

Sistem Informasi Pengarsipan Berbasis Web Menggunakan Algoritma Levenshtein Distance

Achmad Noe'man^{1*}, Agus Hidayat², Arif Rifai Dwiyanto³, Mohammad Hoki Rezky⁴

Prodi Informatika, Universitas Bhayangkara Jakarta Raya^{1,2,3,4}

Email : achmad.noeman@dsn.ubharajaya.ac.id, agus.hidayat@dsn.ubharajaya.ac.id, arif.dwiyanto@ubharajaya.ac.id, 202010225110@mhs.ubharajaya.ac.id

ABSTRAKSI

Penelitian ini mengusulkan pengembangan sistem informasi pengarsipan berbasis web di PMI Kota Jakarta dengan dukungan algoritma Levenshtein Distance sebagai fitur utama pencarian dokumen. Sistem dirancang untuk mengatasi permasalahan pada metode pengarsipan manual, seperti keterbatasan dalam pencarian, potensi kesalahan pencatatan, serta risiko kehilangan arsip fisik. Dengan menerapkan pendekatan Waterfall, penelitian dilakukan melalui tahapan analisis kebutuhan, perancangan antarmuka, implementasi, dan pengujian. Hasil pengujian menunjukkan bahwa algoritma Levenshtein Distance mampu meningkatkan akurasi pencarian dokumen, meskipun terdapat perbedaan ejaan atau kesalahan pengetikan pada kata kunci. Hal ini menjadikan sistem lebih adaptif dan efisien dibandingkan metode manual. Selain itu, penerapan sistem berbasis web juga berkontribusi dalam mengurangi ketergantungan pada arsip fisik, meningkatkan keamanan penyimpanan, serta mempercepat proses akses informasi. Sistem yang dikembangkan memberikan solusi praktis dan modern untuk mendukung pengelolaan arsip di PMI Kota Jakarta.

Kata Kunci: Sistem Informasi Pengarsipan, Algoritma Levenshtein Distance, Pengelolaan Dokumen, Pencarian Dokumen

ABSTRACT

This study proposes the development of a web-based archiving information system at PMI Jakarta City, supported by the Levenshtein Distance algorithm as the core feature for document retrieval. The system is designed to overcome the limitations of manual archiving methods, such as restricted search capabilities, potential recording errors, and the risk of losing physical records. By adopting the Waterfall approach, the research was conducted through several stages: requirements analysis, interface design, implementation, and system testing. The results demonstrate that the Levenshtein Distance algorithm effectively improves the accuracy of document searches, even when spelling variations or typographical errors occur in the query. This makes the system more adaptive and efficient compared to traditional manual methods. Furthermore, the implementation of a web-based system reduces reliance on physical archives, enhances storage security, and accelerates information access. Overall, the developed system provides a practical and modern solution to support document management at PMI Jakarta City.

Keywords: Archiving Information System, Levenshtein Distance Algorithm, Document Management, Document Search

Penulis Korespondensi

Achmad Noeman

Tanggal Submit : 02/10/2025
Tanggal Diterima : 21/12/2025
Tanggal Terbit : 23/12/2025

This is an open access article under the [CC-BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license



Copyright: © 2023 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 (CC BY-NC-SA 4.0) International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Publisher's Note: JPPM stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

I. PENDAHULUAN

Seiring dengan pesatnya perkembangan teknologi informasi, pengelolaan arsip dalam berbagai organisasi, termasuk di Palang Merah Indonesia (PMI) Kota Jakarta, mengalami tantangan besar. Pengelolaan arsip

yang sebelumnya dilakukan secara manual dengan menggunakan dokumen fisik, sering kali menyebabkan kesulitan dalam pencarian dokumen, kesalahan pencatatan, serta risiko kehilangan dokumen akibat kerusakan atau bencana tak terduga. Di samping itu,

keterbatasan ruang penyimpanan dan efisiensi dalam pengelolaan dokumen juga menjadi masalah utama yang menghambat kelancaran operasional organisasi [1].

Pencarian dokumen yang akurat dan cepat menjadi kunci utama dalam meningkatkan produktivitas dan mengurangi ketergantungan pada dokumen fisik [2]. Salah satu pendekatan yang dapat digunakan untuk meningkatkan efisiensi pencarian dokumen adalah dengan menggunakan algoritma Levenshtein Distance, sebuah algoritma yang mengukur perbedaan antara dua string (kata) berdasarkan jumlah operasi yang diperlukan untuk mengubah satu string menjadi string lainnya. Algoritma ini sangat efektif untuk mendeteksi kesalahan pengetikan atau variasi ejaan dalam pencarian dokumen, sehingga dapat membantu memperbaiki akurasi pencarian di sistem pengarsipan berbasis web [3][4].

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem informasi pengarsipan berbasis web di PMI Kota Jakarta yang dilengkapi dengan fitur analisis dokumen menggunakan algoritma Levenshtein Distance. Sistem ini diharapkan dapat mempercepat proses pencarian dokumen, mengurangi ketergantungan pada dokumen fisik yang rentan terhadap kerusakan, serta meningkatkan efisiensi pengelolaan dokumen di lingkungan PMI Kota Jakarta. Dengan penerapan sistem berbasis web, diharapkan pengelolaan dokumen dapat dilakukan dengan lebih sistematis, aman, dan efisien [5].

Tujuan utama dari penelitian ini adalah untuk membangun sebuah sistem yang dapat mengoptimalkan pencarian dokumen dengan memanfaatkan teknologi dan algoritma yang tepat guna. Selain itu, penelitian ini juga bertujuan untuk menganalisis peran algoritma Levenshtein Distance dalam meningkatkan akurasi pencarian dokumen meskipun ada perbedaan kecil dalam penulisan atau kesalahan ketik. Dengan sistem ini, diharapkan PMI Kota Jakarta dapat lebih efektif dalam mengelola arsip dan mempermudah akses informasi yang dibutuhkan.

II. PENELITIAN YANG TERKAIT

Beberapa penelitian terdahulu telah mengembangkan sistem berbasis web untuk pengelolaan dokumen dengan menggunakan algoritma Levenshtein Distance. Dalam mengembangkan sistem pencarian surat berbasis web di Biro Teknologi Informasi Kepolisian menggunakan algoritma ini untuk meningkatkan efisiensi pencarian meskipun ada kesalahan pengetikan [6]. Penelitian serupa dilakukan oleh, yang menerapkan algoritma Levenshtein Distance pada aplikasi pengelolaan surat berbasis web untuk mempercepat proses pencarian dan mengurangi kesalahan pencatatan [7]. Kedua penelitian ini menunjukkan hasil yang positif dalam meningkatkan akurasi pencarian dokumen, yang juga menjadi tujuan utama dalam penelitian ini, meskipun konteks dan jenis

dokumen yang dikelola berbeda. Selain itu, pada sistem e-dictionary untuk saran kueri, obat juga mengaplikasikan algoritma yang sama untuk mengatasi kesalahan ketik, yang relevan dengan penelitian ini karena memfokuskan pada peningkatan akurasi pencarian meskipun terdapat perbedaan dalam ejaan.

