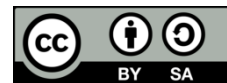


RANCANG BANGUN SISTEM 'UNIMED LOST & FOUND' BERBASIS ALGORITMA LEVENSHTAIN DISTANCE DAN KNUTH-MORRIS-PRATT UNTUK PENCOCOKAN DATA LAPORAN KEHILANGAN BARANG

Khairany Zuhriyyah Jinan Hsb¹, Sebastian Saut Marulitua Sinaga², Muhammad Rafli Wijaya³,
Angelina S. Saragih⁴, Syti Salwaa Nafiisah⁵, Debi Yandra Niska⁶
1,2,3,4) Prodi Ilmu Komputer, Fakultas MIPA, Universitas Negeri Medan, Indonesia

Article Info	ABSTRACT
Article history: Received: 03 Juni 2026 Revised: 03 Juli 2026 Accepted: 16 Juli 2026	Abstrak Kehilangan barang di lingkungan perguruan tinggi merupakan permasalahan yang sering terjadi dan berpotensi menimbulkan ketidakefisienan dalam proses pelaporan serta pencarian barang. Universitas Negeri Medan (UNIMED) sebagai institusi dengan mobilitas civitas akademika yang tinggi membutuhkan suatu sistem terintegrasi yang mampu mengelola data laporan kehilangan dan penemuan barang secara efektif. Penelitian ini bertujuan untuk merancang dan membangun sistem <i>UNIMED Lost & Found</i> yang memanfaatkan algoritma <i>string matching</i> sebagai metode utama dalam melakukan sinkronisasi data laporan kehilangan dan penemuan barang. Algoritma ini digunakan untuk mencocokkan kesamaan informasi berdasarkan atribut teks, seperti nama barang, ciri-ciri, dan lokasi kejadian, sehingga dapat meningkatkan akurasi pencarian dan mempercepat proses identifikasi kecocokan data. Metode pengembangan sistem meliputi tahap analisis kebutuhan, perancangan sistem, implementasi, serta pengujian fungsional menggunakan metode <i>Black Box Testing</i>. Hasil penelitian menunjukkan bahwa penerapan algoritma <i>string matching</i> mampu membantu proses sinkronisasi data secara lebih sistematis dan mengurangi ketergantungan pada pencarian manual, dengan tingkat keberhasilan pengujian mencapai 100% pada seluruh skenario yang diujikan. Sistem yang dibangun diharapkan dapat meningkatkan efisiensi dan efektivitas pengelolaan laporan kehilangan barang di lingkungan UNIMED. Kata Kunci: <i>Lost and Found, String Matching, Sinkronisasi Data, Sistem Informasi, Kehilangan Barang</i> Abstract Loss of personal belongings in a university environment is a common issue that can lead to inefficiencies in reporting and retrieval processes. Universitas Negeri Medan (UNIMED), as an institution with high academic community mobility, requires an integrated system to effectively manage reports of lost and found items. This study aims to design and develop the <i>UNIMED Lost & Found</i> system by implementing a string matching algorithm as the primary method for synchronizing lost and found item reports. The algorithm is utilized to identify textual similarities based on item attributes such as item name, characteristics, and incident location, thereby improving matching accuracy and accelerating the identification process. The system development method consists of requirement analysis, system design, implementation, and functional testing using the Black Box Testing method. The results indicate that the application of the string matching algorithm facilitates more systematic data synchronization and reduces reliance on manual searching, achieving a 100% success rate across all tested scenarios. Therefore, the proposed system is expected to enhance the efficiency and effectiveness of lost item management within the UNIMED environment. Keywords: <i>Lost and Found, String Matching, Data Synchronization, Information System, Lost Items</i> Djtechno: Jurnal Teknologi Informasi oleh Universitas Dharmawangsa Artikel ini bersifat open access yang didistribusikan di bawah syarat dan ketentuan dengan Lisensi



Corresponding Author:E-mail : [sytsalwaanafiisah@gmail.com](mailto:sytisalwaanafiisah@gmail.com)

1. PENDAHULUAN

Perkembangan teknologi informasi mendorong transformasi layanan di perguruan tinggi, tidak hanya pada bidang akademik tetapi juga layanan pendukung. Implementasi sistem informasi terintegrasi terbukti meningkatkan efektivitas pengelolaan data dan mempercepat pengambilan keputusan [1]. Salah satu bentuk layanan yang masih menghadapi berbagai kendala di lingkungan perguruan tinggi adalah pengelolaan laporan kehilangan dan penemuan barang (*lost and found*). Tingginya mobilitas mahasiswa, dosen, tenaga kependidikan, maupun pengunjung kampus menyebabkan potensi terjadinya kehilangan barang pribadi, seperti kartu identitas, telepon genggam, dompet, laptop, kunci kendaraan, maupun dokumen penting, menjadi semakin tinggi [2].

Pada praktiknya, pelaporan kehilangan maupun penemuan barang di sebagian besar perguruan tinggi masih dilakukan melalui media konvensional seperti grup media sosial, aplikasi perpesanan instan, atau papan pengumuman. Mekanisme ini memiliki berbagai keterbatasan: tidak ada basis data terpusat, informasi mudah tertutup pesan baru, sulit ditelusuri kembali, dan belum ada mekanisme otomatis untuk mencocokkan laporan kehilangan dengan laporan barang temuan [3]. Akibatnya, proses identifikasi kepemilikan barang menjadi kurang efektif sehingga memperbesar kemungkinan barang yang telah ditemukan tidak dapat dikembalikan kepada pemiliknya dalam waktu yang relatif cepat [4]

Pengembangan sistem *Lost & Found* berbasis web merupakan salah satu alternatif solusi yang dapat mengatasi permasalahan tersebut melalui penyediaan media pelaporan yang terintegrasi, terdokumentasi, dan dapat diakses secara daring. Akan tetapi, keberhasilan sistem *Lost & Found* tidak hanya ditentukan oleh kemampuan dalam menyimpan dan menampilkan data, melainkan juga oleh kemampuan sistem dalam melakukan pencocokan (*matching*) antara laporan kehilangan dan laporan barang temuan secara cepat serta akurat [5]. Proses pencocokan menjadi tantangan karena informasi yang dimasukkan oleh pengguna umumnya ditulis dalam bentuk teks bebas

(*free text*), sehingga sangat memungkinkan terjadinya variasi penulisan, penggunaan sinonim, singkatan, perbedaan urutan kata, maupun kesalahan pengetikan (*typing error*). Sebagai contoh, barang yang sama dapat dideskripsikan sebagai "dompet kulit hitam", "dompet warna hitam", atau "dompet hitam", yang secara semantik memiliki makna serupa, tetapi secara karakter menghasilkan representasi string yang berbeda [6].

