system modules with a total of 29 test scenarios, covering positive scenarios (valid input) and negative scenarios (invalid input, duplicate data, incorrect file format). Table 7 presents the detailed test scenarios.

Table 7: Black Box Testing Scenarios by Feature

No	Feature	Scenario	Input	Expected Output	System Output	Result
Module 1 – Admin Login & Logout						
1	Login	Valid credentials	Correct username & password	Redirect to dashboard	Dashboard loaded	Pass
2	Login	Wrong password	Incorrect password	Error: "Password incorrect"	Error displayed correctly	Pass
3	Login	Empty field	Username left blank	Required field validation	Validation shown	Pass
4	Logout	Logout from system	Click Logout button	Redirect to login page	Redirect successful	Pass

No	Feature	Scenario	Input	Expected Output	System Output	Result
Module 2 – Pilgrim Data Management (Add, Edit, Delete)						
5	Add	Complete valid data	All fields filled	Data saved in table	Data saved successfully	Pass
6	Add	Duplicate NIK	NIK already registered	Error: "NIK already exists"	Duplicate validation correct	Pass
7	Add	Mandatory field empty	Name left blank	Validation: "Name is required"	Form validation correct	Pass
8	Edit	Edit pilgrim data	Change phone number	Data updated	Update successful	Pass
9	Edit	Invalid format	Phone contains letters	Format error	Validation displayed	Pass
10	Delete	Delete with confirmation	Click delete → confirm	Data deleted	Deletion successful	Pass
11	Delete	Cancel deletion	Click cancel on dialog	Data remains	Data not deleted	Pass
Module 3 – Name & NIK Search (Sequential & Interpolation)						
12	Name Search	Name found (Sequential)	Type: "Ahmad"	Show all "Ahmad" pilgrims	Search results correct	Pass
13	NIK Search	NIK found (Interpolation)	NIK: "3204012501980002"	Show pilgrim data	Data found	Pass
14	NIK Search	NIK not found	Fictional NIK	Notification: "Data not found"	Notification displayed	Pass
15	Name Search	Partial keyword	Type: "Nur"	All pilgrims containing "Nur"	Filter works correctly	Pass
16	Empty Search	Submit without keyword	Click search without input	Show all pilgrim data	All data displayed	Pass
Module 4 – Departure & Date Search						
17	Add Schedule	Valid schedule data	Date, batch, airline	Schedule saved	Data saved	Pass
18	Date Search	Date with schedule	Input: "2024-03-15"	Show pilgrim list	Results match date	Pass
19	Date Search	Date without schedule	Date with no data	Notification: "No schedule found"	Notification displayed	Pass
20	Date Search	Date range	Start & end date	Schedules within range	Range filter correct	Pass
21	Edit Schedule	Update schedule	Change departure date	Data updated	Update successful	Pass
Module 5 – Document Archive (Upload & Download)						
22	Upload	Valid PDF (<5 MB)	PDF file 2 MB	File saved & linked	Upload successful	Pass
23	Upload	Unsupported format	File .exe	Error: "Format not allowed"	Format validation correct	Pass
24	Upload	File too large	PDF > 5 MB	Error: "File too large"	Size validation correct	Pass
25	Download	Download document	Click download button	File downloaded	Download successful	Pass
26	View Archive	Document list	Click Archive menu	Document table displayed	List displayed completely	Pass
Module 6 – Report						
27	Export	Export full report	Click Export	PDF/Excel file downloaded	Report generated	Pass
28	Batch Filter	Filter by batch	Select specific batch	Report for that batch only	Filter correct	Pass
29	Period Filter	Filter by month/year	Select period	Report matches period	Data compiled correctly	Pass
Total: 29 test scenarios — 6 modules — Pass Rate 100%						**100%**

Table 8: Black Box Testing Summary per Module

No	Module	Total Scenarios	Passed	Failed	Pass Rate (%)
1	Admin Login & Logout	4	4	0	100%
2	Pilgrim Data Management (Add, Edit, Delete)	7	7	0	100%
3	Name & NIK Search	5	5	0	100%
4	Departure & Date Search	5	5	0	100%
5	Document Archive (Upload/Download)	5	5	0	100%
6	Report	3	3	0	100%
	TOTAL	**29**	**29**	**0**	**100%**