Selain itu, algoritma Levenshtein Distance untuk optimasi pencarian data pencarian arsip desa [8] dalam aplikasi berbasis web juga memberikan kontribusi terhadap pengembangan sistem pencarian berbasis algoritma ini, dengan hasil peningkatan akurasi pencarian hingga 94%. Penelitian ini menunjukkan aplikasi algoritma dalam meningkatkan efisiensi pencarian data yang memiliki variasi ejaan, yang juga sejalan dengan tujuan penelitian ini. Dalam menguji efisiensi algoritma dalam menghitung kesamaan dokumen menggunakan pendekatan berbasis kompresi, yang meningkatkan kinerja pencarian dokumen dalam dataset besar [9]. Semua penelitian ini menunjukkan efektivitas algoritma Levenshtein Distance dalam meningkatkan akurasi pencarian, yang menjadi dasar utama dalam pengembangan sistem pengarsipan berbasis web di PMI Kota Jakarta, dengan menekankan pengelolaan dokumen yang efisien dan terstruktur.

III. METODE PENELITIAN

1. Metode Waterfall

Penelitian ini mengadopsi metode pengembangan waterfall yang mencakup tahapan-tahapan seperti analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan. Bab ini akan membahas hasil yang diperoleh dari setiap tahapan tersebut, termasuk penerapan sistem yang telah dirancang serta analisis kinerja sistem berdasarkan hasil pengujian.

2. Algoritma Levenshtein Distance

Algoritma Levenshtein distance memanfaatkan matriks untuk menilai sejauh mana dua string berbeda. Perhitungan jarak didasarkan pada jumlah operasi minimum yang diperlukan agar satu string dapat diubah menjadi string lain [6]. Tiga operasi utama yang dianalisis meliputi penyisipan karakter (insertion), penghapusan karakter (deletion), serta penggantian karakter (substitution) [10].

Jarak Levenshtein antara dua string a dan b dihitung berdasarkan jumlah operasi minimal penyisipan, penghapusan, dan substitusi karakter yang diperlukan untuk mentransformasi a menjadi b . Secara matematis, jarak tersebut dirumuskan sebagai [7]:

$$\text{lev}_{a,b}(i,j) \begin{cases} \text{if } \min(i,j) = 0 \\ \max(i,j) \\ \min \begin{cases} \text{lev}_{a,b}(i-1,j) + - \\ \text{lev}_{a,b}(i,j-1) + 1 \\ \text{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} \end{cases} \quad (1)$$

Keterangan:

$a = \text{String Sumber}$

b = *String Target*
 lev = *Jarak Levenshtein Distance*
 i = *Nilai Karakter String Sumber*
 j = *Nilai Karakter String Target*

3. Perhitungan Nilai Similaritas

Hasil dari perhitungan Algoritma Levenshtein Distance selanjutnya akan dihitung kedalam sebuah perhitungan similaritas, untuk mendapatkan persentase kemiripan kata pencarian terhadap kata yang tersimpan di dalam database. Rumus similaritas yang digunakan adalah persamaan [11].

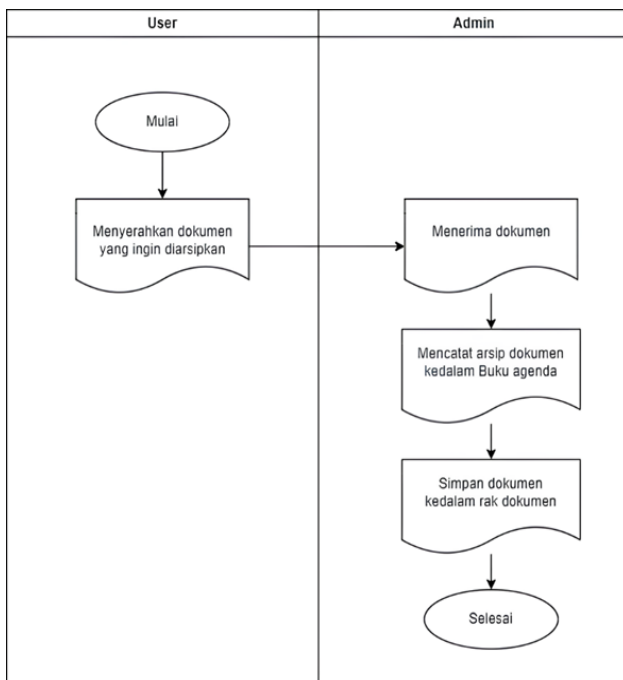
$$\text{Similarity} = \left(1 - \frac{\text{LevenshteinDistance}(s1,s2)}{\max(|s1|,|s2|)}\right) \times 100\% \quad (2)$$

Dimana,

s1 = kata yang dimasukkan, s2 = kata yang tersimpan dalam database

4. Aliran Sistem Berjalan

Saat ini, pengolahan dan pencarian dokumen masih dilakukan secara manual menggunakan kertas dan diurutkan berdasarkan tanggal diterimanya dokumen. Proses pencarian dokumen seperti ini sangat memakan waktu, terutama ketika dokumen yang dibutuhkan berada dalam kondisi mendesak.



Gambar 1. Aliran Sistem Berjalan

Pada gambar 1 Dari alur pada Gambar 1, dapat disimpulkan bahwa proses pengarsipan masih berjalan secara manual. User hanya berperan sebagai penyerah dokumen, sementara Admin bertanggung jawab penuh mulai dari penerimaan, pencatatan, hingga penyimpanan dokumen. Proses ini memastikan adanya jejak administrasi dan

ketersediaan dokumen secara fisik di rak penyimpanan.

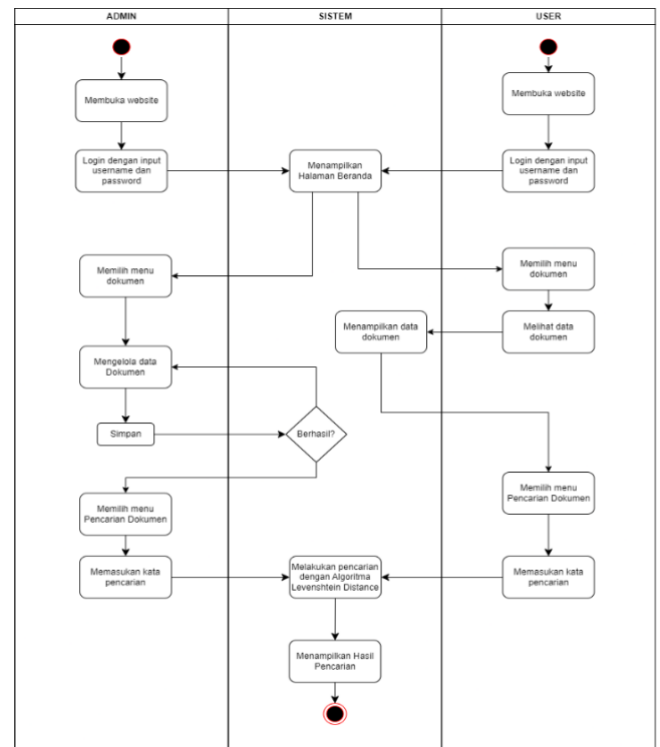
Pada pengarsipan di PMI UDD Kota Jakarta :

- User akan membawa dokumen untuk melakukan penyimpanan dokumen.
- Admin akan menerima dokumen dan melakukan pengarsipan terhadap dokumen yang akan diarsipkan tersebut.
- Admin akan mencatat dokumen tersebut kedalam buku agenda yang masih manual.
- Admin menyimpan dokumen kedalam kardus dan menyimpannya kedalam ruangan.