Permasalahan tersebut menunjukkan bahwa pendekatan pencarian berbasis *exact matching* belum mampu menghasilkan proses pencocokan yang optimal karena hanya dapat menemukan data dengan susunan karakter yang identik. Oleh sebab itu, diperlukan penerapan algoritma *string matching* yang mampu meningkatkan ketepatan proses pencarian melalui identifikasi kesamaan maupun kemiripan antarstring. Dalam bidang temu kembali informasi (*information retrieval*), algoritma *string matching* telah banyak dimanfaatkan untuk proses pencarian dokumen, koreksi ejaan, pencarian kata kunci, sistem rekomendasi, hingga pengelolaan arsip digital karena terbukti mampu meningkatkan efisiensi dan akurasi pencarian data berbasis teks [7]. Salah satu algoritma *exact string matching* yang banyak digunakan adalah Knuth-Morris-Pratt (KMP). Algoritma ini melakukan proses *preprocessing* terhadap pola pencarian menggunakan tabel *failure function* sehingga mampu menghindari proses perbandingan karakter yang berulang. Karakteristik tersebut menjadikan algoritma KMP memiliki kompleksitas pencarian yang lebih efisien dibandingkan metode pencarian konvensional, khususnya pada kumpulan data berukuran besar [8]. Berbagai penelitian telah membuktikan bahwa algoritma KMP memberikan performa yang baik pada proses pencarian dokumen digital, pencarian data arsip, maupun sistem temu kembali informasi yang membutuhkan waktu respons cepat [9].

Namun demikian, algoritma KMP hanya bekerja secara optimal ketika pola pencarian memiliki kesamaan karakter secara tepat (*exact matching*), sehingga kurang mampu mengakomodasi kesalahan pengetikan ataupun variasi penulisan yang sering terjadi pada data yang diinput oleh pengguna [10]. Sebagai alternatif, algoritma Levenshtein Distance merupakan metode *approximate string matching* yang menghitung tingkat kemiripan dua buah string berdasarkan jumlah operasi penyisipan (*insertion*), penghapusan (*deletion*), dan penggantian karakter (*substitution*) yang diperlukan untuk mengubah suatu string menjadi string lainnya [11]. Pendekatan ini memungkinkan sistem mengenali kata yang bermakna serupa meskipun terdapat perbedaan penulisan.

Algoritma Levenshtein Distance telah banyak diterapkan pada sistem koreksi ejaan, validasi kata kunci, dan pencarian informasi berbasis teks [12].

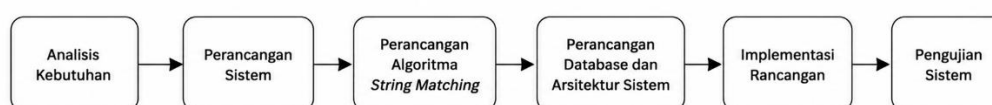
Meskipun berbagai penelitian telah mengimplementasikan algoritma Knuth-Morris-Pratt maupun Levenshtein Distance pada beragam bidang, sebagian besar penelitian masih menerapkan kedua algoritma tersebut secara terpisah sesuai karakteristik permasalahan yang diselesaikan. Penelitian yang menggunakan algoritma KMP umumnya berfokus pada kecepatan pencarian pola, sedangkan penelitian Levenshtein Distance lebih menitikberatkan kemiripan teks, sehingga proses pencocokan belum mengakomodasi kebutuhan sistem yang memerlukan kecepatan sekaligus toleransi variasi penulisan [13]. Selain itu, implementasi kombinasi kedua algoritma tersebut pada sistem *Lost & Found* di lingkungan perguruan tinggi masih sangat terbatas, sehingga efektivitasnya dalam mendukung proses pencocokan laporan kehilangan dan laporan barang temuan belum banyak dikaji [14] [15].

Berdasarkan kondisi tersebut, penelitian ini mengusulkan rancang bangun sistem UNIMED Lost & Found berbasis algoritma Knuth-Morris-Pratt dan Levenshtein Distance untuk meningkatkan kualitas proses pencocokan data laporan kehilangan barang. Pada penelitian ini, algoritma KMP dimanfaatkan untuk mempercepat proses pencarian kandidat data yang relevan, sedangkan algoritma Levenshtein Distance digunakan untuk menghitung tingkat kemiripan deskripsi barang sehingga sistem tetap mampu menghasilkan rekomendasi pencocokan meskipun terdapat perbedaan penulisan pada data yang diinput pengguna. Kebaruan penelitian ini terletak pada integrasi kedua algoritma *string matching* dalam mekanisme pencocokan laporan kehilangan dan laporan barang temuan pada sistem *Lost & Found* berbasis web yang dirancang khusus untuk mendukung kebutuhan pelayanan di lingkungan Universitas Negeri Medan, sehingga diharapkan mampu meningkatkan efektivitas, efisiensi, dan akurasi proses identifikasi kepemilikan barang.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan Research and Development (R&D) yang berorientasi pada perancangan dan pengembangan sistem informasi berbasis web. Pendekatan R&D dipilih karena penelitian tidak hanya bertujuan menganalisis permasalahan, tetapi juga menghasilkan suatu rancangan sistem Lost & Found yang

mampu menyelesaikan permasalahan kehilangan barang di lingkungan Universitas Negeri Medan. Fokus utama penelitian ini adalah merancang sistem *Unimed Lost & Found* yang mampu melakukan sinkronisasi dan pencocokan laporan kehilangan dan temuan barang secara otomatis menggunakan algoritma *string matching*. Implementasi algoritma ini diharapkan dapat meningkatkan akurasi pencarian, mempercepat proses klaim barang, serta mengurangi ketergantungan pada media informal seperti grup WhatsApp atau media sosial. Tahapan penelitian disusun secara sistematis dan dilakukan secara berurutan Adapun tahapan penelitian adalah sebagai berikut.



Gambar 1. Tahapan Penelitian

1. Analisis Kebutuhan

Tahap awal penelitian dilakukan untuk mengidentifikasi permasalahan dan kebutuhan sistem. Analisis dilakukan dengan mengkaji mekanisme pelaporan kehilangan dan temuan barang yang selama ini digunakan di lingkungan Universitas Negeri Medan. Hasil analisis menunjukkan bahwa penyebaran informasi melalui media informal memiliki keterbatasan dari sisi struktur data, jangkauan informasi, serta tidak adanya mekanisme pencocokan otomatis. Pada tahap ini juga dirumuskan kebutuhan pengguna sistem yang meliputi mahasiswa, admin, dan masyarakat kampus, serta kebutuhan fungsional seperti pelaporan barang, autentikasi akun mahasiswa, unggah bukti foto, pencarian, dan klaim barang.

2. Perancangan Sistem

Tahap perancangan sistem bertujuan menyusun gambaran umum sistem yang akan dibangun. Pada tahap ini ditentukan alur kerja sistem, peran dan hak akses pengguna, serta mekanisme interaksi antara pengguna dengan sistem. Perancangan sistem dilakukan agar proses pelaporan, pencarian, dan klaim barang dapat berjalan secara terstruktur dan mudah digunakan.