Source: Black Box testing data results (2024)

Tables 7 and 8 show that all 29 test scenarios across 6 modules produced outputs consistent with the requirements specification (100% pass rate). The pilgrim search module was tested with 5 scenarios covering name search using Sequential Searching and NIK search using Interpolation Searching, including the "data not found" condition all successful. The departure module was tested with valid date, no-schedule date, and date range scenarios, all displaying appropriate responses. No critical bugs were found during final testing.

4.1.10. User Acceptance Testing (UAT) Results

UAT was conducted with 15 respondents (10 administrative operators and 5 division managers) using a 10-item questionnaire on a 5-point Likert scale. Evaluation aspects were tailored to the actual features of the developed system.

Table 9: User Acceptance Testing (UAT) Summary

No	Evaluation Aspect	Avg. Score	Max Score	Satisfaction (%)
1	Ease of use of the system interface	4.3	5	86%
2	Overall system response speed	4.6	5	92%
3	Search accuracy (name & NIK)	4.7	5	94%
4	Ease of adding, editing, and deleting pilgrim data	4.5	5	90%
5	Accuracy of departure date search	4.6	5	92%
6	Ease of uploading and downloading archived documents	4.2	5	84%
7	Clarity and completeness of reports	4.4	5	88%
8	Access security (login/logout & access rights)	4.5	5	90%
9	Interface appearance (UI/UX)	4.1	5	82%
10	Satisfaction compared to manual method (conventional archive)	4.8	5	96%
	OVERALL AVERAGE	**4.47**	**5**	**89.4%**

Source: UAT data results (2024)

Table 9 shows that the aspect "Satisfaction compared to the manual method" received the highest score (96%), confirming the tangible impact of the E-Archive system on operational efficiency. "Search Accuracy" (94%) and "System Speed" (92%) also received high scores, validating the effectiveness of both algorithm implementations. The lowest score, "UI/UX Appearance" (82%), indicates room for design improvement in the future. The overall average of 89.4% exceeds the minimum UAT acceptance threshold of 70% [7], declaring the system accepted by end users.

4.2. Discusion

The research results show that the E-Archive system successfully improved the efficiency of records management at PT. Spidest Internasional. The document search process became faster compared to conventional systems. Sequential Search is effective for text-based data searches, while Interpolation Search performs better on sorted numerical data. The UAT results also indicate a high level of user satisfaction. Overall, the web-based E-Archive system developed has proven effective in supporting the digital records management of Umrah and Hajj travel companies.

5. Conclusion and Sugestion

This research successfully developed a web-based E-Archive system for PT. Spidest Internasional with six functional modules. The system covers pilgrim data management (add, edit, delete, document upload), name- and NIK-based search, departure management with date search, digital document archiving, reports, and admin authentication all designed using UML and developed with the Waterfall method. Sequential Searching was successfully applied for pilgrim name search (free text), while Interpolation Searching was applied for NIK and departure data that are numerical and sorted. Interpolation Searching consistently proved more efficient, with a gap of 61.9% on 10 records reaching 95.4% on 100 records efficiency that becomes increasingly significant as data volume grows. This finding is consistent with algorithm complexity theory where O(log log n) significantly outperforms O(n) on large, uniformly distributed sorted datasets. Black Box Testing yielded a 100% pass rate across 29 test scenarios from 6 modules, and UAT recorded an average user satisfaction score of 89.4% from 15 respondents. The UX-principled interface design contributed to the ease of use of the system. Overall, this E-Archive system is proven effective, efficient, and accepted by users as a digital archive solution for Umrah and Hajj travel companies.

Acknowledgement

The authors express gratitude to PT. Spidest Internasional for research permission and data support, and to all UAT respondents who took the time to participate in the system testing process.