Permasalahan di UDD PMI Kota Jakarta adalah penyimpanan dokumen yang masih dilakukan secara manual, di mana dokumen-dokumen dimasukkan ke dalam kardus yang saling tertumpuk, menyulitkan pencarian dan terbatasnya ruang penyimpanan. Hal ini memerlukan sistem yang dapat menyimpan dokumen secara digital, memudahkan pencarian, serta mengurangi risiko kehilangan arsip fisik. Penerapan algoritma Levenshtein Distance dapat meningkatkan akurasi pencarian dokumen, bahkan dengan kesalahan penulisan, dan memastikan pengelolaan arsip yang lebih efisien.

5. Analisis Usulan

Analisis usulan sistem yang diberikan penulis dalam Admin dan User Melakukan pengelolaan dokumen dan pencarian arsip dokumen menggunakan algoritma Levenshtein Distance digambarkan pada gambar berikut:

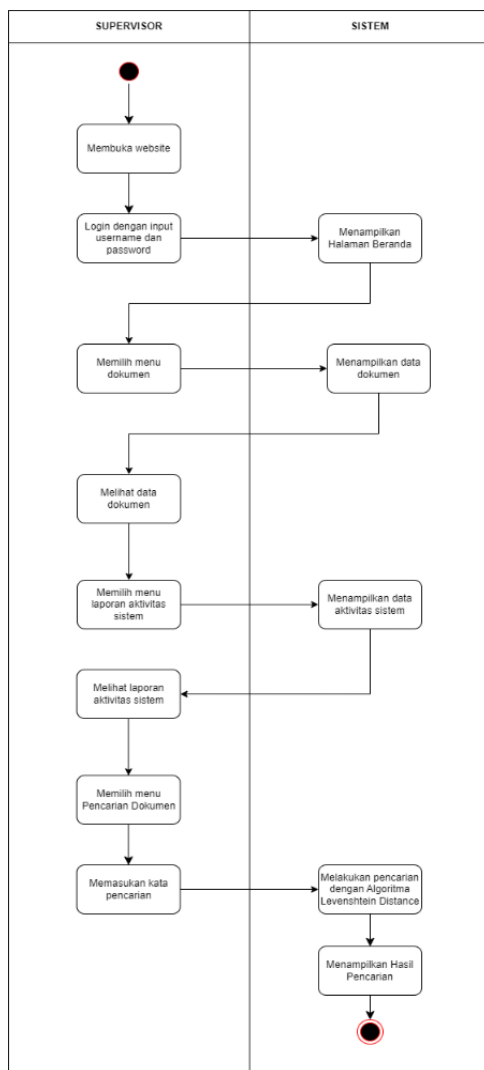


Gambar 2. Analisis Usulan User dan Admin

Penjelasan Flowchart sistem untuk User dan Admin yang diusulkan adalah sebagai berikut:

- Admin dan user membuka website arsip dokumen UDD PMI Kota Jakarta.
- Admin dan user login ke dalam sistem dengan menggunakan username dan Password yang terdaftar.
- Admin dan user memilih menu dokumen.
- Admin mengelola data dokumen dan menyimpan data dokumen ke dalam sistem.
- User melihat data dokumen.
- Admin dan user mencari dokumen pada menu pencarian dokumen.
- Admin dan user memasukkan kata pencarian ke dalam kolom pencarian.
- Sistem melakukan pencarian menggunakan algoritma levenshtein distance.
- Sistem menampilkan hasil pencarian.

Berikut adalah analisis usulan sistem yang diberikan penulis dalam Supervisor Melakukan melihat laporan aktivitas, mengakses dokumen dan pencarian arsip menggunakan algoritma Levenshtein Distance digambarkan pada Gambar 3.



Gambar 3. Analisis Usulan SuperVisor

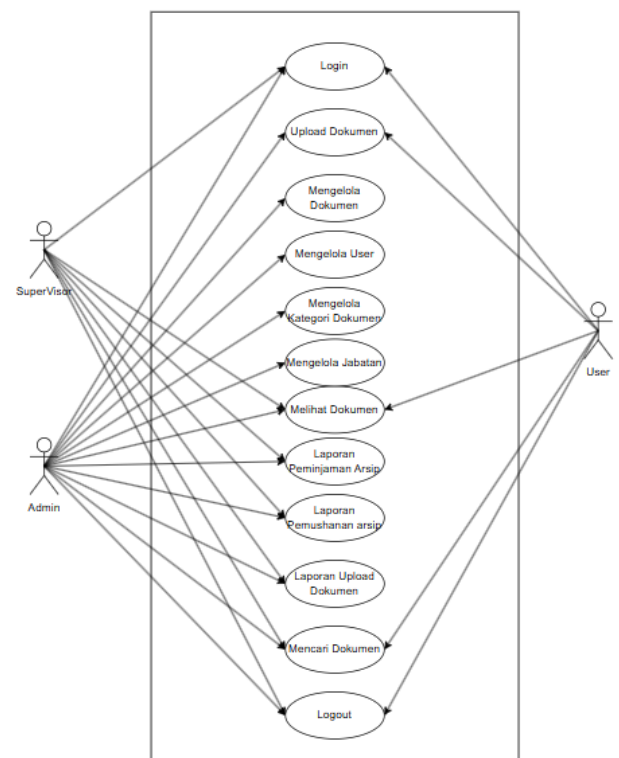
Penjelasan Flowchart sistem untuk User dan Admin yang diusulkan adalah sebagai berikut:

- Supervisor membuka website arsip dokumen UDD PMI Kota Jakarta.
- Supervisor login ke dalam sistem dengan menggunakan username dan password yang terdaftar.
- Supervisor memilih menu dokumen.
- Sistem menampilkan data dokumen.
- Supervisor melihat data dokumen.
- Supervisor memilih menu laporan aktivitas sistem.
- Sistem menampilkan data aktivitas sistem.
- Supervisor melihat laporan aktivitas sistem.
- Supervisor memilih menu pencarian dokumen.
- Supervisor memasukkan kata pencarian ke dalam kolom pencarian.
- Sistem melakukan pencarian menggunakan algoritma Levenshtein Distance.
- Sistem menampilkan hasil pencarian.