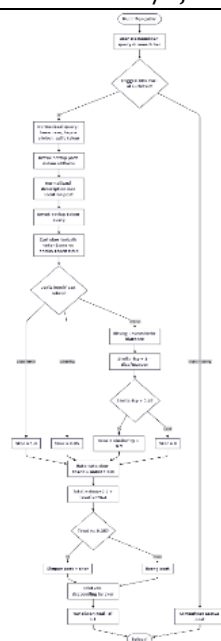
3. Perancangan Algoritma String Matching

Tahap ini difokuskan pada perancangan mekanisme pencocokan data antara laporan kehilangan dan laporan penemuan barang. Pemilihan kedua algoritma yang digunakan dalam penelitian ini didasarkan pada pertimbangan kesesuaian karakteristik masing-

masing algoritma dengan kebutuhan sistem. Untuk fitur pencarian, algoritma Levenshtein Distance dipilih karena kemampuannya menoleransi perbedaan ejaan dan kesalahan ketik melalui perhitungan jarak edit, sehingga lebih unggul dibandingkan algoritma exact matching seperti Boyer-Moore atau Rabin-Karp yang hanya mampu mendeteksi kecocokan persis tanpa toleransi variasi penulisan. Levenshtein Distance juga lebih tepat digunakan dibandingkan metode berbasis vektor seperti TF-IDF atau Cosine Similarity yang memerlukan corpus besar dan tidak optimal untuk pencarian kata tunggal secara real-time. Untuk fitur notifikasi pencocokan otomatis, algoritma Knuth-Morris-Pratt (KMP) dipilih karena kompleksitas waktu linearnya $O(n+m)$ yang lebih efisien dibandingkan algoritma naive string matching dengan kompleksitas $O(n \times m)$, serta lebih ringan secara komputasi dibandingkan pendekatan berbasis machine learning yang memerlukan data latih dan infrastruktur tambahan. Dengan demikian, kombinasi Levenshtein Distance untuk pencarian toleran-kesalahan dan KMP untuk pencocokan pola eksak secara efisien merupakan pilihan yang paling sesuai dengan karakteristik sistem ini. Dua algoritma *string matching* dirancang untuk dua kebutuhan yang berbeda: *Fuzzy Search* berbasis *Levenshtein Distance* untuk fitur pencarian barang, dan *Knuth-Morris-Pratt* (KMP) untuk fitur notifikasi pencocokan otomatis antar laporan.

A. Perancangan algoritma *Fuzzy Search* berbasis *Levenshtein Distance*

Algoritma ini dirancang agar pencarian tetap memberikan hasil yang relevan meskipun terdapat perbedaan ejaan, kesalahan ketik, atau variasi penulisan pada kata kunci yang dimasukkan pengguna. Rancangan alurnya dimulai dari normalisasi kata kunci (mengubah menjadi huruf kecil, menghapus simbol, memecah menjadi token), dilanjutkan dengan perbandingan tiap token terhadap dua atribut laporan, yaitu deskripsi barang (bobot 0,7) dan lokasi kejadian (bobot 0,3). Perbandingan dilakukan bertingkat: kecocokan identik diberi skor tertinggi, kecocokan sebagian (*substring*) diberi skor menengah, dan sisanya dihitung menggunakan jarak edit (*Levenshtein Distance*) untuk memperoleh nilai kemiripan. Skor akhir tiap laporan diperoleh dari akumulasi skor berbobot tersebut, kemudian laporan yang skornya berada di bawah ambang batas disingkirkan, sedangkan sisanya diurutkan dari yang paling relevan. Rancangan alur ini digambarkan pada Gambar berikut.



Gambar 2. Diagram Alir Algoritma Levenshtein Distance

Secara matematis, jarak edit antara dua string a dan b dirancang mengikuti relasi rekurens berikut:

$D(i, 0) = i$, untuk $i = 0 \dots m$ $D(0, j) = j$, untuk $j = 0 \dots n$ $D(i, j) = D(i-1, j-1)$, jika $a[i] = b[j]$ $D(i, j) = 1 + \min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}$, jika $a[i] \neq b[j]$... (1) dengan m dan n masing-masing adalah panjang string a dan b , serta $D(i-1, j)$, $D(i, j-1)$, dan $D(i-1, j-1)$ berturut-turut merepresentasikan operasi penghapusan, penyisipan, dan penggantian karakter.

Nilai kemiripan (*similarity*) antara dua string kemudian dirumuskan sebagai:

$\text{sim}(a, b) = 1 - [D(a, b) / \max(|a|, |b|)]$... (2) di mana $|a|$ dan $|b|$ adalah panjang string a dan b , dengan nilai $\text{sim}(a, b)$ berada pada rentang 0 sampai 1.

Rancangan alur tersebut dituangkan dalam pseudocode berikut:

ALGORITMA FuzzySearch(query, posts, threshold = 0.35)

INPUT : query (kata kunci pencarian), posts (daftar laporan), threshold

OUTPUT : daftar posts yang relevan, terurut menurun berdasarkan skor

1. JIKA query kosong MAKA KEMBALIKAN posts
2. queryTokens $\leftarrow$ Normalisasi(query) // lowercase, hapus simbol, split spasi
3. hasil $\leftarrow$ array kosong
4. UNTUK SETIAP post DALAM posts DO
5. score $\leftarrow$ 0
6. UNTUK SETIAP (field, bobot) DALAM {description: 0.7, location: 0.3} DO
7. fieldScore $\leftarrow$ 0
8. UNTUK SETIAP qt DALAM queryTokens DO
9. best $\leftarrow$ 0

-
10. UNTUK SETIAP tt DALAM Token(field) DO
 11. JIKA $tt = qt$ MAKA nilai $\leftarrow 1.0$
 12. SELAIN JIKA (tt mengandung qt) ATAU (qt mengandung tt) MAKA nilai $\leftarrow 0.85$
 13. SELAIN ITU
 14. $dist \leftarrow \text{LevenshteinDistance}(qt, tt)$ // Persamaan (1)
 15. $sim \leftarrow 1 - dist / \text{MAX}(\text{panjang}(qt), \text{panjang}(tt))$ // Persamaan (2)
 16. nilai $\leftarrow$ JIKA $sim > 0.5$ MAKA $sim \times 0.7$ SELAIN ITU 0
 17. $best \leftarrow \text{MAX}(best, nilai)$
 18. $fieldScore \leftarrow fieldScore + best$
 19. $fieldScore \leftarrow fieldScore / \text{panjang}(\text{queryTokens})$
 20. $score \leftarrow score + (fieldScore \times \text{bobot})$
 21. JIKA $score \geq \text{threshold}$ MAKA TAMBAHKAN ($post, score$) ke hasil
 22. URUTKAN hasil DESCENDING berdasarkan score
 23. KEMBALIKAN daftar $post$ dari hasil

