

Sistem Temu Balik Informasi Berbasis Fuzzy String Matching untuk Pencarian Nama Kontak

Najwa Salsabila Damanik^{1,*}, Putri Afrilia², Adinda Syahara³, Safrizal⁴

^{1,2,3}Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Kaputama, Binjai, Indonesia

⁴Sistem Informasi, Fakultas Ekonomi Sains dan Teknologi, Universitas Muhammadiyah Asahan, Indonesia

Email: ¹*salsabila101819@gmail.com, ²putriaprlaa3@gmail.com, ³adindasyahara757@gmail.com,

⁴rizalsyl75@gmail.com

(* Email Corresponding Author: salsabila101819@gmail.com)

Received: January 13, 2026 | Revision: January 17, 2026 | Accepted: January 17, 2026

Abstrak

Pencarian yang kurang efektif sering kali terjadi karena kesalahan ketik, salah ejaan, atau pengguna yang lupa cara mengeja nama dengan benar saat mencari seseorang di daftar kontak. Untuk membuat pencarian kontak lebih akurat dan fleksibel, penelitian ini membangun dan menerapkan sistem penelusuran informasi berbasis web yang memanfaatkan algoritma pencocokan string fuzzy. Metode pencocokan string perkiraan ini menggunakan algoritma Jarak Levenshtein, yang membantu sistem menemukan kesamaan antara kata kunci yang diketik pengguna dan nama-nama di database, bahkan jika ada perbedaan karakter dari kesalahan ketik atau singkatan. Sistem dikembangkan dengan bahasa pemrograman PHP, database MySQL, dan framework UI Bootstrap. Sistem ini mampu menampilkan daftar kontak relevan berdasarkan jarak edit terkecil, dengan hasil diurutkan dari yang paling mirip, seperti yang terlihat dari hasil pengujian fungsional. Pengujian menunjukkan bahwa pencocokan fuzzy efektif menangani input yang tidak tepat saat mencari nama. Akibatnya, sistem ini lebih adaptif dibandingkan metode pencarian tradisional yang ketat, karena menyederhanakan, mempercepat pencarian, dan lebih tahan terhadap kesalahan data. Harapannya, sistem ini akan berguna baik untuk keperluan pribadi maupun organisasi.

Kata Kunci: fuzzy string matching, levenshtein distance, pencarian nama, daftar kontak, sistem temu balik informasi.

Abstract

Ineffective searches often occur due to typos, misspellings, or users forgetting how to spell a name correctly when searching for someone in a contact list. To make contact search more accurate and flexible, this study developed and implemented a web-based information retrieval system that utilizes a fuzzy string matching algorithm. This approximate string matching method uses the Levenshtein Distance algorithm, which helps the system find similarities between user-typed keywords and names in the database, even if there are character differences from typos or abbreviations. The system was developed using the PHP programming language, MySQL database, and the Bootstrap UI framework. The system is capable of displaying a list of relevant contacts based on the smallest edit distance, with results sorted from the most similar, as seen from functional testing results. Testing shows that fuzzy matching effectively handles imprecise input when searching for names. As a result, this system is more adaptive than traditional, strict search methods, as it simplifies, speeds up searches, and is more robust to data errors. It is hoped that this system will be useful for both personal and organizational purposes.

Keywords: fuzzy string matching, levenshtein distance, name search, contact list, information retrieval system.

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah meningkatkan kebutuhan akan sistem pencarian data yang cepat, akurat, dan fleksibel. Hampir semua aktivitas digital modern, seperti sistem manajemen kontak, layanan pelanggan, aplikasi pesan instan, dan sistem informasi organisasi, sangat bergantung pada kemampuan sistem pencarian untuk menemukan data yang relevan berdasarkan istilah pencarian yang dimasukkan pengguna[1]. Namun, dalam praktiknya, proses pencarian seringkali menemui hambatan karena kesalahan ketik, perbedaan ejaan, penggunaan singkatan, atau variasi ejaan nama, terutama untuk data tekstual seperti nama diri[2].

Sebagian besar sistem pencarian tradisional masih mengandalkan pencocokan tepat atau pencarian awalan. Metode ini hanya dapat mengembalikan hasil jika input pengguna sama persis dengan data yang tersimpan dalam basis data, termasuk urutan karakter, spasi, dan kapitalisasi[3]. Pendekatan ini menjadi kurang efektif ketika pengguna melakukan kesalahan ketik atau lupa ejaan nama yang tepat. Hal ini sangat umum terjadi ketika mencari nama kontak, terutama untuk nama dengan variasi ejaan seperti "Ahmad" atau "Achmad," yang dieja berbeda di Indonesia[4].

Masalah kesalahan ketik ini berdampak langsung pada efisiensi pencarian. Beberapa penelitian menunjukkan bahwa sistem pencarian kecocokan persis memiliki tingkat kesalahan yang tinggi ketika dihadapkan dengan kesalahan ejaan. Akibatnya, pengguna harus mengulangi pencarian mereka beberapa kali,

atau bahkan gagal menemukan data yang tersedia dalam sistem[1]. Hal ini tidak hanya mengurangi waktu pencarian tetapi juga berdampak negatif pada pengalaman pengguna secara keseluruhan.

Untuk mengatasi masalah ini, metode pencocokan string berdasarkan kesamaan telah menjadi solusi yang banyak dikembangkan dalam sistem penelusuran informasi. Salah satu teknik yang umum digunakan adalah pencocokan string fuzzy, metode yang tidak memerlukan kesamaan absolut tetapi lebih mengukur tingkat kesamaan antara dua string[5]. Pendekatan ini memungkinkan sistem untuk memberikan hasil yang relevan bahkan jika terdapat perbedaan karakter karena kesalahan ketik, singkatan, atau variasi ejaan[6].

Pencocokan string fuzzy umumnya diimplementasikan menggunakan konsep pencocokan string perkiraan, yang melibatkan penghitungan jarak atau ketidakmiripan antara string berdasarkan operasi dasar seperti penyisipan, penghapusan, dan substitusi karakter[7]. Algoritma jarak Levenshtein adalah algoritma yang populer dan banyak digunakan dalam pendekatan ini. Algoritma ini menghitung jarak edit minimum yang diperlukan untuk mengubah satu string menjadi string lain dan karenanya dapat berfungsi sebagai indikator kesamaan antara dua kata atau frasa[8].