IV. HASIL DAN PEMBAHASAN

1. Perancangan

Sebelum membuat sistem pengarsipan ini, penulis melakukan perancangan terlebih dahulu. Desainnya dijelaskan dalam 4 diagram UML yaitu use case diagram, activity Diagram, sequence diagram dan class diagram. Diagram-Diagram tersebut yang umumnya digunakan dalam membuat perancangan. Berikut ini adalah desain sistem pengarsipan yang akan dibuat.



Gambar 4. Use case diagram sistem

a. Use Case Diagram

Usecase diagram berikut menggambarkan perancangan sistem pencarian dokumen berbasis web yang menggunakan algoritma Levenshtein Distance. Diagram ini bertujuan untuk memvisualisasikan interaksi antara aktor-aktor yang terlibat, seperti pengguna dan sistem, serta mendeskripsikan fungsi-fungsi utama yang mendukung proses pencarian dokumen secara efektif.

b. Implementasi

Implementasi merujuk pada tahap pelaksanaan suatu sistem setelah proses perancangannya selesai. Berikut ini adalah hasil dari pelaksanaan sistem ini.

1) Tampilan Halaman Login

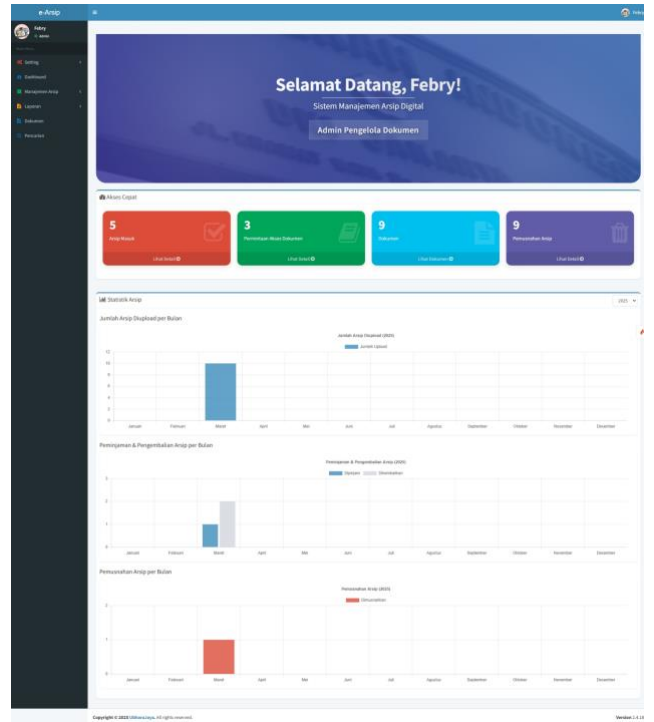
Halaman Login Manajemen Arsip, yang muncul pertama kali ketika pengguna mengakses sistem. Di sini, pengguna diminta untuk memasukkan username dan password yang telah terdaftar.

2) Tampilan Dashboard

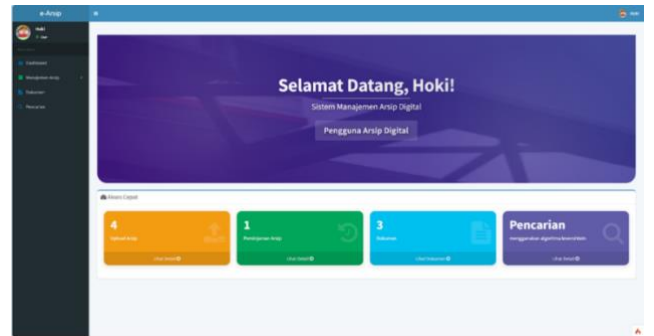
Halaman Dashboard merupakan tampilan yang muncul setelah proses login, yang berisi panel kontrol. Berikut ini adalah tampilan dari halaman dashboard tersebut.



Gambar 5. Tampilan Halaman Login



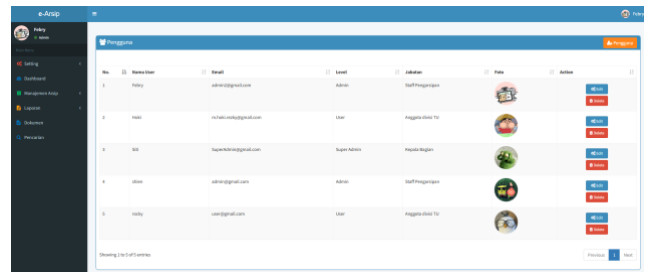
Gambar 7. Tampilan Dashboard Admin



Gambar 8. Tampilan Dashboard User

3) Tampilan Halaman Tambah pengguna

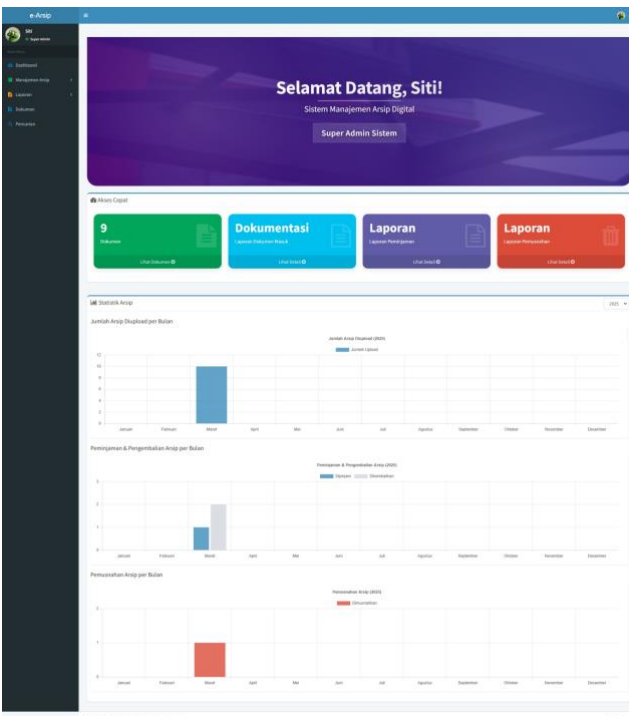
Halaman Tambah Pengguna, Admin dapat melihat daftar pengguna yang terdaftar dalam sistem dan melakukan tindakan seperti menambah, mengedit, atau menghapus akun pengguna yang sudah tidak aktif.