B. Perancangan algoritma *Knuth-Morris-Pratt* (KMP)

Algoritma KMP dirancang untuk mendeteksi kecocokan kata kunci antara judul dan deskripsi laporan kehilangan dengan judul dan deskripsi laporan penemuan milik pengguna lain. Karakteristik utama yang dimanfaatkan dari algoritma ini adalah efisiensi pencocokan melalui *failure function* (tabel LPS, *Longest Prefix Suffix*), yang dibangun terlebih dahulu dari setiap kata kunci sebelum proses pencarian dilakukan. Untuk setiap posisi karakter pada kata kunci, tabel ini menyimpan panjang awalan (*prefix*) terpanjang yang juga merupakan akhiran (*suffix*) dari pola sejauh posisi tersebut. Dengan tabel ini, ketika terjadi ketidakcocokan karakter saat pemindaian, posisi pencarian pada pola tidak perlu diulang dari awal, melainkan langsung melompat ke posisi yang ditunjukkan tabel, sehingga kompleksitas waktu pencarian tetap linear, yakni $O(n+m)$ dengan n adalah panjang teks dan m adalah panjang pola. Rancangan mekanismenya adalah: kata kunci diekstrak dari judul dan deskripsi laporan kehilangan (setelah menyaring *stopword*), kemudian dicari keberadaannya pada teks gabungan judul dan deskripsi tiap laporan penemuan. Jumlah kata kunci yang cocok pada satu pasangan laporan menjadi skor kecocokan pasangan tersebut, yang kemudian menentukan label notifikasi ("Sangat Mirip", "Cukup Mirip", atau "Mungkin Cocok"). Rancangan alur ini digambarkan pada Gambar berikut.



Sebagai ilustrasi, pola tanpa pengulangan karakter seperti "dompet" menghasilkan tabel LPS = [0,0,0,0,0,0], sehingga ketidakcocokan di posisi manapun membuat pencarian direset ke posisi awal. Sebaliknya, pola berpengulangan seperti "abab" menghasilkan tabel LPS = [0,0,1,2], sehingga ketidakcocokan setelah mencocokkan 4 karakter cukup mundur ke posisi 2 (memanfaatkan kecocokan prefix "ab") tanpa perlu reset total. Mekanisme inilah yang membuat KMP tidak pernah melakukan perbandingan mundur pada teks, sehingga kompleksitas waktu tetap $O(n+m)$. Rancangan pembentukan *failure function* dan proses pencocokan tersebut dituangkan dalam pseudocode berikut:

ALGORITMA MatchLostFound(userId)

INPUT : userId (identitas pengguna yang sedang login)

OUTPUT : daftar notifikasi pasangan laporan yang cocok

1. myLost $\leftarrow$ daftar laporan Kehilangan milik userId
2. otherFound $\leftarrow$ daftar laporan Ditemukan milik pengguna lain
3. notifications $\leftarrow$ array kosong
4. UNTUK SETIAP lostPost DALAM myLost DO
5. UNTUK SETIAP foundPost DALAM otherFound DO
6. lostText $\leftarrow$ gabung(judul, deskripsi) dari lostPost
7. foundText $\leftarrow$ gabung(judul, deskripsi) dari foundPost
8. keywords $\leftarrow$ EkstrakKataKunci(lostText) // buang stopword, hapus duplikat
9. matched $\leftarrow$ array kosong
10. UNTUK SETIAP kw DALAM keywords DO
11. table $\leftarrow$ BangunFailureTable(kw)
12. JIKA KMPSearch(foundText, kw, table) = true MAKA
13. TAMBAHKAN kw ke matched
14. JIKA panjang(matched) ≥ 1 MAKA
15. TAMBAHKAN {lostPost, foundPost, matched, score: panjang(matched)} ke notifications
16. URUTKAN notifications DESCENDING berdasarkan score
17. KEMBALIKAN notifications

FUNGSI BangunFailureTable(pattern)

1. table[0] $\leftarrow 0$; prefix $\leftarrow 0$; i $\leftarrow 1$
2. SELAMA i < panjang(pattern) DO
3. JIKA pattern[i] = pattern[prefix] MAKA
4. prefix $\leftarrow$ prefix + 1; table[i] $\leftarrow$ prefix; i $\leftarrow$ i + 1
5. SELAIN JIKA prefix $\neq 0$ MAKA
6. prefix $\leftarrow$ table[prefix - 1]
7. SELAIN ITU
8. table[i] $\leftarrow 0$; i $\leftarrow$ i + 1
9. KEMBALIKAN table

FUNGSI KMPSearch(text, pattern, table)

1. ti $\leftarrow 0$; pi $\leftarrow 0$
2. SELAMA ti < panjang(text) DO
3. JIKA text[ti] = pattern[pi] MAKA
4. ti $\leftarrow$ ti + 1; pi $\leftarrow$ pi + 1
5. JIKA pi = panjang(pattern) MAKA KEMBALIKAN true
6. SELAIN JIKA pi $\neq 0$ MAKA
7. pi $\leftarrow$ table[pi - 1]
8. SELAIN ITU

9. $ti \leftarrow ti + 1$

10. KEMBALIKAN false

4. Perancangan Database dan Arsitektur Sistem

Pada tahap ini dilakukan perancangan struktur basis data dan arsitektur sistem secara teknis. Basis data dirancang menggunakan PostgreSQL melalui Supabase yang mencakup tabel pengguna, laporan kehilangan dan temuan, foto barang, klaim, serta status verifikasi. Selain itu, dirancang pula arsitektur sistem berbasis web dengan pemisahan antara frontend dan backend untuk memastikan keamanan, skalabilitas, dan kemudahan pengelolaan sistem.

5. Implementasi Rancangan

Tahap implementasi dilakukan dengan merealisasikan rancangan sistem ke dalam bentuk aplikasi web. Backend dikembangkan menggunakan layanan Supabase yang menyediakan autentikasi pengguna, basis data, penyimpanan file, serta pengaturan keamanan data. Frontend dikembangkan menggunakan React.js dan Vite untuk membangun antarmuka pengguna yang responsif. Pada tahap ini juga diimplementasikan algoritma string matching untuk fitur pencarian dan notifikasi pencocokan laporan.

6. Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan yang telah dirancang. Pengujian dilakukan menggunakan metode *Black Box Testing*, yaitu pengujian yang berfokus pada kesesuaian antara input dan output sistem tanpa memeriksa struktur kode secara internal. Skenario pengujian dirancang khusus untuk mengevaluasi ketahanan algoritma *Levenshtein Distance* terhadap variasi ejaan dan kesalahan ketik, akurasi algoritma KMP dalam mendeteksi kecocokan kata kunci antar laporan, serta seluruh fungsi utama sistem secara menyeluruh. Pengujian meliputi proses autentikasi pengguna, pembuatan dan pengelolaan laporan, unggah gambar, pencarian berbasis *string matching*, notifikasi kecocokan, serta mekanisme klaim barang. Hasil pengujian digunakan untuk memastikan bahwa sistem mampu mendukung proses pelaporan dan pencarian barang secara efektif dan konsisten.