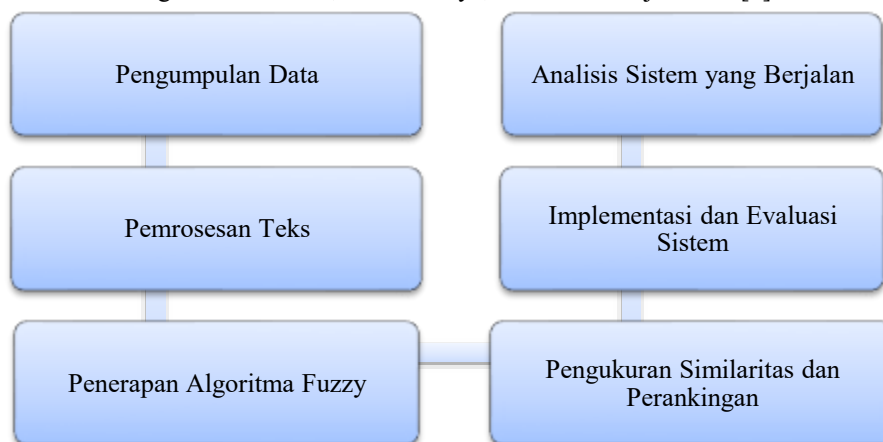
Studi sebelumnya telah menunjukkan efektivitas algoritma jarak Levenshtein dalam meningkatkan akurasi pencarian teks dalam berbagai konteks, seperti sistem pengambilan metadata, FAQ, chatbot layanan publik, sistem perpustakaan digital, dan validasi dokumen teks[7][9]. Algoritma ini mentolerir kesalahan ejaan ringan hingga sedang, sehingga sangat cocok untuk mencari nama kontak, yang dicirikan oleh kumpulan data pendek dan rentan terhadap kesalahan ejaan[10].

Dalam konteks sistem manajemen kontak, kebutuhan akan pencarian yang toleran terhadap kesalahan menjadi semakin penting. Pengguna seringkali hanya mengingat sebagian nama, menggunakan nama panggilan, atau memasukkan ejaan yang tidak konsisten dengan data yang tercatat. Sistem pencarian yang kaku tidak dapat memperhitungkan situasi ini, sehingga memerlukan pendekatan yang lebih adaptif dan cerdas[11]. Dengan menerapkan pencocokan string fuzzy berdasarkan jarak Levenshtein, sistem dapat menampilkan daftar kontak yang terkait erat dengan istilah pencarian dan kemudian mengurutkannya berdasarkan kemiripan yang menurun[12].

Penelitian ini bertujuan merancang dan mengimplementasikan sistem temu balik informasi berbasis web yang mampu melakukan pencarian nama kontak secara lebih fleksibel dengan memanfaatkan metode fuzzy string matching dan algoritma Levenshtein Distance[13]. Sistem yang dikembangkan diharapkan mampu meningkatkan efektivitas pencarian, mempercepat waktu temu balik data, serta memberikan pengalaman pengguna yang lebih baik dibandingkan metode pencarian eksak[14]. Selain itu, penelitian ini juga diharapkan dapat memberikan kontribusi akademik dalam penerapan algoritma pencocokan string pada sistem informasi berbasis web, khususnya dalam pengelolaan data kontak[15].

2. METODOLOGI PENELITIAN

Penelitian ini dirancang secara sistematis untuk mengembangkan sistem penelusuran informasi yang tepat untuk melacak nama kontak. Metodologi dimulai dengan pengumpulan data kontak, diikuti dengan pra-pemrosesan teks untuk menstandarisasi format nama, penerapan algoritma pencocokan string fuzzy berbasis jarak Levenshtein untuk mengukur kesamaan, dan akhirnya, evaluasi kinerja sistem[4].



Gambar 1. Struktur penelitian

2.1. Pengumpulan Data

Penelitian ini menggunakan data primer yang dihasilkan secara manual untuk simulasi sistem. Data tersebut terdiri dari daftar nama kontak yang disimpan dalam basis data MySQL. Data kontak mencakup atribut kunci

seperti nama lengkap, nomor telepon, dan alamat email. Karakteristik data nama, seperti ejaan, singkatan, dan potensi kesalahan ketik, memerlukan penanganan khusus agar dapat diproses secara efektif oleh algoritma pencocokan. Data tersebut disusun secara terstruktur agar dapat diproses dan ditelusuri oleh sistem secara efisien.

2.2. Pra-pemrosesan Teks (Text Preprocessing)

Untuk memastikan pencocokan string yang konsisten, nama kontak yang akan dicari dan kata kunci yang dimasukkan oleh pengguna harus diproses terlebih dahulu. Sebagai langkah pertama, semua huruf dalam nama dan kueri pencarian diubah menjadi huruf kecil untuk menghilangkan sensitivitas huruf besar/kecil. Selanjutnya, karakter non-alfanumerik seperti tanda baca atau spasi tambahan dihilangkan menggunakan ekspresi reguler[5]. Saat mencari nama kontak, kata-kata penghenti umumnya tidak dihilangkan, karena setiap kata dalam nama dianggap relevan.

2.3. Penerapan Fuzzy String Matching dan Levenshtein Distance

Inti dari sistem pencarian ini adalah penerapan pencocokan string fuzzy dengan algoritma jarak Levenshtein. Metode ini digunakan untuk mengukur kesamaan antara kueri pencarian yang dimasukkan pengguna dan nama yang tersimpan dalam basis data. Jarak Levenshtein menghitung jumlah minimum penyisipan, penghapusan, dan penggantian karakter yang diperlukan untuk mengubah satu string menjadi string lain. Semakin kecil jarak edit yang dihasilkan, semakin tinggi kesamaan antara kedua string tersebut[12]. Sistem menghitung jarak ini untuk setiap nama dalam basis data relatif terhadap kueri pencarian.