Gambar 9. Tampilan Halaman Tambah Pengguna

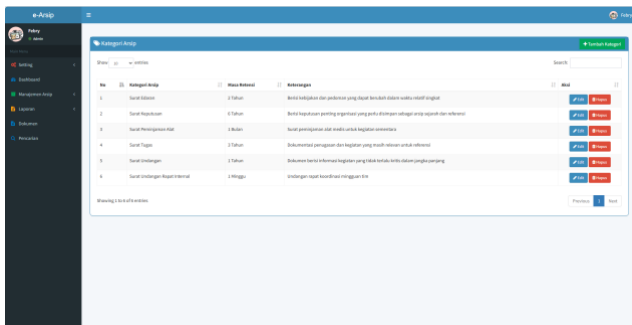
4) Tampilan Halaman Tambah Kategori

Halaman Tambah Kategori digunakan oleh Admin untuk mengelola kategori arsip dalam sistem. Admin dapat menambah kategori baru untuk mengelompokkan dokumen berdasarkan jenis atau topik tertentu dan membuat masa retensi dari kategori. Selain itu, Admin



Gambar 6. Tampilan Dashboard Supervisor

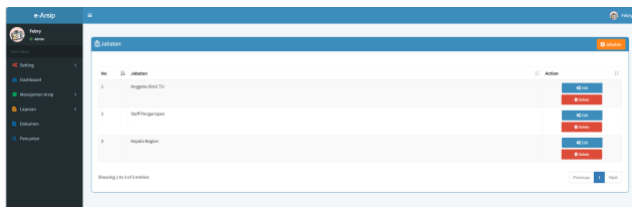
dapat mengedit atau menghapus kategori yang sudah ada.



Gambar 10. Tampilan Halaman Tambah Kategori

5) Tampilan Halaman Tambah Jabatan

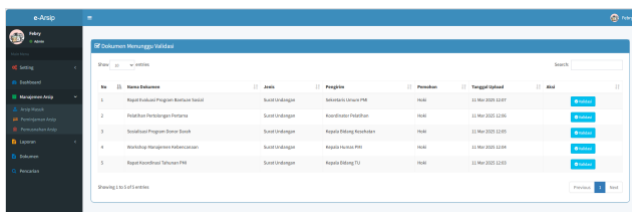
Halaman Tambah Jabatan digunakan oleh Admin untuk mengelola jabatan-jabatan dalam sistem. Admin dapat menambah jabatan baru, mengedit jabatan yang ada, atau menghapus jabatan yang tidak diperlukan.



Gambar 11. Tampilan Halaman Tambah Jabatan

6) Tampilan Halaman Daftar Dokumen yang Menunggu Validasi

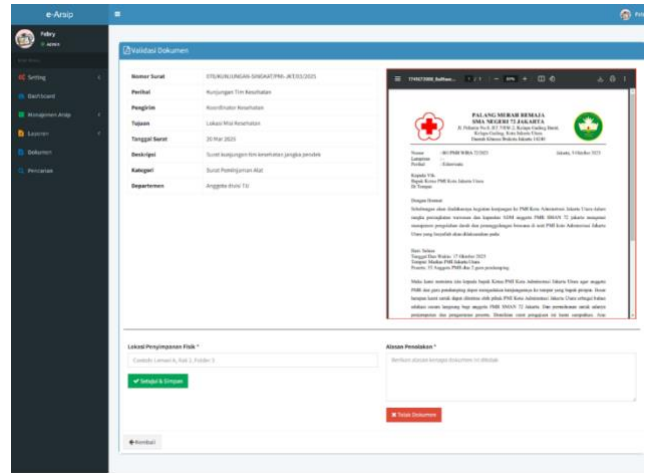
Halaman Daftar Dokumen yang Menunggu Validasi digunakan oleh Admin untuk memvalidasi arsip yang masuk. Admin memeriksa kelengkapan dan keabsahan dokumen yang diunggah oleh User. Jika dokumen memenuhi standar, Admin dapat menyetujuinya dan memindahkannya ke arsip utama. Jika tidak, dokumen akan ditolak dan Admin akan memberikan alasan penolakan.



Gambar 12. Tampilan Halaman daftar Dokimen yang Menunggu Validasi

7) Tampilan Halaman Validasi Dokumentasi

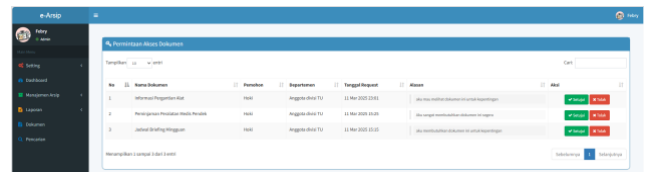
Halaman Validasi Dokumentasi digunakan untuk Admin memeriksa dokumen yang telah diunggah oleh User. Admin meninjau detail dokumen untuk memastikan kelengkapannya. Jika dokumen memenuhi syarat, Admin dapat menyetujuinya dan memindahkan dokumen ke arsip utama. Jika dokumen tidak memenuhi syarat, Admin dapat menolaknya dan memberikan alasan penolakan kepada User.



Gambar 13. Tampilan Halaman Validasi Dokumentasi

8) Tampilan Halaman Permintaan Akses Dokumen

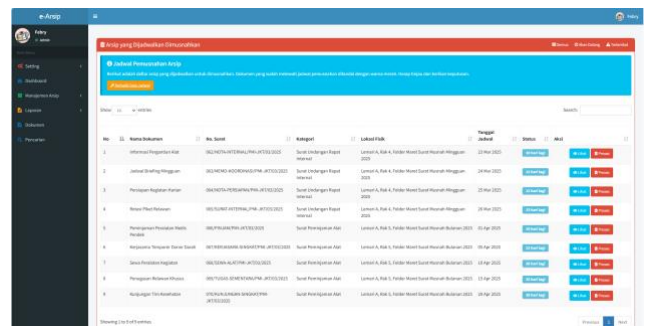
Halaman Permintaan Akses Dokumen digunakan oleh User dan Supervisor untuk mengajukan permintaan peminjaman dokumen. Setelah memilih dokumen yang ingin dipinjam, pengguna mengisi formulir permintaan peminjaman. Permintaan ini kemudian akan diproses oleh Admin. Sistem akan memeriksa status dokumen dan memastikan apakah dokumen tersebut tersedia untuk dipinjam. Setelah disetujui, dokumen akan tersedia untuk diakses atau diunduh oleh pengguna yang memintanya.



Gambar 14. Tampilan Halaman Validasi Dokumentasi

9) Tampilan Halaman Pemusnahan Dokumen

Halaman Pemusnahan Dokumen digunakan oleh Admin untuk dapat memantau dan mengelola dokumen yang akan dimusnahkan. Admin melihat daftar dokumen yang telah masuk dalam kategori pemusnahan dan mengonfirmasi pemusnahan dokumen sesuai dengan prosedur yang berlaku di organisasi.