3. HASIL DAN PEMBAHASAN

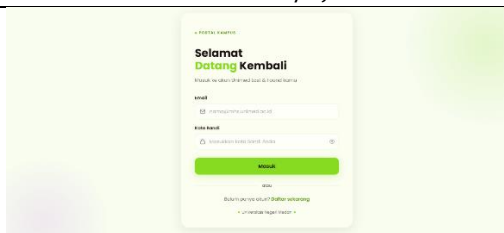
Penelitian ini menghasilkan prototipe sistem *UNIMED Lost & Found* berbasis web yang dapat diakses melalui laman *unimed-lostnfound.vercel.app*. Sistem dirancang untuk mengatasi keterbatasan mekanisme pelaporan barang hilang yang selama ini bergantung pada media informal di lingkungan Universitas Negeri Medan. Pembahasan berikut mencakup hasil studi pendahuluan, perancangan dan implementasi sistem, arsitektur teknologi yang digunakan, serta evaluasi terhadap penerapan algoritma *string matching*.

1. Studi Pendahuluan dan Analisis Kebutuhan

Tahap awal penelitian diawali dengan observasi lapangan terhadap pola pelaporan barang hilang di lingkungan Unimed. Hasil observasi mengonfirmasi bahwa pelaporan masih sepenuhnya bertumpu pada grup pesan instan dan platform media sosial. Pola ini membawa sejumlah kelemahan struktural, antara lain informasi tidak terdokumentasi secara terpusat, jangkauan penyebaran terbatas pada anggota komunitas tertentu, tidak tersedianya mekanisme pencocokan otomatis antara laporan kehilangan dan penemuan, serta sulitnya melakukan penelusuran laporan lama yang sudah tertimpa oleh pesan-pesan baru. Kondisi ini sejalan dengan temuan beberapa penelitian sebelumnya yang menyatakan bahwa penyebaran informasi melalui media informal bersifat tidak terstruktur dan berpotensi menimbulkan kesalahan interpretasi [5][6]. Berdasarkan hasil studi ini, dirumuskan kebutuhan fungsional sistem yang meliputi autentikasi akun mahasiswa, pelaporan kehilangan dan penemuan, unggah foto bukti, pencarian berbasis teks, serta mekanisme klaim dan konfirmasi.

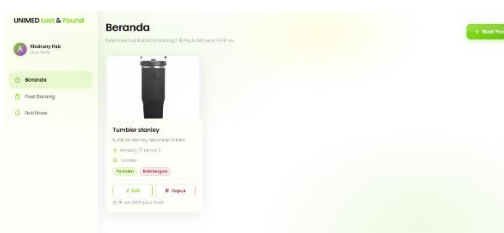
2. Implementasi Sistem

Berdasarkan analisis kebutuhan yang telah dilakukan, tim peneliti merancang dan mengimplementasikan prototipe sistem dengan beberapa fitur utama. Fitur pertama adalah autentikasi pengguna, di mana sistem mewajibkan login menggunakan alamat email institusi (nama@mhs.unimed.ac.id) beserta kata sandi. Mekanisme ini memastikan setiap laporan yang diunggah dapat dipertanggungjawabkan secara identitas sehingga mencegah laporan palsu dan penyalahgunaan layanan. Pendekatan autentikasi berbasis akun institusi seperti ini telah terbukti efektif dalam menjaga keabsahan data pada sistem informasi layanan kampus [4].

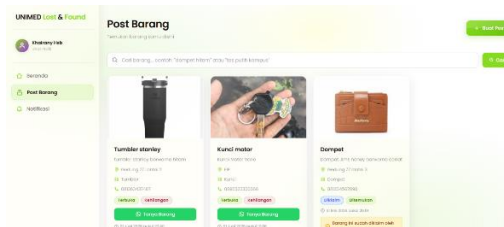


Gambar 2. Tampilan halaman login

Fitur kedua adalah beranda atau *dashboard* pribadi yang menampilkan seluruh laporan milik pengguna yang sedang aktif, dilengkapi dengan foto barang, deskripsi, lokasi, kategori, dan status laporan. Pengguna dapat melakukan operasi pengelolaan laporan secara langsung dari halaman ini. Fitur ketiga adalah halaman *Post Barang* yang menampilkan keseluruhan laporan dari semua pengguna baik kehilangan maupun penemuan dengan informasi yang tersaji dalam format kartu, dilengkapi kolom pencarian teks dan filter status.

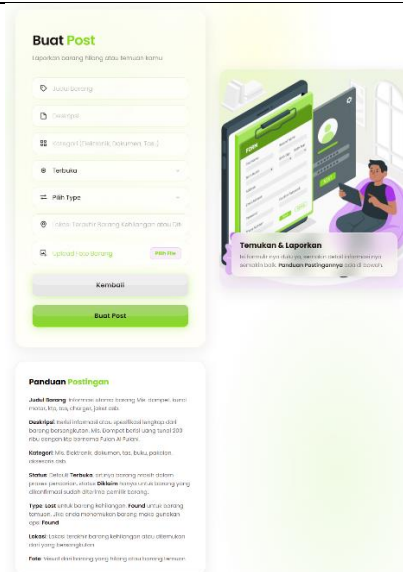


Gambar 3. Dashboard pengguna

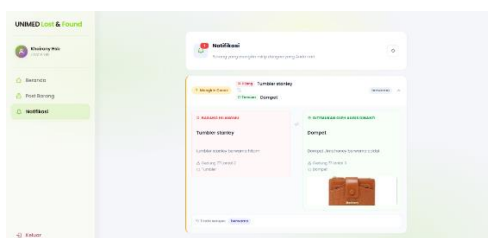


Gambar 4. Fitur post barang dan pencarian

Fitur keempat adalah formulir pembuatan laporan yang memuat sejumlah bidang wajib, yaitu judul barang, deskripsi, kategori, status, tipe laporan (*Lost/Found*), lokasi, dan foto barang. Kelengkapan atribut ini menyediakan informasi yang cukup bagi sistem untuk pencocokan otomatis. Fitur kelima dan paling krusial adalah halaman notifikasi pencocokan otomatis, yang menampilkan pasangan laporan kehilangan dan penemuan yang terdeteksi mirip.



Gambar 5. Fitur buat post



Gambar 6. Fitur notifikasi pencocokan otomatis

3. Arsitektur dan Teknologi Sistem

Sistem dibangun di atas arsitektur berbasis layanan *cloud* dengan pemisahan yang jelas antara sisi *frontend* dan *backend*. Pada sisi *backend*, sistem memanfaatkan Supabase yang menyediakan layanan terintegrasi berupa basis data PostgreSQL, autentikasi pengguna, penyimpanan file, dan kemampuan *realtime* dalam satu platform. Pemanfaatan layanan terintegrasi semacam ini memungkinkan tim pengembang untuk berkonsentrasi pada logika bisnis dan antarmuka tanpa harus membangun infrastruktur dari awal. Sisi *frontend* dikembangkan menggunakan React.js dengan Vite sebagai *build tool*, sementara proses *hosting* dilakukan melalui Vercel yang dihubungkan langsung dengan repositori GitHub sehingga setiap pembaruan kode ter-*deploy* secara otomatis ke server produksi. Pemilihan tumpukan teknologi ini sejalan dengan praktik pengembangan sistem informasi modern yang mengutamakan skalabilitas, kemudahan pengelolaan, dan efisiensi kolaborasi tim [7][8].

Selain fitur-fitur antarmuka, sistem ini juga mengimplementasikan mekanisme pengelolaan data otomatis menggunakan ekstensi *pg_cron* pada Supabase. Mekanisme ini menerapkan tiga aturan penghapusan data secara terjadwal: laporan bertipe *Found*

berstatus *Claimed* dihapus setelah tiga hari, laporan bertipe *Lost* berstatus *Open* dihapus setelah empat belas hari, dan laporan bertipe *Found* berstatus *Open* yang tidak diklaim dihapus setelah dua puluh satu hari. Aturan ini menjaga basis data tetap relevan dan tidak membengkak seiring bertambahnya laporan.

4. Penerapan Algoritma *String Matching*

Dua algoritma utama diimplementasikan dalam sistem ini untuk keperluan yang berbeda. Pada fitur pencarian di halaman *Post Barang*, diterapkan algoritma *Fuzzy Search* berbasis *Levenshtein Distance*. Algoritma ini menghitung jarak edit antara kata kunci yang dimasukkan pengguna dengan data yang tersimpan, sehingga hasil pencarian tetap relevan meskipun terdapat perbedaan ejaan, kesalahan ketik, atau variasi penulisan. Kemampuan toleransi terhadap variasi input ini menjadi keunggulan utama pendekatan *fuzzy matching* dibandingkan pencarian eksak, khususnya pada sistem yang melibatkan input teks bebas dari pengguna awam [13][14].

Pada fitur notifikasi pencocokan otomatis, diterapkan algoritma Knuth-Morris-Pratt (KMP) untuk membandingkan atribut laporan seperti judul barang, deskripsi. Algoritma KMP bekerja secara efisien dengan membangun *failure function* dari pola yang dicari, sehingga proses pencocokan dapat dilakukan tanpa perlu mengulang perbandingan karakter yang sudah diproses. Kompleksitas waktu KMP yang linear, yakni $O(n+m)$ di mana n adalah panjang teks dan m adalah panjang pola, menjadikannya pilihan yang tepat untuk proses pendeteksian kesamaan pola teks secara *real-time* [15]. Apabila tingkat kemiripan antar laporan melampaui ambang batas yang ditetapkan, sistem secara otomatis menampilkan notifikasi bertanda "Mungkin Cocok" yang menghubungkan kedua laporan tersebut.

Berdasarkan hasil pengujian pada subbab 3.5, kedua algoritma terbukti bekerja sesuai rancangan pada seluruh skenario yang diujikan. Meskipun demikian, skenario pengujian masih menggunakan data simulasi dengan variasi kesalahan ketik terbatas pada satu huruf, sehingga ketahanan algoritma terhadap variasi input yang lebih kompleks dari pengguna nyata (singkatan, ejaan tidak baku, kesalahan ketik lebih dari satu huruf) belum sepenuhnya tercakup. Hal ini menjadi arah pengembangan lanjutan, sebagaimana juga disarankan pada penelitian-penelitian terdahulu di bidang sistem informasi berbasis teks [9][11][12].

5. Hasil Pengujian Sistem

Untuk memvalidasi fungsionalitas sistem serta akurasi algoritma *string matching* yang telah diimplementasikan, dilakukan pengujian menggunakan metode *Black Box Testing*, yaitu pengujian yang mengevaluasi kesesuaian antara input yang diberikan dengan output yang dihasilkan sistem tanpa meninjau struktur kode di dalamnya. Pengujian dilakukan dalam tiga tahap: pengujian khusus terhadap algoritma *Fuzzy Search* berbasis *Levenshtein Distance*, pengujian khusus terhadap algoritma KMP pada fitur notifikasi, serta pengujian fungsional menyeluruh terhadap seluruh fitur sistem.

a. Pengujian Algoritma *Fuzzy Search* (Levenshtein Distance)

Tabel 1. Pengujian kolom pencarian Algoritma Fuzzy Search (Levenshtein Distance)

No	Query yang Diketik	Kondisi	Hasil Pengujian
1	dompet	Ejaan benar	Menampilkan post barang yang sesuai
2	dompet kulit	Ejaan benar, 2 kata	Menampilkan post barang yang sesuai
3	dompet kulti	Typo 1 huruf	Menampilkan post barang yang sesuai
4	kunci motor	Ejaan benar	Menampilkan post barang yang sesuai
5	kunci mtor	Typo 1 huruf	Menampilkan post barang yang sesuai
6	hp samsung	Ejaan benar	Menampilkan post barang yang sesuai
7	hp samsug	Typo 1 huruf	Menampilkan post barang yang sesuai
8	KTM	Ejaan benar	Menampilkan post barang yang sesuai
9	KT	Typo 1 huruf	Menampilkan post barang yang sesuai
10	tas ransel	Ejaan benar	Menampilkan post barang yang sesuai
11	tas ransl	Typo 1 huruf	Menampilkan post barang yang sesuai
12	Payung	Ejaan benar	Menampilkan post barang yang sesuai
13	Pyung	Typo 1 huruf	Menampilkan post barang yang sesuai
14	xxxxyy	Kata tidak ada di sistem	Tidak menampilkan post barang apapun karena data yang dicari tidak ada
15	12345	Angka acak	Tidak menampilkan post barang apapun karena data yang dicari tidak ada

Pengujian dilakukan terhadap 15 skenario kata kunci pencarian yang mencakup enam kategori barang berbeda (dompet, kunci, ponsel, kartu tanda mahasiswa, tas, dan payung), masing-masing diuji dalam kondisi ejaan benar dan ejaan yang mengandung kesalahan ketik satu huruf, serta dua skenario tambahan berupa kata kunci acak yang tidak terdapat pada data sistem. Hasil pengujian pada nomor 1 sampai 13 menunjukkan bahwa sistem tetap mampu menampilkan post barang yang relevan meskipun kata kunci mengandung kesalahan ketik, membuktikan bahwa mekanisme *scoring* berbasis *Levenshtein Distance* dengan ambang batas *similarity* 0,5 berhasil mentoleransi variasi ejaan.