Contoh Perhitungan Levenshtein Distance:

Untuk memberikan gambaran konkret tentang cara kerja algoritma, berikut perhitungan manual antara kata kunci yang dimasukkan pengguna "ahmad" dan nama "achmad" yang ada dalam basis data:

- String A (Query): "ahmad"
- String B (Data): "achmad"

Matriks perhitungan Levenshtein Distance:

Tabel 1. Matriks perhitungan Levenshtein Distance antara kata "ahmad" dan "achmad" :

| | a | c | h | m | a | d | |
|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| a | 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| h | 2 | 1 | 1 | 1 | 2 | 3 | 4 |
| m | 3 | 2 | 2 | 2 | 1 | 2 | 3 |
| a | 4 | 3 | 3 | 3 | 2 | 1 | 2 |
| d | 5 | 4 | 4 | 4 | 3 | 2 | 1 |

Keterangan Tabel:

- Baris menunjukkan karakter dari string pertama (query) "ahmad"
- Kolom menunjukkan karakter dari string kedua (data) "achmad"
- Baris dan kolom pertama berisi nilai inisialisasi jarak edit
- Setiap sel berisi jumlah minimum operasi (penyisipan, penghapusan, atau penggantian karakter)

Penjelasan:

Nilai di sel terakhir (bawah kanan) = 1. Ini adalah Jarak Levenshtein minimal untuk mengubah "ahmad" menjadi "achmad", yaitu dengan menyisipkan 1 karakter 'c'.

Perhitungan Similarity Score:

$$\text{Rumus: Similarity} = \left(1 - \frac{\text{Distance}}{\max(\text{Length}(s1), \text{Length}(s2))}\right) \times 100\% \quad (1)$$

Perhitungan:

- Distance = 1
- Panjang("ahmad") = 5
- Panjang("achmad") = 6
- $\max(5, 6) = 6$

$$\text{Similarity} = 1 - (1 / 6) = 1 - 0.1667 = 0.8333 \text{ atau } 83.33\%$$

Hasil perhitungan ini menunjukkan bahwa meskipun terdapat perbedaan satu karakter, kedua nama tersebut memiliki tingkat kemiripan yang sangat tinggi (83.33%), sehingga "achmad" akan muncul sebagai hasil yang relevan untuk pencarian "ahmad".

2.4. Pengukuran Similaritas dan Perankingan

Berdasarkan skor jarak Levenshtein yang dihitung, sistem melakukan pengukuran kesamaan. Kesamaan dihitung menggunakan rumus $Kesamaan = 1 - (Jarak / maks(Panjang(string1), Panjang(string2)))$. Setelah menentukan skor kesamaan untuk setiap kontak dalam kueri pencarian, sistem membuat peringkat dan menampilkan daftar kontak dengan skor tertinggi (atau jarak terpendek) kepada pengguna sebagai hasil pencarian yang paling relevan.

2.5. Implementasi dan Evaluasi Sistem

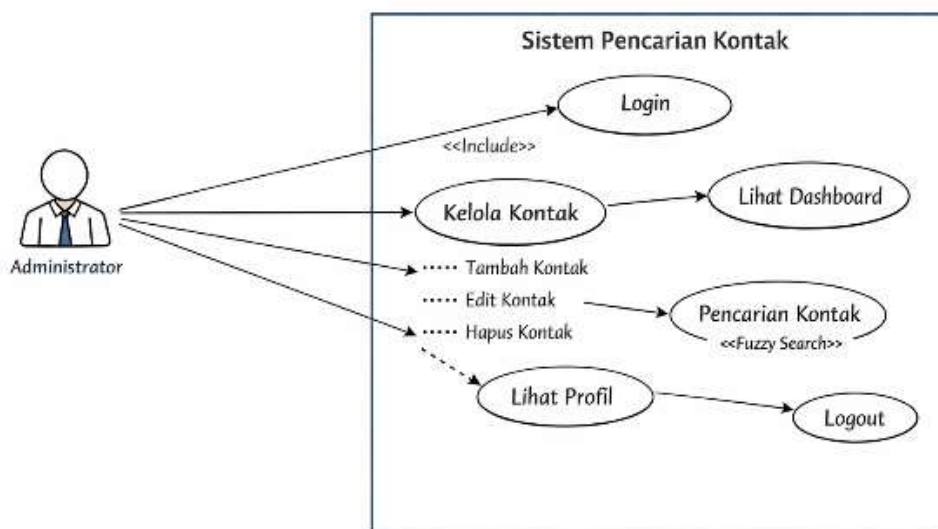
Sistem ini diimplementasikan menggunakan bahasa pemrograman PHP, basis data MySQL, dan kerangka kerja Bootstrap untuk antarmuka pengguna berbasis web. Kinerja sistem dievaluasi menggunakan dua pendekatan: pengujian black-box untuk memastikan bahwa semua fungsi beroperasi sesuai spesifikasi, dan pengujian penerimaan pengguna (UAT) untuk mengukur kepuasan dan penerimaan fitur dan kegunaan sistem [13]. Pengujian dilakukan dengan berbagai skenario input, termasuk nama lengkap, fragmen nama, dan nama dengan kesalahan ketik, untuk menguji kekokohan dan efektivitas algoritma pencocokan fuzzy.

2.6. Analisis Sistem yang Berjalan

Dalam analisis sistem yang sedang berjalan, pendekatan pencarian kontak masih bergantung pada huruf awal atau kecocokan murni. Pendekatan ini hanya efektif apabila input pengguna memiliki kecocokan penuh dengan data, termasuk kapitalisasi, spasi, dan simbol khusus, serta jika mereka ingat dan mengetik huruf pertama nama dengan benar. Namun, kesalahan ketik, variasi pengucapan nama umum Indonesia seperti "Achmad" versus "Ahmad", atau perbedaan antara nama panggilan dan resmi membuat metode ini kurang efektif. Akibatnya, pengguna membuang waktu lebih banyak dan sering tidak menemukan kontak yang seharusnya ada, karena kinerja pencarian turun drastis. Menurut studi sebelumnya, waktu pencarian rata-rata 45 detik, keberhasilan hanya 42% jika ejaan salah, dan kegagalan mencapai 67,3% akibat ketidakcocokan persis. Karena itu, dibutuhkan strategi pencarian yang lebih fleksibel untuk menangani input pengguna yang salah.