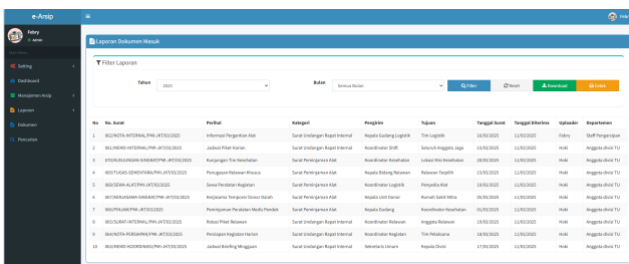


Gambar 15. Tampilan Halaman Pemusnahan Dokumen

10) Tampilan Halaman Laporan Dokumen Masuk

Halaman Laporan Dokumen Masuk memperlihatkan laporan mengenai dokumen yang telah

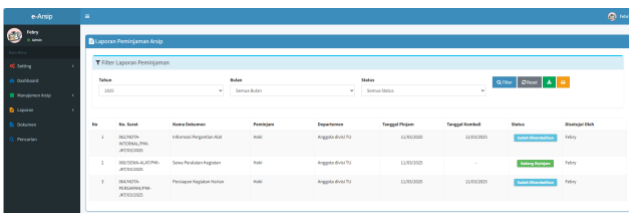
masuk ke dalam sistem. Admin dan Supervisor dapat melihat laporan yang mencakup rincian dokumen, tanggal masuk, dan statusnya.



Gambar 16. Tampilan Halaman Laporan Dokumen Masuk

11) Tampilan Halaman Laporan Peminjaman

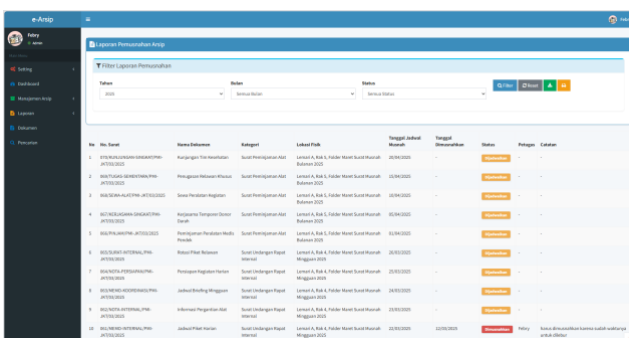
Halaman Laporan Peminjaman memperlihatkan laporan mengenai peminjaman dokumen. Admin dan Supervisor dapat memantau siapa saja yang meminjam dokumen, status peminjaman, serta informasi terkait dengan dokumen yang dipinjam.



Gambar 17. Tampilan Halaman Laporan Peminjaman

12) Tampilan Halaman Laporan Pemusnahan

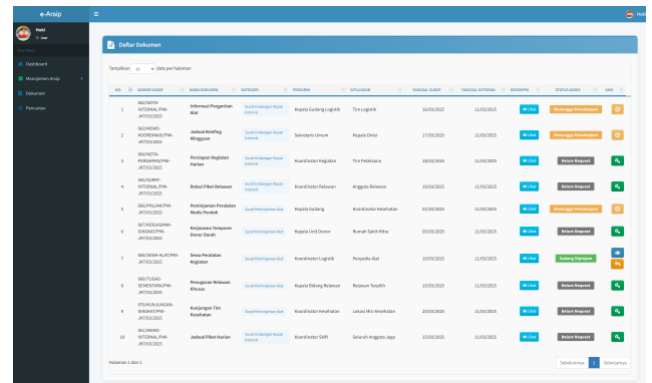
Halaman Laporan Pemusnahan digunakan oleh Admin dan Supervisor untuk melihat laporan pemusnahan dokumen yang telah dilakukan. Laporan ini mencakup informasi mengenai dokumen yang telah dimusnahkan, alasan pemusnahan, dan tanggal pemusnahan.



Gambar 18. Tampilan Halaman Laporan Pemusnahan

13) Tampilan Halaman Daftar Dokumen

Halaman Daftar Dokumen adalah halaman yang menyajikan semua dokumen secara terorganisir dalam satu tempat, memudahkan pengguna untuk mengakses dan mengelola dokumen-dokumen yang tersedia. Setiap dokumen yang terdaftar akan terlihat dengan jelas, memungkinkan navigasi yang efisien bagi pengguna dalam mencari dan memilih dokumen yang diperlukan.



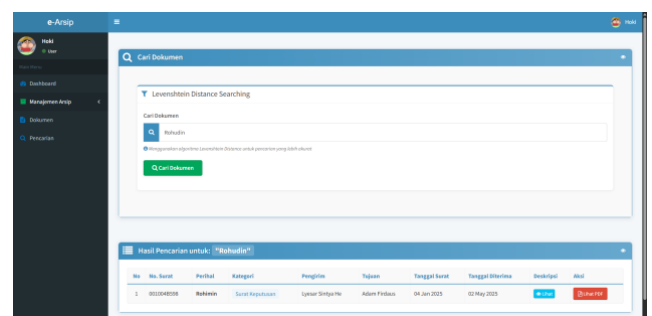
Gambar 19. Tampilan Halaman Daftar Dokumen Supervisor dan User



Gambar 20. Tampilan Halaman Daftar Dokumen Admin

14) Tampilan Halaman Mencari Dokumen dengan Algoritma Levenshtein Distance

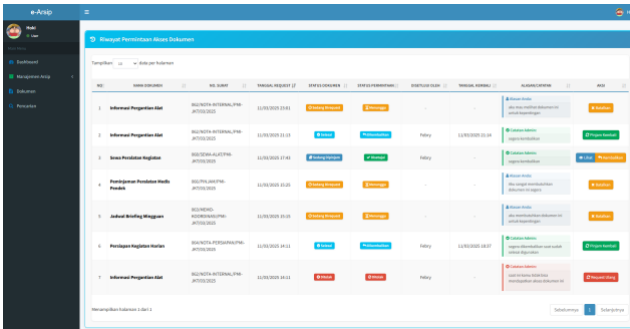
Halaman ini memungkinkan pengguna untuk mencari dokumen dengan memasukkan kata kunci yang diinginkan. Algoritma Levenshtein Distance kemudian menghitung jarak antara kata kunci dan dokumen dalam database, memperhitungkan perbedaan karakter dengan operasi penyisipan, penghapusan, dan penggantian karakter.



Gambar 21. Tampilan Halaman Mencari Dokumen dengan Algoritma Levenshtein Distance

15) Tampilan Halaman Riwayat Permintaan Akses Dokumen

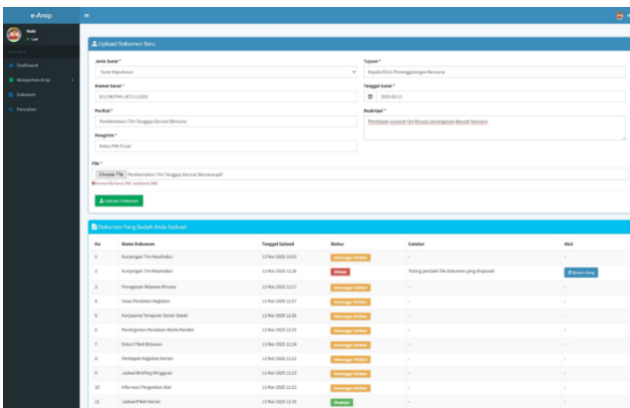
Halaman riwayat permintaan Akses Dokumen memungkinkan pengguna, baik Admin maupun Supervisor, untuk melihat riwayat permintaan akses dokumen yang diajukan oleh User atau Supervisor lainnya.