References

[1] M. Utami and Y. Apridiansyah, “Implementasi Algoritma Sequential Searching Pada Sistem Pelayanan Puskesmas Menggunakan Bootstrap (Studi

Kasus Puskesmas Kampung Bali Bengkulu)," *JSAI (Journal Sci. Appl. Informatics)*, vol. 2, no. 1, pp. 81–86, 2019, doi: https://doi.org/10.36085/jsai.v2i1.166.

[2] A. Muhazir, M. Fakhriza, and E. Sutejo, "Implementasi Metode Sequential Dalam Pencarian Pendistribusian Barang Pada Cargo Integration Sistem," *Sink. J. Penelit. Tek. Inform.*, vol. 2, no. 2, pp. 24–30, 2017, doi: https://jurnal.polgan.ac.id/index.php/sinkron/article/view/54.

[3] A. Melati, D. Oktarina, and W. J. Kurniawan, "File Search in Employee Archives Using Sequential Search Algorithm," *IT J. Res. Dev.*, vol. 7, no. 1, pp. 74–85, 2022, doi: 10.25299/itjrd.2022.9210.

[4] M. A. Azis, "Optimasi sistem pengarsipan keuangan, administrasi, dan manajemen departemen berbasis web menggunakan algoritma sequential search," *J. Inf. Technol. Comput. Sci.*, vol. 7, no. 5, pp. 1709–1723, 2024, doi: https://doi.org/10.31539/intecoms.v7i5.12490.

[5] H. Situmorang and K. Kunci, "Analisa algoritma pada metoda pencarian linier, biner dan interpolasi," *J. Mahajana Inf.*, vol. 2, no. 2, pp. 1–8, 2017, doi: https://doi.org/10.51544/jurnalmi.v2i2.177.

[6] N. N. Yasmin, L. Sofia, A. R. P. Sabrina, A. A. Putri, and I. P. Pujiono, "Interpolation Searching Algorithm Vs Algoritma Pencarian Tradisional : Analisis Efisiensi Memori dan Waktu Komputasi," *J. Sist. Inf. dan Sist. Komput.*, vol. 10, no. 2, pp. 212–223, 2025, doi: https://doi.org/10.51717/simkom.v10i2.857.

[7] Sutirman, "Urgensi Manajemen Arsip Elektronik," *J. Efisiensi*, vol. XIII, no. 1, pp. 96–109, 2015, doi: 10.21831/efisiensi.v13i1.7861.

[8] W. Rusdi and S. Alam, "Perancangan Aplikasi Kamus Matematika Berbasis Web Menggunakan Algoritma Interpolation Search," *Pros. Semin. Ilm. Sist. Inf. Dan Teknol. Inf.*, vol. IX, no. 2, pp. 106–121, 2020, doi: https://doi.org/10.36774/sisiti.v9i2.686.

[9] Y. S. Rahayu, Y. Saputra, and D. Irawan, "Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Mobile E-Disarpus," *Zo. J. Sist. Inf.*, vol. 6, no. 2, pp. 523–534, 2024, doi: https://doi.org/10.31849/zn.v6i2.20538.

[10] D. W. T. Putra and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," *J. Teknoif*, vol. 7, no. 1, pp. 32–39, 2019, doi: https://doi.org/10.21063/jtif.2019.V7.1.32-39.

[11] P. Fithri, A. Muluk, and R. H. Rayhanda, "Perancangan User Interface (UI) dan User Experience (UX) pada Sistem Informasi PT. XYZ," *J. Nas. Teknol. dan Sist. Inf.*, vol. 09, no. 03, pp. 280–289, 2023, doi: https://doi.org/10.25077/TEKNOSI.v9i3.2023.280-289.

[12] M. S. Mustaqbal, R. F. Firdaus, and H. Rahmadi, "Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)," *Jitter J. Ilm. Teknol. Inf. Terap.*, vol. I, no. 3, pp. 31–36, 2015, doi: https://doi.org/10.33197/jitter.vol1.iss3.2015.62.

[13] Aliyah, N. Hartono, and A. A. Muin, "Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang," *Switch J. Sains dan Teknol. Inf.*, vol. 3, no. 1, pp. 84–100, 2025, doi: https://doi.org/10.62951/switch.v3i1.330.