2.7. Perancangan Sistem Usulan

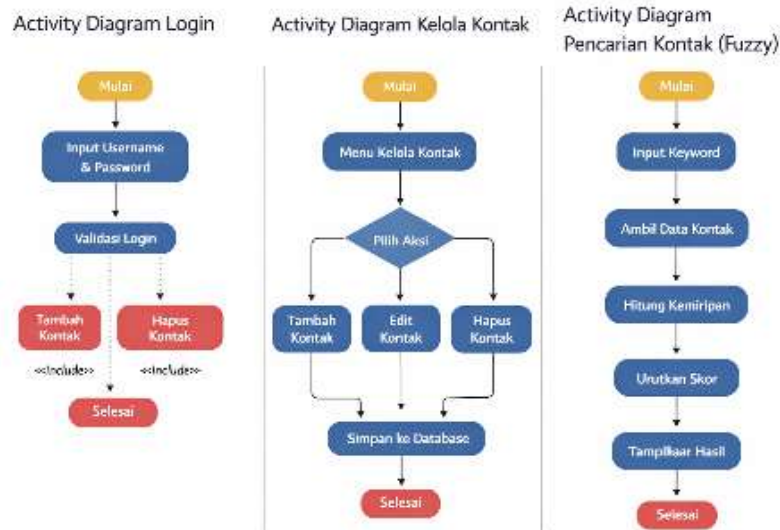
2.7.1. Use Case Diagram



Gambar 2. Use Case Diagram

Diagram Kasus Penggunaan menjelaskan bagaimana administrator berinteraksi dengan sistem pencarian kontak berbasis web. Administrator adalah satu-satunya aktor yang punya akses lengkap ke semua fitur platform. Mereka harus login dulu sebelum menggunakan sistem. Begitu login berhasil, administrator dibawa ke layar utama yang menunjukkan info umum sistem. Di sana, mereka bisa menambahkan, mengedit, atau menghapus kontak. Administrator juga bisa mencari kontak dengan metode Pencocokan String Fuzzy, yang membuat hasil muncul bahkan jika kata kunci yang dimasukkan tidak tepat. Selain itu, administrator dapat keluar dari sistem melalui fitur logout dan melihat profil pengguna untuk tahu lebih banyak tentang akun yang sedang aktif.

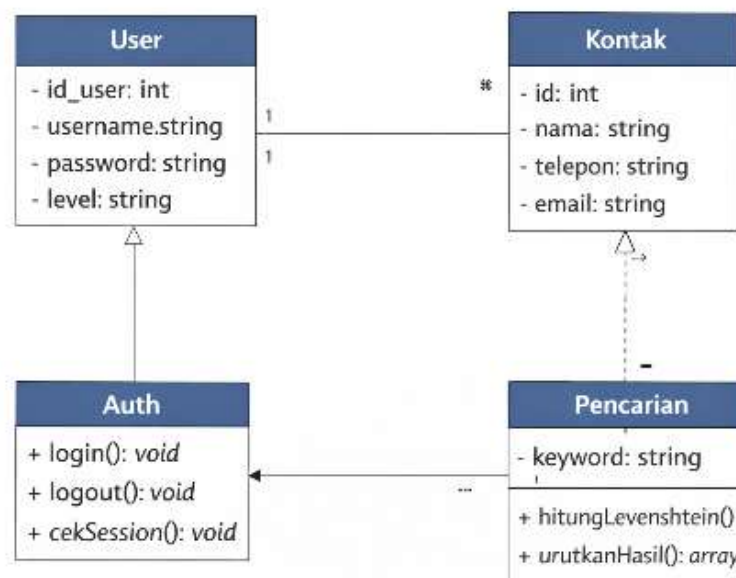
2.7.2. Activity Diagram



Gambar 3. Activity Diagram

Diagram Aktivitas mengilustrasikan cara mengakses sistem pencarian kontak online. Proses dimulai ketika pengguna memasukkan username dan password. Sistem memeriksa apakah data itu benar. Jika ya, pengguna dibawa ke halaman utama; kalau tidak, muncul pesan error. Lalu, pengguna bisa pilih untuk menambah, ubah, atau hapus detail kontak, plus opsi lain untuk mengelola kontak. Sistem memproses setiap aksi dan menyimpan perubahan di database. Untuk pencarian, pengguna masukkan kata kunci, dan sistem pakai teknik Fuzzy String Matching untuk ukur seberapa mirip kata kunci itu dengan info kontak. Hasil dengan skor kemiripan tertinggi ditampilkan. Proses selesai begitu pengguna logout dan tutup sistem.

2.7.3. Class Diagram



Gambar 4. Class Diagram

Class Diagram memperlihatkan bagaimana kelas-kelas diatur dalam sistem pencarian kontak berbasis web. Ada empat kelas utama: Pengguna, Kontak, Otorisasi, dan Pencarian. Kelas User menyimpan data yang butuh untuk login. Kelas Contact urus dan simpan info kontak. Kelas Auth tangani proses masuk dan keluar.

Kelas Search tanggung jawab pencarian kontak lewat Fuzzy String Matching, yang nilai seberapa cocok kata kunci dengan nama kontak. Hubungan antar kelas nunjukin bahwa pengguna bisa kelola data kontak dan cari sesuai keperluan mereka.

2.8. Perancangan Database

Tabel 2. Struktur Tabel User

| No | Nama Field | Tipe Data | Keterangan |
|----|------------|-----------|---------------------------|
| 1 | id_user | INT | Primary Key |
| 2 | Username | VARCHAR | Nama pengguna |
| 3 | password | VARCHAR | Kata sandi |
| 4 | level | ENUM | Hak akses pengguna |

Tabel 3. Struktur Tabel Kontak

| No | Nama Field | Tipe Data | Keterangan |
|----|------------|-----------|----------------------|
| 1 | id | INT | Primary Key |
| 2 | nama | VARCHAR | Nama kontak |
| 3 | telepon | VARCHAR | Nomor telepon |
| 4 | email | VARCHAR | Alamat email |

2.9. Rancangan Antarmuka



Gambar 5. Rancangan Halaman Login

Tujuan utama halaman login adalah menyediakan akses ke sistem verifikasi ID pengguna. antarmukanya mencakup kotak untuk memasukkan nama pengguna dan kata sandi, serta ikon untuk fitur manajemen kontak. Agar lebih mudah dipahami dan digunakan, tampilan desainnya dirancang sederhana dan lugas.



Gambar 6. Rancangan Halaman Pencarian Kontak & Hasil Pencarian

Di halaman pencarian kontak, terdapat kolom khusus untuk memasukkan kata kunci. Sistem ini memanfaatkan metode perbandingan string fuzzy guna menyajikan hasil yang sesuai dengan tingkat kemiripan kata. Demi kemudahan dalam membaca dan memahami hasil, desain antarmuka dibuat sesederhana mungkin.