Sementara itu, hasil pengujian pada nomor 14 dan 15 menunjukkan bahwa sistem secara konsisten tidak menampilkan hasil apa pun ketika kata kunci yang dimasukkan

benar-benar tidak berkaitan dengan data yang tersimpan, sehingga membuktikan bahwa mekanisme *threshold* skor 0,35 efektif mencegah munculnya hasil pencarian yang tidak relevan (*false positive*). Secara keseluruhan, seluruh 15 skenario pengujian menghasilkan luaran yang sesuai dengan yang diharapkan.

b. Pengujian Algoritma KMP pada Fitur Notifikasi

Tabel 2. Pengujian Notifikasi Algoritma KMP (Knuth-Morris-Pratt)

No	Post Hilang (Akun A)	Post Temuan (Akun B)	Harapan	Hasil Pengujian
1	Dompot kulit coklat	Dompot kulit coklat	Muncul	Muncul
2	Kunci motor Honda Vario	Kunci Honda Vario ditemukan	Muncul	Muncul
3	HP Samsung hitam	HP Samsung hitam ditemukan di kantin	Muncul	Muncul
4	Tas ransel merah	Ransel merah ketinggalan di kelas	Muncul	Muncul
5	KTM	KTM ditemukan di parkir	Muncul	Muncul
6	Laptop HP silver	Laptop HP silver ditemukan di perpustakaan	Muncul	Muncul
7	Payung biru	Payung biru ketinggalan di aula	Muncul	Muncul
8	Jam tangan hitam	Jam tangan hitam	Muncul	Muncul
9	-	Sepatu olahraga ditemukan	Tidak Muncul	Tidak Muncul
10	-	Buku catatan ditemukan di kantin	Tidak Muncul	Tidak Muncul
11	-	Jaket hoodie warna merah	Tidak Muncul	Tidak Muncul
12	-	Charger laptop ditemukan	Tidak Muncul	Tidak Muncul

Pengujian algoritma KMP dilakukan dengan membentuk 12 skenario pasangan laporan antar dua akun pengguna berbeda. Delapan skenario pertama (nomor 1–8) dirancang dengan kata kunci yang saling tumpang tindih antara judul dan deskripsi laporan kehilangan dan laporan penemuan, sehingga notifikasi kecocokan diharapkan muncul. Empat skenario terakhir (nomor 9–12) merupakan laporan penemuan tanpa laporan kehilangan yang berkaitan, sehingga notifikasi tidak seharusnya muncul. Hasil pengujian menunjukkan bahwa seluruh 12 skenario menghasilkan luaran yang sesuai dengan harapan, tanpa satu pun kasus notifikasi yang gagal muncul (*false negative*) maupun notifikasi yang keliru muncul pada pasangan yang tidak berkaitan (*false positive*).

Hasil ini membuktikan bahwa mekanisme pembentukan *failure function* dan proses pencarian pola berbasis KMP yang dirancang pada subbab 2.3 mampu mendeteksi kata kunci yang identik maupun yang muncul sebagai bagian dari kalimat deskripsi yang lebih panjang secara akurat, sekaligus tidak menghasilkan kecocokan keliru pada laporan yang benar-benar tidak berkaitan.

c. Pengujian Fungsional Sistem (*Black Box Testing*)

Tabel 3. Pengujian Fungsional (Black Box Testing)

No	Fitur yang Diuji	Hasil yang Diharapkan	Hasil Aktual
1	Registrasi akun baru	Akun berhasil dibuat dan diarahkan ke halaman login	Akun berhasil dibuat dan diarahkan ke halaman login
2	Login	Berhasil masuk ke dashboard	Berhasil masuk ke dashboard
3	Buat post kehilangan dengan foto	Post tampil di dashboard dan halaman Post Barang	Post tampil di dashboard dan halaman Post Barang
4	Buat post temuan dengan foto	Post tampil di dashboard dan halaman Post Barang	Post tampil di dashboard dan halaman Post Barang
5	Edit post	Perubahan tersimpan dan tampil di card	Perubahan tersimpan dan tampil di card
6	Hapus post	Post terhapus dari daftar	Post terhapus dari daftar
7	Pencarian kata kunci benar	Post yang relevan muncul di hasil pencarian	Post yang relevan muncul di hasil pencarian
8	Pencarian kata kunci typo	Post yang relevan tetap muncul di hasil pencarian	Post yang relevan tetap muncul di hasil pencarian
9	Notifikasi muncul saat ada kecocokan	Kartu notifikasi kecocokan muncul di halaman Notifikasi	Kartu notifikasi kecocokan muncul di halaman Notifikasi
10	Tombol Sudah Ketemu mengubah status	Status post berubah menjadi Diklaim	Status post berubah menjadi Diklaim
11	Edit profil	Data baru tersimpan dan tampil di sidebar	Data baru tersimpan dan tampil di sidebar
12	Ganti password	Password berhasil diubah	Password berhasil diubah
13	Tombol WhatsApp mengarah ke nomor benar	Diarahkan ke WhatsApp nomor pemilik post	Diarahkan ke WhatsApp nomor pemilik post
14	Logout	Sesi berakhir dan diarahkan ke halaman login	Sesi berakhir dan diarahkan ke halaman login

Pengujian fungsional dilakukan terhadap 14 fitur inti sistem, mencakup alur penggunaan dari awal hingga akhir: registrasi dan autentikasi (nomor 1–2), pengelolaan laporan (nomor 3–6), pencarian dan notifikasi berbasis algoritma *string matching* (nomor 7–9), mekanisme klaim barang (nomor 10), pengelolaan akun (nomor 11–12), hingga integrasi dengan layanan eksternal dan sesi pengguna (nomor 13–14). Pengujian dilakukan dengan memberikan input sesuai skenario penggunaan nyata pada antarmuka sistem, kemudian membandingkan luaran yang dihasilkan dengan hasil yang diharapkan tanpa meninjau logika internal kode program.

Hasil pengujian menunjukkan bahwa seluruh 14 fitur menghasilkan luaran aktual yang sepenuhnya sesuai dengan hasil yang diharapkan, termasuk pada fitur nomor 7, 8, dan 9 yang secara langsung melibatkan algoritma *Fuzzy Search* dan KMP. Hal ini menunjukkan bahwa kedua algoritma *string matching* yang dirancang pada subbab 2.3 tidak hanya bekerja dengan baik secara independen sebagaimana dibuktikan pada Tabel

1 dan Tabel 2, tetapi juga terintegrasi secara konsisten dengan keseluruhan alur fungsional sistem.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa penerapan algoritma *string matching* baik *Fuzzy Search* berbasis *Levenshtein Distance* maupun KMP mampu menjadi fondasi yang kuat bagi mekanisme sinkronisasi laporan kehilangan dan penemuan barang secara otomatis. Hal ini didukung oleh hasil pengujian pada subbab 3.5 yang menunjukkan tingkat keberhasilan 100% pada seluruh 41 skenario pengujian, baik pada pengujian algoritma secara khusus maupun pengujian fungsional sistem secara menyeluruh. Sistem yang dihasilkan telah memenuhi kebutuhan fungsional yang diidentifikasi pada tahap analisis dan berpotensi menjadi solusi yang lebih terstruktur, terverifikasi, dan terpercaya dibandingkan mekanisme pelaporan informal yang selama ini digunakan di lingkungan Universitas Negeri Medan [1][10].

4. SIMPULAN

Penelitian ini berhasil merancang dan membangun prototipe sistem *UNIMED Lost & Found* berbasis web yang menjawab permasalahan ketidakefisienan pengelolaan laporan kehilangan dan penemuan barang di lingkungan Universitas Negeri Medan. Sistem yang dikembangkan mampu menyediakan platform terpusat yang menggantikan ketergantungan pada media informal, dengan mekanisme pelaporan yang terstruktur, terverifikasi melalui autentikasi akun mahasiswa, dan terdokumentasi konsisten. Penerapan algoritma *Fuzzy Search* berbasis *Levenshtein Distance* pada fitur pencarian berhasil meningkatkan relevansi hasil pencarian meskipun terdapat variasi ejaan atau kesalahan ketik dari pengguna, sementara algoritma Knuth-Morris-Pratt (KMP) pada fitur notifikasi terbukti mampu mendeteksi kemiripan antar laporan secara otomatis dan efisien berdasarkan kesamaan atribut teks barang dan lokasi, sehingga proses sinkronisasi data laporan kehilangan dan penemuan tidak lagi bergantung pada penelusuran manual, sebagaimana dibuktikan melalui pengujian *Black Box Testing* dengan tingkat keberhasilan 100% pada seluruh skenario yang diujikan. Dengan demikian, kombinasi kedua algoritma *string matching* yang diterapkan telah membuktikan potensinya sebagai fondasi teknis yang kuat dalam sistem informasi layanan kampus berbasis teks, sekaligus membuka peluang pengembangan lanjutan, termasuk pengujian dengan pengguna nyata dan perluasan menuju versi aplikasi mobile.

REFERENCES

- [1] J. Homepage, N. Pratama, H. Ramdani, K. Kurniawan, and A.-S. Arasya, "IJIRSE: Indonesian Journal of Informatic Research and Software Engineering Missing Goods Information System at the Faculty of Science and Technology Using the Prototype Method Sistem Informasi Barang Hilang Di Fakultas Sains dan Teknologi Menggunakan Meto," vol. 4, no. 2, pp. 118–126, 2024.
- [2] S. Muhammad Aziezi, A. Aminuddin, and I. Nuryasin, "Sistem Informasi Pencarian Barang Hilang di Kota Bandung," *Jurnal Repositor*, vol. 2, no. 5, p. 591, 2020.
- [3] L. D. Mufida, "BARANG HILANG DI KABUPATEN BANGKALAN BERBASIS WEBSITE," vol. 15, no. 1, 2026.
- [4] D. Abdurahman and Deffy Susanti, "PERANCANGAN APLIKASI SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN ALGORITMA LEVENSHTAIN DISTANCE DI PERPUSTAKAAN SMA ISLAM AL-MIZAN," *INFOTECH journal*, vol. 9, no. 2, pp. 371–376, Jul. 2023, doi: 10.31949/infotech.v9i2.6263.
- [5] M. I. Rumlatur and F. Ardiani, "Implementasi REST API Media Informasi Kehilangan dan Penemuan Barang Berbasis Android (Studi Kasus: Kampus 1 Universitas Teknologi Yogyakarta)," *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 5, no. 1, pp. 454–463, 2024, doi: 10.35870/jimik.v5i1.510.
- [6] A. Ayu Yandani and A. Voutama, "Implementasi Agile dalam Pengembangan Sistem Informasi Pencarian Barang Hilang Berbasis Website," *Informatics and Digital Expert (INDEX)*, vol. 7, no. 1, pp. 71–76, 2025, doi: 10.36423/index.v7i1.2180.
- [7] N. S. Damanik, P. Afrilia, A. Syahara, and Safrizal, "Sistem Temu Balik Informasi Berbasis Fuzzy String Matching untuk Pencarian Nama Kontak," *Jurnal Ilmu Komputer dan Teknik Informatika*, vol. 2, no. 1, pp. 43–54, Jan. 2026, doi: 10.64803/juikti.v2i1.97.
- [8] G. Gilang Chandra Wijaya, I. N. Tri Anindia Putra, and M. Nia Naryantika, "Perancangan Aplikasi Mobile Lost and Found Di Lingkungan Kampus Menggunakan Metode Design Thinking," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3, 2025, doi: 10.23960/jitet.v13i3.6585.
- [9] A. Y. Priyono, G. Aryotejo, and S. Adhy, "Penerapan Metode Design Thinking untuk Perancangan Prototype Lost and Found," *Jurnal Masyarakat Informatika*, vol. 14, no. 2, pp. 96–107, 2023, doi: 10.14710/jmasif.14.2.52662.
- [10] Julianto, "Analisis Sistem Informasi Dalam Melacak dan Menemukan Keberadaan Barang Yang Hilang Ala Detektif Menggunakan Media Sosial Dan Marketplace Meta (Facebook) Studi Kasus Pada Tindak Kejahatan Kriminal Pencurian," *Jurnal Mahasiswa Teknik Informatika*, vol. 4, no. 2, pp. 65–73, 2025, doi: 10.35473/jamastika.v4i2.2622.
- [11] I. Chandra, E. Irawati, and R. Archie Kosasih, "Aplikasi Web Pencarian Barang Hilang dengan Deteksi Lokasi Berbasis Google Map API," *Prosiding TAU SNARS-TEK Seminar Nasional Rekayasa dan Teknologi*, vol. 2, no. 1, pp. 145–149, 2024, doi: 10.47970/snarstek.v2i1.604.
- [12] D. S. Abyansyah and M. Zakarijah, "Pengembangan Aplikasi Pengaduan Barang Hilang Berbasis Android dengan Deteksi Penipuan," *Journal of Information Engineering and Technology (JIETY)*, vol. 3, no. 1, pp. 28–40, 2025, doi: 10.21831/jiety.v3i1.1261.
- [13] I Putu Riyan Sandhiprasta, Luh Gede Astuti, and I Komang Ari Mogi, "Redesain Aplikasi Pelaporan Barang Hilang Di Bandara (Lost & Found Mobile) Pada Pt. Jasa Angkasa Semesta," *Jurnal Pengabdian Informatika*, vol. 1, no. 1, pp. 119–124, 2022, doi: 10.24843/jupita.2022.v01.i01.p17.
- [14] M. R. Ilham, S. Suwitno, and A. H. Gunawan, "Perancangan Sistem Informasi Pencarian Orang Hilang Berbasis Web dengan Reward Menggunakan Metode Analisis Jaringan Sosial," *Algor*, vol. 4, no. 1, pp. 139–153, 2022, doi: 10.31253/algor.v4i1.1523.
- [15] A. L. Sari and I. M. Suartana, "Perancangan UI / UX Sistem Informasi Lost-And-Found Menggunakan Metode Goal-Directed Design," vol. 07, pp. 719–731, 2026.