Gambar 22. Tampilan Halaman Riwayat Permintaan Akses Dokumen

16) Tampilan Halaman User Upload Dokumen

Halaman Upload User Dokumen. Halaman ini memungkinkan User untuk mengunggah dokumen baru ke dalam sistem.



Gambar 23. Tampilan Halaman User Upload Dokumen

2. Pengujian Algoritma Levenshtein Distance

Pengujian ini dilakukan untuk mengevaluasi sejauh mana algoritma Levenshtein Distance dapat mendukung fungsi pencarian arsip. Hasil pencarian akan tetap ditampilkan meskipun terdapat kesalahan dalam kata yang dimasukkan.

Berikut contoh penerapan perhitungan Algoritma Levenshtein Distance secara manual pada sebuah tabel matriks, yaitu antara string sumber "ROHUDIN" dan string target "ROHIMIN". Dimana string sumber memiliki kesalahan.

Tabel 1. Operasi Matriks "ROHUDIN"

| | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|
| | | | R | O | H | I | M | I | N |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | R | 1 | | | | | | | |
| | O | 2 | | | | | | | |
| | H | 3 | | | | | | | |
| | U | 4 | | | | | | | |
| | D | 5 | | | | | | | |
| | I | 6 | | | | | | | |
| | N | 7 | | | | | | | |

Perhitungan Baris 2 (R):

- Sel [R,R] = $\min(1+1, 1+1, 0+0) = 0$ (karakter sama, ambil diagonal kiri atas)
- Sel [R,O] = $\min(0+1, 1+1, 1+1) = 1$ (karakter berbeda, ambil minimum + 1)
- Sel [R,H] = $\min(1+1, 2+1, 2+1) = 2$ (karakter berbeda)
- Sel [R,I] = $\min(2+1, 3+1, 3+1) = 3$ (karakter berbeda)
- Sel [R,M] = $\min(3+1, 4+1, 4+1) = 4$ (karakter berbeda)
- Sel [R,I] = $\min(4+1, 5+1, 5+1) = 5$ (karakter berbeda)
- Sel [R,N] = $\min(5+1, 6+1, 6+1) = 6$ (karakter berbeda)

Perhitungan Baris 3 (O):

- Sel [O,R] = $\min(1+1, 0+1, 1+1) = 1$ (karakter berbeda)
- Sel [O,O] = $\min(1+1, 1+1, 0+0) = 0$ (karakter sama, ambil diagonal kiri atas)
- Sel [O,H] = $\min(0+1, 2+1, 1+1) = 1$ (karakter berbeda)
- Sel [O,I] = $\min(1+1, 3+1, 2+1) = 2$ (karakter berbeda)
- Sel [O,M] = $\min(2+1, 4+1, 3+1) = 3$ (karakter berbeda)
- Sel [O,I] = $\min(3+1, 5+1, 4+1) = 4$ (karakter berbeda)
- Sel [O,N] = $\min(4+1, 6+1, 5+1) = 5$ (karakter berbeda)

Perhitungan Baris 4 (H):

- Sel [H,R] = $\min(2+1, 0+1, 1+1) = 1$ (karakter berbeda)
- Sel [H,O] = $\min(1+1, 1+1, 0+1) = 1$ (karakter berbeda)
- Sel [H,H] = $\min(1+1, 1+1, 0+0) = 0$ (karakter sama, ambil diagonal kiri atas)
- Sel [H,I] = $\min(0+1, 2+1, 1+1) = 1$ (karakter berbeda)
- Sel [H,M] = $\min(1+1, 3+1, 2+1) = 2$ (karakter berbeda)
- Sel [H,I] = $\min(2+1, 4+1, 3+1) = 3$ (karakter berbeda)
- Sel [H,N] = $\min(3+1, 5+1, 4+1) = 4$ (karakter berbeda)

Perhitungan Baris 5 (U):

- Sel [U,R] = $\min(3+1, 0+1, 1+1) = 1$ (karakter berbeda)
- Sel [U,O] = $\min(2+1, 1+1, 0+1) = 1$ (karakter berbeda)
- Sel [U,H] = $\min(1+1, 0+1, 1+1) = 1$ (karakter berbeda)
- Sel [U,I] = $\min(0+1, 1+1, 1+1) = 1$ (karakter berbeda)

- Sel [U,M] = $\min(1+1, 2+1, 2+1) = 2$ (karakter berbeda)
- Sel [U,I] = $\min(2+1, 3+1, 3+1) = 3$ (karakter berbeda)
- Sel [U,N] = $\min(3+1, 4+1, 4+1) = 4$ (karakter berbeda)

Perhitungan Baris 6 (D):

- Sel [D,R] = $\min(4+1, 3+1, 2+1) = 3$ (karakter berbeda)
- Sel [D,O] = $\min(3+1, 2+1, 1+1) = 2$ (karakter berbeda)
- Sel [D,H] = $\min(2+1, 1+1, 0+1) = 1$ (karakter berbeda)
- Sel [D,I] = $\min(1+1, 1+1, 1+1) = 2$ (karakter berbeda)
- Sel [D,M] = $\min(1+1, 2+1, 2+1) = 2$ (karakter berbeda)
- Sel [D,I] = $\min(2+1, 3+1, 3+1) = 3$ (karakter berbeda)
- Sel [D,N] = $\min(3+1, 4+1, 4+1) = 4$ (karakter berbeda)

Perhitungan Baris 7 (I):

- Sel [I,R] = $\min(5+1, 4+1, 3+1) = 4$ (karakter berbeda)
- Sel [I,O] = $\min(4+1, 3+1, 2+1) = 3$ (karakter berbeda)
- Sel [I,H] = $\min(3+1, 2+1, 1+1) = 2$ (karakter berbeda)
- Sel [I,I] = $\min(2+1, 1+1, 2+0) = 2$ (karakter sama, ambil diagonal kiri atas)
- Sel [I,M] = $\min(2+1, 2+1, 2+1) = 3$ (karakter berbeda)
- Sel [I,I] = $\min(3+1, 3+1, 3+0) = 2$ (karakter sama, ambil diagonal kiri atas)
- Sel [I,N] = $\min(2+1, 4+1, 3+1) = 3$ (karakter berbeda)

Perhitungan Baris 8 (N):