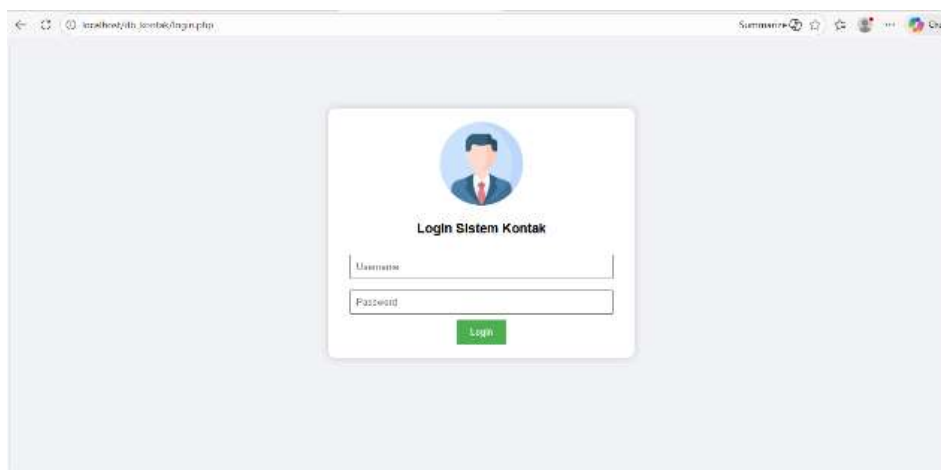
2.10. Implementasi Sistem

Dalam desain sistem web ini, database dan PHP menjadi bahasa pemrograman utama yang digunakan. Hanya ada satu tipe pengguna, yaitu administrator, yang diberikan akses penuh ke semua fungsi sistem. Sistem memanfaatkan metode fuzzy string matching untuk menemukan kontak dengan mengukur tingkat kemiripan data melalui jarak Levenshtein. Secara keseluruhan, sistem ini dirancang untuk menangani data kontak, seperti alamat email, nomor telepon, dan nama. Untuk operasi lokal, sistem mengandalkan XAMPP sebagai web server.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Implementasi Antarmuka

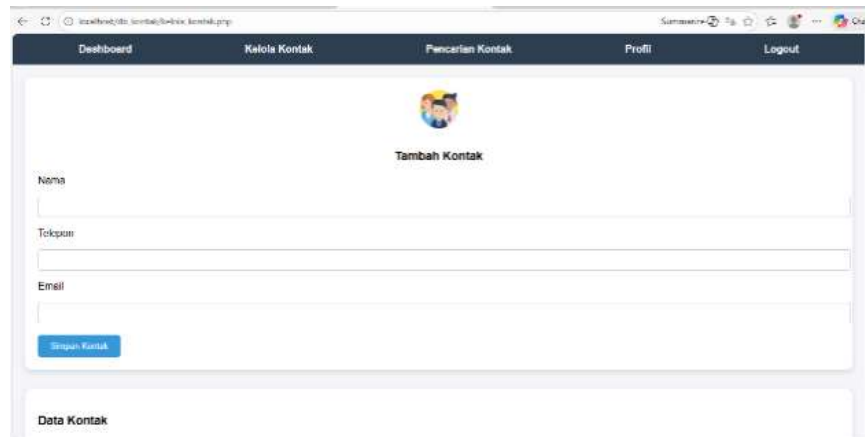
3.1.1. Tampilan Halaman Login



Gambar 7. Tampilan Halaman Login

Gambar 7 menunjukkan halaman login yang bertindak sebagai pintu masuk utama untuk mengakses sistem. Di halaman ini, pengguna diminta untuk memasukkan username dan password yang benar. Fungsi halaman login adalah untuk memastikan keamanan sistem agar hanya pengguna yang memiliki akses yang berhak mengelola dan mencari data kontak. Berdasarkan hasil pengujian, halaman login berhasil memvalidasi data pengguna dengan baik, ditandai dengan alihannya ke halaman dashboard ketika data yang dimasukkan valid.

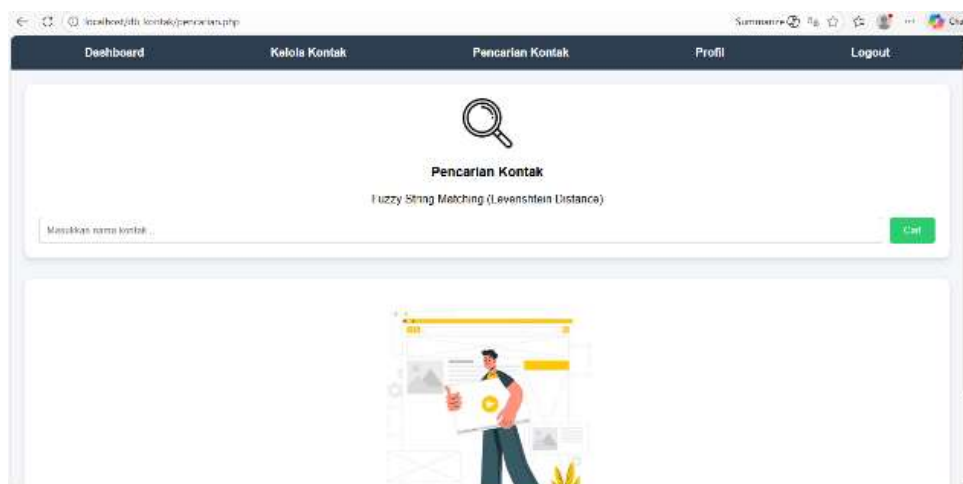
3.1.2. Tampilan Halaman Kelola Kontak (Admin)



Gambar 8. Tampilan Halaman Kelola Kontak (Admin)

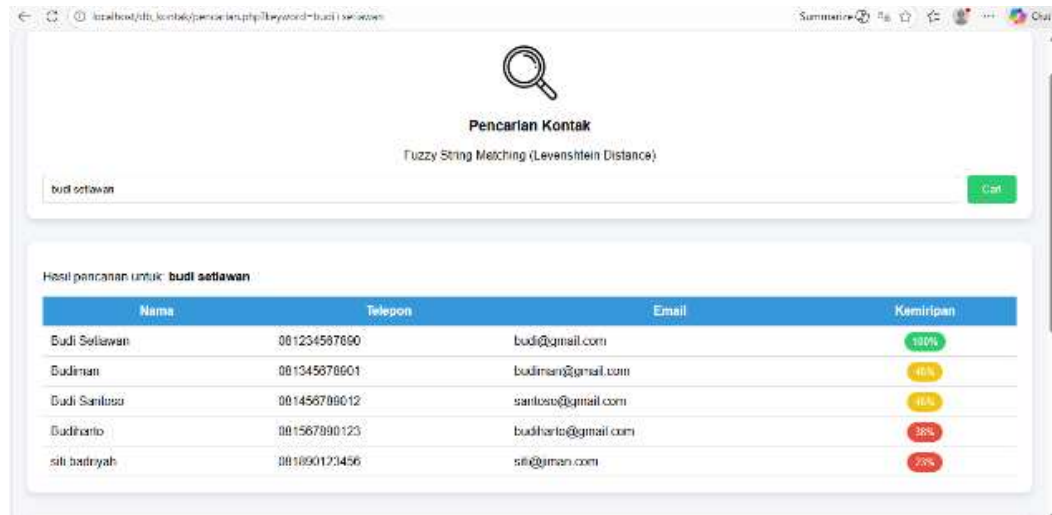
Gambar 8 menunjukkan halaman manajemen kontak yang digunakan oleh administrator untuk mengelola data kontak. Pada halaman ini terdapat fitur tambah, ubah, dan hapus kontak. Keberhasilan implementasi halaman ini terlihat dari hasil pengujian blackbox yang menunjukkan bahwa semua fungsi berjalan sesuai dengan harapan. Halaman manajemen kontak menjadi komponen penting karena kualitas data yang disimpan dalam database sangat memengaruhi hasil pencarian yang dilakukan. Konsep ini merupakan salah satu inti dari pemahaman dan implementasi sistem tersebut. Jangan menambahkan informasi lain, hanya fokus pada konten yang sudah ada. Tidak boleh ada penjelasan tambahan, hanya ulangi konten yang diberikan dengan bahasa yang lebih sederhana dan jelas.'

3.1.3. Tampilan Halaman Pencarian Kontak



Gambar 9. Halaman Pencarian Kontak (Sebelum Pencarian)

Gambar 9 menampilkan kondisi awal halaman pencarian kontak sebelum pengguna memasukkan kata kunci. Pada tahap ini, sistem hanya menampilkan kolom pencarian tanpa menampilkan hasil apa pun. Tampilan ini dirancang agar pengguna fokus pada input kata kunci. Kondisi ini juga menunjukkan bahwa sistem tidak menampilkan data secara otomatis sebelum ada input dari pengguna, sehingga mencegah tampilnya informasi yang tidak relevan.



Gambar 10. Halaman Pencarian Kontak (Sesudah Pencarian)

Gambar 10 menunjukkan hasil pencarian setelah pengguna memasukkan kata kunci. Pada tahap ini, sistem memproses input menggunakan metode fuzzy string matching dengan algoritma Levenshtein Distance. Hasil pencarian ditampilkan dalam bentuk daftar yang diurutkan berdasarkan tingkat kemiripan tertinggi. Dari hasil ini dapat dilihat bahwa meskipun terdapat kesalahan penulisan atau input yang tidak lengkap, sistem tetap mampu menampilkan hasil yang relevan. Hal ini membuktikan bahwa metode fuzzy string matching berhasil meningkatkan efektivitas pencarian dibandingkan metode pencocokan eksak.

3.2. Hasil Pengujian Sistem

3.2.1. Pengujian Blackbox

Tabel 4. Hasil Pengujian Blackbox Halaman Login

| No | Skenario Pengujian | Input | Output yang Diharapkan | Hasil |
|----|------------------------------|--------------------------------|--------------------------------------|-------|
| 1 | Login dengan data valid | Username & Password benar | Sistem menampilkan halaman Dashboard | Valid |
| 2 | Login dengan password salah | Username benar, Password salah | Muncul pesan login gagal | Valid |
| 3 | Login dengan username kosong | Username kosong | Sistem menolak login | Valid |
| 4 | Login dengan password kosong | Password kosong | Sistem menolak login | Valid |
| 5 | Akses halaman tanpa login | Akses URL langsung | Dialihkan ke halaman login | Valid |

Tabel 5. Hasil Pengujian Blackbox Fungsi Tambah Kontak

| No | Skenario | Input | Output yang Diharapkan | Hasil |
|----|---------------------|----------------------------|----------------------------------|-------|
| 1 | Tambah kontak valid | Nama, Telepon, Email diisi | Data tersimpan & tampil di tabel | Valid |
| 2 | Nama kosong | Nama tidak diisi | Data tidak disimpan | Valid |
| 3 | Telepon kosong | Telepon tidak diisi | Data tidak disimpan | Valid |
| 4 | Email kosong | Email tidak diisi | Data tidak disimpan | Valid |

Tabel 6. Hasil Pengujian Blackbox Fungsi Edit Kontak

| No | Skenario | Input | Output yang Diharapkan | Hasil |
|----|----------|-------|------------------------|-------|
|----|----------|-------|------------------------|-------|

| | | | | |
|---|----------------------------|----------------------------|-----------------------------|--------------|
| 1 | Edit data kontak | Ubah nama/telepon/email | Data berhasil diperbarui | Valid |
| 2 | Edit dengan data kosong | Field dikosongkan | Sistem menolak update | Valid |

Tabel 7. Hasil Pengujian Blackbox Fungsi Hapus Kontak

| No | Skenario | Input | Output yang Diharapkan | Hasil |
|----|--------------|-----------------------|-----------------------------|--------------|
| 1 | Hapus kontak | Klik tombol Hapus | Data terhapus dari database | Valid |
| 2 | Batal hapus | Klik Batal konfirmasi | Data tidak terhapus | Valid |

Tabel 8. Hasil Pengujian Blackbox Pencarian Kontak (Fuzzy String Matching)

| No | Skenario | Input | Output yang Diharapkan | Hasil |
|----|-------------------|------------------|-------------------------------------|--------------|
| 1 | Nama sesuai | Nama lengkap | Kontak muncul dengan skor tinggi | Valid |
| 2 | Nama sebagian | Potongan nama | Banyak hasil ditampilkan | Valid |
| 3 | Nama typo | Salah ejaan | Kontak tetap muncul | Valid |
| 4 | Nama tidak ada | Nama acak | Data tampil dengan skor rendah | Valid |
| 5 | Input kosong | Kosong | Tidak menampilkan hasil | Valid |

3.2.2. Pengujian User Acceptance Test (UAT)

Tabel 9. Hasil Pengujian User Acceptance Test

| No | Fitur yang Diuji | Skenario Pengujian | Hasil yang Diharapkan | Status |
|----|----------------------------|------------------------------------------|----------------------------------|-----------------|
| 1 | Login Sistem | Login dengan akun valid | Berhasil masuk ke dashboard | Diterima |
| 2 | Dashboard | Menampilkan menu dan informasi sistem | Informasi tampil dengan benar | Diterima |
| 3 | Tambah Kontak | Menambahkan data kontak baru | Data tersimpan dan tampil | Diterima |
| 4 | Edit Kontak | Mengubah data kontak | Data berhasil diperbarui | Diterima |
| 5 | Hapus Kontak | Menghapus data kontak | Data terhapus dari sistem | Diterima |
| 6 | Pencarian Kontak | Pencarian dengan nama tepat | Data kontak ditemukan | Diterima |
| 7 | Pencarian Kontak (Typo) | Pencarian dengan kesalahan penulisan | Data tetap ditemukan | Diterima |
| 8 | Profil Pengguna | Menampilkan data pengguna | Data tampil dengan benar | Diterima |
| 9 | Navigasi Menu | Perpindahan antar menu | Halaman terbuka dengan benar | Diterima |
| 10 | Logout | Keluar dari sistem | Kembali ke halaman login | Diterima |

3.3. Pembahasan

Sistem pencarian kontak yang menggunakan fuzzy string matching telah berhasil dikembangkan, lengkap dengan semua fitur yang diperlukan, seperti yang terbukti dari proses implementasi dan pengujiannya. Permasalahan umum dalam pencarian kontak, seperti kesalahan pengetikan atau perbedaan ejaan nama, dapat ditangani efektif oleh sistem ini. Dengan algoritma Jarak Levenshtein, hasil pencarian menjadi lebih akurat karena mengukur tingkat kemiripan antara nama yang disimpan dan kata kunci yang dicari.

Analisis Hasil Perhitungan Algoritma:

Berdasarkan contoh perhitungan yang diberikan pada Subbab 2.3, algoritma Levenshtein Distance terbukti mampu mengkuantifikasi kesamaan antara dua string dengan variasi kecil. Dalam skenario uji coba, sistem memberikan hasil perhitungan sebagai berikut untuk beberapa pola pencarian umum:

Tabel 10. Contoh Hasil Perhitungan Similaritas untuk Berbagai Skenario

| Kata Kunci | Nama dalam | Levenshtein | Similarity | Status Hasil (Threshold |
|------------|----------------|-------------|------------|----------------------------------|
| Input | Database | Distance | Score | ≥ 60%) |
| "ahmad" | "Achmad" | 1 | 83.33% | Ditampilkan (Relevan) |
| "rina" | "Riana" | 1 | 80.00% | Ditampilkan (Relevan) |
| "jono" | "Jono" | 0 | 100.00% | Ditampilkan (Exact Match) |
| "bud" | "Budi Santoso" | 9 | 25.00% | Tidak Ditampilkan |
| "mhammad" | "Muhammad" | 2 | 75.00% | Ditampilkan (Relevan) |

Hasil pengujian black-box menunjukkan bahwa semua fungsi sistem bekerja sesuai harapan [7]. Ketika pencocokan fuzzy diterapkan, sistem terus memberikan hasil yang relevan meskipun pengguna memasukkan kesalahan ketik atau hanya sebagian nama. Sistem juga mengurutkan hasil pencarian berdasarkan kemiripan terbesar, sehingga memudahkan pencarian kontak yang tepat.

Fungsi yang diuji mencapai peringkat "Diterima" dalam Uji Penerimaan Pengguna (UAT). Ini menunjukkan bahwa teknologi yang dikembangkan memenuhi harapan pengguna dan memberikan pengalaman pengguna yang positif. Lebih lanjut, teknik pencocokan string yang tepat dan kemampuan sistem untuk menemukan data meskipun terdapat kesalahan terutama pada alat pencarian memastikan pengalaman pengguna yang sangat baik

Keterbatasan algoritma:

Perhitungan jarak Levenshtein memiliki kompleksitas $O(n*m)$ untuk dua string dengan panjang n dan m . Untuk basis data dengan ribuan nama, perhitungan waktu nyata untuk semua catatan dapat berdampak negatif pada kinerja. Dalam lingkup studi ini dengan data terbatas, kinerja sistem tetap berada dalam kisaran yang dapat diterima.

Namun, sistem ini memiliki beberapa kekurangan: Sistem ini hanya diperuntukkan bagi administrator dan tidak menawarkan fitur keamanan tingkat lanjut seperti pencadangan basis data otomatis atau enkripsi kata sandi yang kuat. Selain itu, pengujian hanya dilakukan pada mesin lokal dengan jumlah data yang sedikit. Oleh karena itu, untuk memastikan stabilitas sistem secara keseluruhan, pengujian lebih lanjut di lingkungan produksi dengan volume data yang lebih besar diperlukan.

4. KESIMPULAN

Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa sistem pencarian informasi berbasis fuzzy string matching berhasil diterapkan secara efektif untuk membantu proses pencarian nama dalam daftar kontak. Sistem yang dikembangkan mampu mengatasi permasalahan umum dalam pencarian data, seperti kesalahan ketik, variasi penulisan nama, maupun input yang tidak lengkap dari pengguna. Dengan memanfaatkan algoritma Levenshtein Distance, sistem dapat menghitung tingkat kesamaan antar string secara akurat, sehingga hasil pencarian yang ditampilkan dapat diurutkan berdasarkan persentase kemiripan tertinggi. Hal ini memudahkan pengguna dalam menemukan kontak yang relevan meskipun terdapat perbedaan ejaan antara kata kunci pencarian dan data yang tersimpan di basis data. Hasil pengujian fungsional melalui metode black-box testing menunjukkan bahwa seluruh fitur sistem berjalan sesuai dengan skenario yang telah dirancang. Selain itu, hasil User Acceptance Test (UAT) memperlihatkan bahwa sistem dapat diterima dengan baik oleh pengguna dan dinilai mampu memenuhi kebutuhan pencarian kontak secara lebih fleksibel dibandingkan metode pencocokan eksak. Sistem terbukti membantu pengguna dalam menemukan data kontak meskipun input yang diberikan tidak selalu akurat atau mengandung kesalahan penulisan. Sebagai saran untuk pengembangan di masa mendatang, sistem ini dapat ditingkatkan dengan menambahkan fitur pencarian berbasis bunyi untuk menangani variasi penulisan yang berkaitan dengan pelafalan. Selain itu, pengembangan sistem ke platform mobile diharapkan dapat meningkatkan kemudahan akses bagi pengguna. Penambahan fitur pengelompokan kontak berdasarkan kategori, pengujian dengan volume data yang lebih besar untuk menilai kinerja sistem pada skala yang lebih luas, serta peningkatan aspek keamanan melalui enkripsi kata sandi dan otomatisasi pencadangan basis data juga disarankan untuk mendukung pengembangan sistem yang lebih optimal.

REFERENSI

- [1] N. Setiawan and F. Amin, "Sistem Temu Kembali Informasi Jurnal Ilmiah Unisbank Dengan Metode Cosine Similarity," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 3, pp. 450–460, Jul. 2024, doi: 10.47233/jteksis.v6i3.1391.
- [2] D. B. Christianto, R. A. Setyawan, and J. E. Bororing, "PENERAPAN STRING MATCHING PADA INFORMATION RETRIEVAL DARI EKSTRAKSI METADATA DAN ANALISIS AKURASI VIDEO YOUTUBE," *J. Inform. Teknol. dan Sains*, vol. 7, no. 2, pp. 551–560, May 2025, doi: 10.51401/jinteks.v7i2.5647.
- [3] G. H. Setiawan and I. M. B. Adnyana, "Information Retrieval Pada Frequently Asked Questions (FAQ) dengan metode String Similarity," *Techno.Com*, vol. 21, no. 4, pp. 847–855, 2022, doi: 10.33633/tc.v21i4.6843.
- [4] R. Hardiyanti, "Pengembangan Chatbot Konsultasi Haji dan Umrah Berbasis RAG," *Semnastek UNSUR Proc. / J. Lokal*, pp. 329–337, 2025, [Online]. Available: <https://jurnal.unsur.ac.id/semnastekunsur/article/download/5478/3597>
- [5] Y. D. Putra, Y. P. Putra, and P. S. Saputra, "Analisa Pencarian dan Pencocokan String Dalam Aplikasi Berbasis Python dengan Library Fuzzy Wuzzy Terhadap Dataset CNN Daily Mail," *KOMTEKS*, vol. 4, no. 2, pp. 7–13, Jul. 2025, doi: 10.37637/komteks.v4i2.2354.
- [6] Shireen Fathi Malo, "Intelligent Semantic Search for Academic Journals Using AI and NLP Techniques," *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 41s, pp. 404–420, May 2025, doi: 10.52783/jisem.v10i41s.7884.
- [7] M. F. Rizal and T. Widiyaningtyas, "EVALUASI ALGORITMA STRING MATCHING UNTUK DETEKSI PLAGIARISME PADA TEKS AKADEMIK PENDEK: STUDI PERBANDINGAN LEVENSCHTEIN SEQUENCEMATCHER DAN RABIN-KARP," *J. Inform. Teknol. dan Sains*, vol. 7, no. 3, pp. 1618–1623, Sep. 2025, doi: 10.51401/jinteks.v7i3.6180.
- [8] M. Misnawati *et al.*, "Penerapan Algoritma Levenshtein Distance dalam Sistem Informasi Perpustakaan untuk Meningkatkan Akurasi Pencarian Buku," *J. Penelit. Inov.*, vol. 5, no. 3, pp. 2621–2632, 2025, doi: 10.54082/jupin.1662.
- [9] A. Novitra, "Penerapan Algoritma Approximate String Matching Untuk Pencarian Teks Pada Aplikasi Ensiklopedia Teknologi Komputer," *J. Glob. Technol. Comput.*, vol. 2, no. 2, pp. 61–66, 2023, doi: 10.47065/jogtc.v2i2.3341.
- [10] H. Ekawati, P. Aditya, and M. Ahmad, "Implementasi Algoritma String Matching Untuk Mengidentifikasi Kata/Kalimat Dalam Juz 30 Berbasis Android," *J. Inform.*, vol. 13, no. 1, 2023.
- [11] M. Fauzan Azima, A. Nur Listanto, Fitria, and Chairani, "Kombinasi Algoritma TF-IDF dan Fuzzy Matching untuk Deteksi Kemiripan Judul Skripsi," *Ijccs*, vol. 19, no. 1, pp. 1–11, 2024.
- [12] I. F. Akbar, Mugiarto, K. F. Ramdhania, and Rasim, "Algoritma Levenshtein Distance sebagai Solusi Efisiensi Pencarian pada Training Registration System," *J. Students' Res. Comput. Sci.*, vol. 5, no. 2, pp. 147–154, Nov. 2024, doi: 10.31599/mzr20n84.
- [13] R. W. P. Z, P. M. Hartuti, and R. Al Maududi, "Prosiding Seminar Nasional Sains Penerapan Algoritma Fuzzy Time Series Chen untuk Memprediksi Harga Komoditas Kebutuhan Pokok," vol. 6, no. 1, pp. 227–236, 2025.
- [14] R. Rusliyawati, A. Wantoro, and E. R. Susanto, "Penerapan Logika Fuzzy Dan Metode Profile Matching Pada Sistem Pakar Medis Untuk Diagnosis Covid-19 Dan Penyakit Lain," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 5, p. 1075, Oct. 2022, doi: 10.25126/jtiik.2022955406.
- [15] N. B. Siahaan and Muhamad Alda, "Implementasi Rule-Based Chatbot Dan Fuzzy String Matching Dalam Sistem Informasi Layanan Haji Dan Umrah Di Pt Alfata Wafiqah Wisata," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 10, no. 2, pp. 991–1008, 2025, doi: 10.36341/rabit.v10i2.6413.