- Sel [N,R] = $\min(6+1, 5+1, 4+1) = 5$ (karakter berbeda)
- Sel [N,O] = $\min(5+1, 4+1, 3+1) = 4$ (karakter berbeda)
- Sel [N,H] = $\min(4+1, 3+1, 2+1) = 3$ (karakter berbeda)
- Sel [N,I] = $\min(3+1, 2+1, 3+1) = 3$ (karakter berbeda)
- Sel [N,M] = $\min(2+1, 3+1, 2+1) = 3$ (karakter berbeda)
- Sel [N,I] = $\min(3+1, 2+1, 3+1) = 3$ (karakter berbeda)
- Sel [N,N] = $\min(3+1, 3+1, 3+0) = 2$ (karakter sama, ambil diagonal kiri atas)

Tabel 2. Hasil Operasi Perhitungan Matriks “ROHUDIN”

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | R | O | H | I | M | I | N |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| R | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | |
| O | 2 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | |
| H | 3 | 2 | 1 | 0 | 1 | 2 | 3 | 4 | |
| U | 4 | 3 | 2 | 1 | 1 | 2 | 3 | 4 | |
| D | 5 | 4 | 3 | 2 | 2 | 2 | 3 | 4 | |
| I | 6 | 5 | 4 | 3 | 2 | 3 | 2 | 3 | |
| N | 7 | 6 | 5 | 4 | 3 | 3 | 3 | 2 | |

Berdasarkan perhitungan diatas maka dapat disimpulkan bahwa jarak antara string “ROHUDIN” dan string “ROHIMIN” adalah sebanyak 2, dimana operasi yang terjadi didalamnya adalah mengubah karakter “U” menjadi “I” (subsitusi) dan mengubah karakter “D” menjadi “M” (subsitusi).

3. Perhitungan Nilai Similaritas

$$\text{Similaritas} = \left(1 - \frac{2}{7}\right) \times 100\%$$

$$= \left(\frac{5}{7}\right) \times 100\%$$

$$= 71,43\%$$

Dari analisis ini, dapat disimpulkan bahwa meskipun terdapat kesalahan pada string sumber, tingkat similaritas yang tinggi (71,43%) menunjukkan bahwa kedua string tersebut memiliki kemiripan yang signifikan dan kemungkinan besar merujuk pada entitas yang sama dengan sedikit variasi dalam penulisan.

V. KESIMPULAN

Hasil penelitian menunjukkan bahwa sistem informasi pengarsipan berbasis web yang dilengkapi dengan algoritma Levenshtein Distance terbukti mampu meningkatkan efisiensi, akurasi, serta keamanan dalam pengelolaan dokumen di PMI Kota Jakarta. Sistem ini berhasil mengatasi keterbatasan metode pengarsipan manual yang rentan terhadap kesalahan pencatatan, kesulitan pencarian, serta risiko kerusakan dan kehilangan arsip fisik. Dengan memanfaatkan algoritma Levenshtein Distance, sistem mampu memberikan hasil pencarian yang akurat meskipun terdapat variasi penulisan atau kesalahan ketik pada kata kunci, sehingga proses akses informasi menjadi lebih cepat dan tepat.

Selain itu, penerapan sistem ini berkontribusi pada peningkatan produktivitas organisasi, pengurangan ketergantungan terhadap dokumen fisik, dan penyederhanaan alur operasional pengarsipan. Namun demikian, pengembangan sistem di masa mendatang masih dapat ditingkatkan, terutama pada aspek keamanan data melalui penerapan enkripsi dokumen dan autentikasi multi-faktor, serta perluasan dukungan format dokumen agar dapat mencakup Word, Excel, maupun gambar. Pemeliharaan dan pembaruan

algoritma juga diperlukan secara berkala guna menjaga kinerja sistem dalam jangka panjang. Di samping itu, pelatihan intensif bagi pengguna, khususnya Admin dan Supervisor, penting dilakukan agar sistem dapat dioperasikan secara optimal sesuai dengan kebutuhan organisasi.

10.1093/nar/gkg745.
 [11] Y. H. Agustin, Y. Septiana, and A. Y. Aspahany, "Search Optimization of PIP Scholarship Recipients In Web-Based Student Data Application Using The Levenshtein Distance Algorithm," vol. 7, no. 4, pp. 2069–2081, 2023.

DAFTAR PUSTAKA

- [1] A. Noeman, D. R. Fauzi, D. Hartanti, A. Hiswara, and P. Kustanto, "Employee Work Schedule Presence System Application Design at PT . Tasya Putry Mandiri Uses Genetic Algorithm Method," vol. 2, no. 2, pp. 51–54, 2024.
- [2] R. Gusdiana, I. Alfian, and C. Juliane, "Implementation of Text Processing for Sentiment Analysis of Tax Payment Interest After the 'Rubicon' Phenomenon," *J. Tek. Inform.*, vol. 4, no. 5, pp. 1157–1164, 2023, doi: 10.52436/1.jutif.2023.4.5.1014.
- [3] M. Haris and A. Farisi, "K-Means Algorithm and Levenshtein Distance Algorithm For Sentiment Analysis of School Zonation System Policy," no. November, 2021, doi: 10.1109/ICIC54025.2021.9632943.
- [4] M. L. Ishimora and S. Rejeki, "Web-Based Mail Search Using the Levenshtein Distance Algorithm," vol. 10, no. 225, pp. 143–156, 2026.
- [5] H. T. Sadiyah, M. Saad, N. Ishlah, and N. N. Rokhmah, "Query Suggestion on Drugs e-Dictionary Using the Levenshtein Distance Algorithm," vol. 10, no. 3, pp. 193–202, 2019.
- [6] H. Schwarz, W. Weber, I. Minderop, and B. Weiß, "In Search of the Best Response Scale in a Mixed-mode Survey (Web and Mail). Evidence from MTMM Experiments in the GESIS Panel," *Methods, Data, Anal.*, vol. 15, no. 2, pp. 161–190, 2021, doi: 10.12758/mda.2021.05.
- [7] I. D. Wijaya, E. Irawan, and A. N. Rahmanto, "Rancang Bangun Pembuatan Sitem Informasi Manajemen Dokumen E-Arsip PT.PLN (Persero) Unit Induk Pembangkitan Sumatera Bagian Selatan," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 4, pp. 468–475, 2023, doi: 10.47233/jteksis.v5i4.939.
- [8] F. W. Arsyta and R. E. Putra, "Penerapan Algoritma Damerau Levenshtein Distance Pada Pencarian Arsip Desa Jerukseger Pendukung ISO," *J. Informatics Comput. Sci.*, vol. 4, no. 4, pp. 423–435, 2023.
- [9] P. Zidni and A. Iskandar, "Perbandingan Algoritma Ratcliff/Obershelp Dan Levenshtein Distance Dalam Chatbot Aimi Untuk Universitas Xyz," *J. Comput. Sci. Inf. Technol.*, vol. 2, no. 2, pp. 227–240, 2025, doi: 10.70248/jcsit.v2i2.2041.
- [10] Z. Zhang and M. Gerstein, "Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes," *Nucleic Acids Res.*, vol. 31, no. 18, pp. 5338–5348, 2003, doi: