

Rancang bangun aplikasi penerjemah bahasa Indonesia – Jawa menggunakan *Levenshtein Distance*

Indonesian – Javanese language translator application design using Levenshtein Distance

Reza Fatkur Rohman*, Abdi Pandu Kusuma

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Islam Balitar, Jl. Imam Bonjol No. 16, Blitar, 66137, Indonesia

E-mail: [*anginsemilir96@gmail.com](mailto:anginsemilir96@gmail.com)

Abstract. *The Indonesian-Javanese language translator application is a technological solution designed to facilitate cross-language communication between the two languages. This study proposes the integration of the Levenshtein Distance method within a web application developed using the Laravel framework. Levenshtein Distance is used to measure the similarity between words in Indonesian and Javanese, taking into account basic operations such as insertion, deletion, and character substitution. The use of the Laravel framework allows for more structured and efficient application development. This application is equipped with additional features such as user management, session handling, and database integration to store and manage the word dictionary and translation results. An intuitive user interface makes it easy for users to input text, select source and target languages, and view translation results directly. Development using Laravel also enables better application scalability and easier code management. Test results show that the Levenshtein Distance algorithm achieves an 80% accuracy rate in suggesting vocabulary, which falls under the "good" category. Meanwhile, results from Beta testing (Closed Beta) showed a percentage score of 72.5%.*

Keywords: *language translator, Javanese, Laravel, Levenshtein Distance*

Abstrak. Aplikasi penerjemah bahasa Indonesia - Jawa merupakan solusi teknologi untuk memfasilitasi komunikasi lintas bahasa antara kedua bahasa tersebut. Penelitian ini mengusulkan integrasi metode *Levenshtein Distance* dalam sebuah aplikasi web yang dikembangkan menggunakan *framework* Laravel. *Levenshtein Distance* digunakan untuk mengukur tingkat kesamaan antara kata-kata dalam bahasa Indonesia dan Jawa, dengan mempertimbangkan operasi dasar seperti penyisipan, penghapusan, dan substitusi karakter. Penggunaan *framework* Laravel memungkinkan pengembangan aplikasi menjadi lebih terstruktur dan efisien. Aplikasi ini dilengkapi dengan fitur-fitur tambahan seperti manajemen pengguna, pengelolaan sesi, dan integrasi basis data untuk menyimpan dan mengelola kamus kata-kata serta hasil terjemahan. Antarmuka pengguna yang intuitif akan memudahkan pengguna untuk memasukkan teks, memilih bahasa sumber dan tujuan, serta melihat hasil terjemahan secara langsung. Pengembangan menggunakan Laravel juga memungkinkan untuk skalabilitas aplikasi yang lebih baik dan pengelolaan kode yang lebih mudah. Hasil pengujian menunjukkan tingkat akurasi dari algoritma *Levenshtein Distance* dalam menyarankan kosakata yaitu sebesar 80% dan berada dalam kategori baik. Sedangkan dari hasil pengujian Beta (*Close Beta*), diperoleh nilai presentase sebesar 72,5%.

Kata kunci: penerjemah bahasa, bahasa Jawa, Laravel, *Levenshtein Distance*

Submitted: 10-10-2024 | *Accepted:* 12-11-2024 | *Published:* 21-04-2025

How to Cite:

R. F. Rohman and A. P. Kusuma, "Rancang bangun aplikasi penerjemah bahasa Indonesia – Jawa menggunakan *Levenshtein Distance*," *Journal of Information System and Application Development (JISAD)*, vol. 3, no. 1, pp. 09-20, 2025, doi: 10.26905/jisad.v3i1.14466.



PENDAHULUAN

Indonesia memiliki lebih dari 700 bahasa daerah yang digunakan secara aktif oleh masyarakatnya. Bahasa Jawa sendiri merupakan salah satu dari bahasa daerah yang paling banyak digunakan dan memiliki pengaruh yang signifikan di Jawa Tengah, Jawa Timur, dan beberapa wilayah lainnya [1]. Bahasa Jawa tidak hanya digunakan sebagai alat komunikasi sehari-hari, tetapi juga menjadi bagian penting dari identitas budaya dan tradisi masyarakat Jawa. Penggunaan bahasa Jawa mencakup berbagai konteks, mulai dari percakapan informal di rumah tangga hingga penggunaan dalam seni dan sastra tradisional [2]. Pengajaran bahasa Jawa telah menjadi bagian integral dari kurikulum di beberapa wilayah di Indonesia, terutama di Jawa Tengah dan Jawa Timur. Materi pengajaran mencakup pemahaman tentang struktur bahasa, kosakata, serta aspek sosial budaya yang terkait dengan penggunaan bahasa tersebut [3].

Meskipun pentingnya bahasa Jawa diakui secara luas, ada beberapa tantangan dalam pembelajaran dan pengajaran bahasa ini. Beberapa tantangan termasuk kurangnya sumber daya pengajar yang memadai, perubahan dalam pola penggunaan bahasa di tengah modernisasi, dan tantangan dalam meningkatkan minat siswa terhadap bahasa daerah di era globalisasi [4]. Perkembangan teknologi informasi juga mempengaruhi cara bahasa Jawa dipelajari dan dipertahankan. Penggunaan aplikasi pembelajaran bahasa, *platform* daring, dan media sosial dapat menjadi sarana yang efektif untuk mempertahankan dan mengembangkan kemampuan berbahasa Jawa [5]. Dalam hal ini, diperlukan pemahaman yang komprehensif tentang pentingnya pembelajaran bahasa Jawa, tantangan yang di hadapi dalam proses pembelajaran, serta upaya-upaya yang dapat dilakukan untuk mempromosikan dan mempertahankan bahasa dan budaya Jawa di masa depan [6].

Oleh sebab itu, muncul kebutuhan akan teknologi yang mudah dan praktis sebagai alternatif dalam pembelajaran bahasa Jawa. Penelitian ini mengusulkan perancangan sebuah aplikasi penerjemah bahasa Indonesia - Jawa menggunakan algoritma *Levenshtein Distance* dengan pengembangan menggunakan sebuah *framework* berbasis Laravel [7], [8]. *Levenshtein Distance*, atau juga disebut *edit distance*, adalah sebuah metode untuk mengukur seberapa berbeda dua rangkaian teks (biasanya *string*) dengan cara menghitung jumlah minimum dari operasi penyisipan (*insertion*), penghapusan (*deletion*), dan substitusi (*substitution*) yang diperlukan untuk mengubah satu *string* menjadi *string* lainnya [9]. Sedangkan Laravel sendiri adalah sebuah *framework* PHP yang sangat populer untuk pengembangan aplikasi web. Dikembangkan pertama kali oleh Taylor Otwell pada tahun 2011, Laravel dirancang untuk menyederhanakan dan mempercepat proses pengembangan web dengan cara menyediakan seperangkat alat fitur yang kuat namun mudah digunakan [10], [11].

Penelitian mengenai implementasi berbagai algoritma untuk mendeteksi kesalahan pengetikan sudah pernah dilakukan sebelumnya. Tingkat akurasi dari algoritma *Levenshtein Distance* dan metode *Regular Search Expression* adalah 67% dalam mendeteksi *typo*, dan 50% dalam menyarankan kata yang mirip dengan *typo* tersebut untuk penulisan paragraf berbahasa Indonesia di *text editor*. *Spell checker* dengan algoritma *Levenshtein Distance* diterapkan di aplikasi penerjemah bahasa Indonesia-Jawa agar dapat memeriksa kesalahan pengetikan dan kemudian merekomendasikan kata yang mungkin dimaksudkan pengguna [12]. Selain itu, algoritma *Levenshtein Distance* dapat mengoptimasi pencarian kata kunci pada kamus aneka bahasa sehingga dapat mengatasi masalah berupa kesalahan pengejaan [13].

Penelitian ini bertujuan sebagai sarana pembelajaran serta untuk memudahkan siswa dalam mempelajari bahasa Jawa menggunakan media aplikasi berbasis web yang akan dibuat. Aplikasi penerjemah ini dilengkapi fitur *spell checker* dengan algoritma *Levenshtein Distance* agar aplikasi dapat menerima dan mengoreksi input *kosakata* dari pengguna yang salah ketik atau tidak sama persis dengan yang ada di basis data. Dalam algoritma *Levenshtein Distance*, pengukuran yang dilakukan merupakan kalkulasi jumlah perbedaan antara dua kata yang berdasarkan jumlah operasi minimum untuk mengganti satu kata ke kata lainnya. Melalui penelitian ini diharapkan dapat meningkatkan kualitas dalam penerapan tutur kata berbahasa serta berperilaku untuk diterapkan di dalam kehidupan sehari-hari [14], [15]. Selain itu, dengan menggunakan aplikasi penerjemah ini diharapkan pembelajaran dalam berbahasa Jawa menjadi lebih mudah dan lancar.

METODE

Penelitian ini dilakukan melalui pengembangan sistem menggunakan metode R&D (*Research and Development*). Tahap pengumpulan data dilakukan melalui observasi, wawancara, dan studi pustaka. Pada tahap perancangan dan pengembangan sistem informasi, dibuat aplikasi berbasis web yang bisa diakses menggunakan jaringan internet di segala *platform* [16]. Selanjutnya, tahap pengujian dilakukan untuk menemukan dan mengidentifikasi sebanyak mungkin *bug* atau masalah yang ada di dalam sistem dan penggunaan aplikasi tersebut.

Teknik Pengumpulan Data

Teknik pengumpulan data berkaitan dengan penentuan kebutuhan pengguna dan pengembang. Pada penelitian ini digunakan beberapa metode untuk mendapatkan data dan informasi yang dibutuhkan, yaitu studi pustaka, observasi, dan wawancara. Studi pustaka dilakukan untuk mengumpulkan segala informasi berkaitan dengan topik yang sedang diteliti dari berbagai sumber. Dalam hal ini, informasi yang harus dikumpulkan adalah untuk mempelajari berbagai karya ilmiah yang relevan dengan aplikasi kamus terjemahan terkait dengan bahasa Jawa.

Tahap observasi dilakukan melalui pengamatan langsung untuk mendapatkan informasi serta membantu memberikan solusi yang tepat. Pengamatan dilakukan di SDN Mandesan 2, yang masih menggunakan sistem manual dalam pembelajaran sehingga masih kurang efektif dalam penyampaian materi kepada para siswa. Sedangkan pada tahap wawancara, informasi dikumpulkan di lapangan berkaitan dengan permasalahan yang terjadi di sekolah SDN Mandesan 2 melalui wawancara langsung dengan salah satu guru bahasa Jawa yang mengajar di sekolah.

Algoritma Levenshtein Distance

Pada algoritma *Levenshtein Distance*, jarak dihitung mulai dari pojok kiri atas sebuah *array* dua dimensi atau matriks yang telah diisi sejumlah karakter *string* awal dan *string* target. *Entry* pada matriks tersebut merepresentasikan nilai terkecil dari transformasi *string* awal menjadi *string* target [17]. Entri yang terdapat pada ujung kanan bawah matriks adalah nilai *distance* yang menggambarkan jumlah perbedaan dua *string*. Langkah-langkah algoritma *Levenshtein Distance* dalam mendapatkan nilai *distance* dijelaskan sebagai berikut.

Jika $S = \text{String awal}$ dan $T = \text{String target}$

Langkah 1 : Inisialisasi

- a) Hitung panjang S dan T , misalkan m dan n
- b) Buat matriks berukuran $0 \dots m$ baris dan $0 \dots n$ kolom
- c) Inisialisasi baris pertama dengan $0 \dots m$
- d) Inisialisasi kolom pertama dengan $0 \dots n$

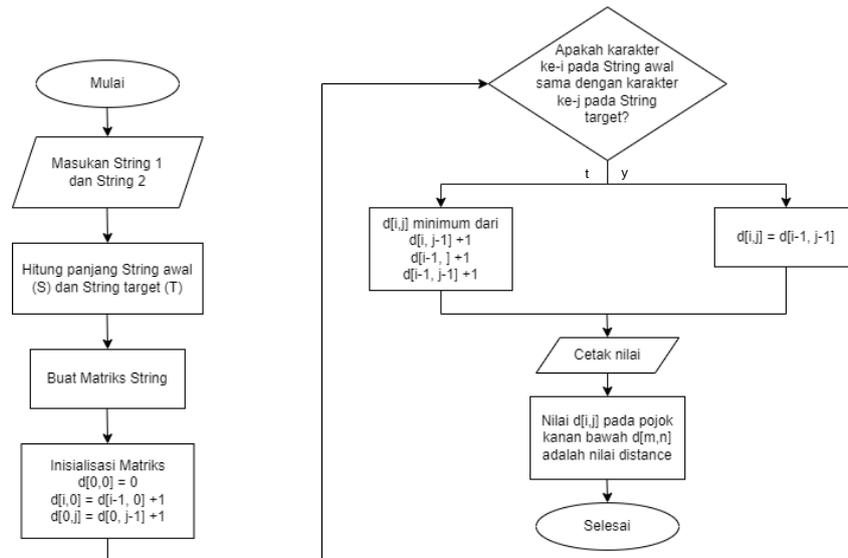
Langkah 2 : Proses

- a) Periksa $S[i]$ untuk $1 < i < m$
- b) Periksa $T[j]$ untuk $1 < j < n$
- c) Jika $S[i] = T[j]$, maka *entry* adalah nilai yang terletak pada tepat diagonal atas kiri, yaitu $d[i,j] = d[i-1, j-1]$
- d) Jika $S[i] \neq T[j]$, maka *entry* adalah $d[i,j]$ minimum dari:
 - Nilai yang terletak tepat di atasnya, ditambah satu, yaitu $d[i, j-1]+1$
 - Nilai yang terletak tepat dikirinya, ditambah satu, yaitu $d[i-1, j]+1$
 - Terletak pada tepat didiagonal atas sebelah kirinya, ditambah satu yaitu $d[i-1, j-1]+1$

Langkah 3 : Hasil entri matriks pada baris ke- i dan kolom ke- j , yaitu $d[i,j]$

Langkah 4 : Langkah 2 diulang hingga *entry* $d[m,n]$ ditemukan

Tahapan dalam proses algoritma perhitungan jarak *Levenshtein Distance* dapat digambarkan dalam bentuk *flowchart* seperti yang diperlihatkan pada Gambar 1. Pada gambar tersebut, ditunjukkan bagaimana algoritma dijalankan, mulai dari memasukkan *string* hingga mendapatkan nilai *entry*.



Gambar 1. Flowchart dari Algoritma *Levenshtein Distance*

HASIL DAN PEMBAHASAN

Analisis Masalah

Masalah-masalah yang terjadi di SDN Mandesan 2 dapat diidentifikasi dengan melaksanakan analisis masalah, dimana masih cukup banyak siswa yang kurang memahami bahasa Jawa. Hal ini membuat komunikasi serta pemahaman dari siswa sendiri dirasa kurang dari segi pembelajaran sehari-hari. Beberapa pertanyaan yang diajukan terkait dengan sistem yang berlaku beserta jawabannya disajikan pada Tabel 1.

Tabel 1. Pertanyaan Terkait Sistem Pembelajaran

No.	Pertanyaan	Jawaban
1.	Bagaimana proses pembelajaran bahasa Jawa di SDN Mandesan 2?	Proses pembelajaran bahasa Jawa di SDN Mandesan 2 menggunakan berbagai metode seperti ceramah, diskusi, permainan bahasa, dan aktivitas interaktif.
2.	Apakah proses pembelajaran sudah efektif?	Bisa dikatakan bahwa proses pembelajaran sudah efektif. Namun, jika ada aspek yang kurang optimal, perlu dilakukan perbaikan dalam metode pendekatan yang digunakan.
3.	Untuk pembelajaran bahasa Jawa memakai materi apa saja?	Untuk materi yang umum digunakan ialah buku teks dan panduan, materi bacaan, buku latihan, sumber daya budaya, dan materi praktik.
4.	Untuk pembelajaran bahasa Jawa diberikan pada kelas berapa saja?	Untuk pembelajaran bahasa Jawa di SDN Mandesan 2 biasanya diberikan dari kelas 1 sampai 6, dimana pada kelas 1 dan 2 meliputi pengenalan dasar dan aktivitas interaktif, kemudian kelas 3 dan 4 meliputi pengembangan keterampilan dan latihan membaca dan menulis, sedangkan di kelas 5 dan 6 meliputi peningkatan kemampuan dan kegiatan budaya.

Analisis Kebutuhan

Analisis kebutuhan dilaksanakan di dalam penelitian agar sistem yang dikembangkan sesuai dengan keinginan dan kebutuhan pengguna. Kegiatan ini dilakukan dengan menganalisis informasi yang didapatkan melalui tahap observasi dan wawancara.

a) Kebutuhan sistem secara umum

- 1) Sistem informasi dibuat agar dapat membantu tugas dari guru dan siswa dalam belajar bahasa Jawa
- 2) Sistem informasi dibuat agar dapat mempermudah siswa untuk belajar mengenal kosakata dalam bahasa Jawa.

- b) Kebutuhan perangkat keras (*Hardware*)
 - 1) PC Desktop
 - 2) RAM 4 GB
 - 3) Koneksi internet
- c) Kebutuhan perangkat lunak (*Software*)
 - 1) Windows 10 Home 64-bit
 - 2) XAMPP versi 3.3.0
 - 3) Visual Studio Code
 - 4) Google Chrome

Perhitungan Jarak *Levenshtein Distance*

Algoritma *Levenshtein Distance* diterapkan pada menu pencarian kosakata dari aplikasi kamus terjemahan sehingga aplikasi dapat menoleransi kesalahan pengetikan dan menyarankan kosakata dari basis data yang paling mirip dengan kosakata yang dimasukkan. Nilai jarak dimulai dari 0 dan dihitung dengan membandingkan karakter berbeda, kemudian nilai jarak bertambah 1. Proses perhitungan jarak terdiri dari perhitungan jarak *string* awal dengan *string* sasaran dilanjutkan perhitungan nilai *similarity*.

Tahap yang pertama yaitu menghitung jarak *string* awal dengan *string* sasaran. Misalkan *string* awal (S) adalah "wedhang" dan *string* sasaran (T) adalah "wedang", mula-mula perhitungan matriks dimulai dari inialisasi urutan karakter pada masing-masing *string*. Untuk mengubah "wedhang" menjadi "wedang" nantinya membutuhkan 1 operasi, yaitu penghapusan huruf "h".

Penjelasan lebih lanjut diperlihatkan pada Tabel 2. Kata "wedhang" memiliki 7 karakter dan "wedang" memiliki 6 karakter. Karakter ke-1 dari keduanya bernilai sama, sehingga nilai matriks yang diberikan sesuai persamaan (1). Jarak karakter ke-1 (s) dengan karakter ke-2 (t) kemudian dihitung dan diketahui bahwa operasi penambahan karakter "e" pada (s) diperlukan, maka nilai matriks yang diberikan sesuai persamaan (2). Jarak karakter ke-1 (s) dengan karakter ke-3 (t) kemudian dihitung dan diperlukan operasi penambahan karakter "e" dan "d" pada (s). Sehingga nilai matriks yang diberikan sesuai persamaan (3).

Tabel 2. Nilai Matriks Setelah Jarak $D[1,1]$ Dihitung

	T	W	E	D	A	N	G
s	D	1	2	3	4	5	6
W	1	0					
E	2						
D	3						
H	4						
A	5						
N	6						
G	7						

$$D[1,1] = D[1-1, 1-1] = 0 \quad (1)$$

$$D[1,2] = D[1, 2-1] + 1 = 0 + 1 = 1 \quad (2)$$

$$D[1,3] = D[1, 3-1] + 1 = 1 + 1 = 2 \quad (3)$$

Perhitungan jarak karakter ke-1 (s) dengan karakter ke-4 hingga ke-6 (t) selanjutnya dilakukan untuk menambahkan karakter "e", "d", "a", "n", dan "g" pada (s), dimana hasilnya dapat dilihat pada Tabel 3. Jarak *Levenshtein Distance* terus dihitung hingga semua nilai pada matriks terisi. Hasil akhir perhitungan jarak setelah jarak $D[7,6]$ dihitung diperlihatkan pada Tabel 4. Jarak *Levenshtein Distance* berada dipojok kanan bawah matriks, yaitu sebesar 1 pada $D[7,6]$. Selanjutnya, kosakata dengan jarak *Levenshtein Distance* yang terkecil dipilih sebagai kosakata yang disarankan, dan akan ditampilkan pada aplikasi.

Tabel 3. Nilai Matriks Setelah Jarak D[1,6] Dihitung

	T	W	E	D	A	N	G
s	D	1	2	3	4	5	6
W	1	0	1	2	3	4	5
E	2						
D	3						
H	4						
A	5						
N	6						
G	7						

Tabel 4. Nilai Matriks Setelah Jarak D[7,6] Dihitung

	T	W	E	D	A	N	G
s	D	1	2	3	4	5	6
W	1	0	1	2	3	4	5
E	2	1	0	1	2	3	4
D	3	2	1	0	1	2	3
H	4	3	2	1	1	2	3
A	5	4	3	2	1	2	3
N	6	5	4	3	2	1	2
G	7	6	5	4	3	2	1

Jarak *Levenshtein Distance* (LD) dari *string* awal dengan beberapa *string* sasaran disajikan pada Tabel 5. *String* sasaran diantaranya kata “Wedang”, “Weding”, “Wedah”, dan “Wedhane”. Jarak terpendek untuk *string* awal “wedhang” diperoleh pada *string* sasaran “wedang” dengan nilai 1.

Tabel 5. Jarak *Levenshtein Distance* *String* Awal dan *String* Sasaran

<i>String</i> awal	<i>String</i> sasaran	Jarak LD
Wedhang	Wedang	1
	Weding	2
	Wedah	3
	Wedhane	2

Tahap yang kedua yaitu menghitung nilai *similarity*. *Similarity* atau kemiripan dari dua *string* dihitung menggunakan persamaan (4), yaitu dengan menghitung jarak LD dan panjang terbesar dari kedua *string*. Nilai *similarity* antara “wedhang” (s) dengan “wedang” (t) dimana jaraknya (d[m,n]) adalah 2 dan panjang “wedhang” serta “wedang” masing-masing adalah 7 (s) dan 6 (t). Dengan menggunakan persamaan (4), nilai *similarity* dari 4 *string* sasaran (t) untuk panjang *string* awal adalah 7 (s), yang dirangkum ke dalam Tabel 6.

$$Sim = \left(1 - \frac{1}{\max [7,6]}\right) \times 100\% = 85,71\% \quad (4)$$

Tabel 6. Hasil Perhitungan *Similarity*

No.	s	t	D[m,n]	Max[s,t]	Sim
1.	Wedang	6	1	7	85,71%
2.	Weding	6	2	7	71,47%
3.	Wedah	6	3	7	57,14%
4.	Wedhane	6	2	7	71,47%

Berdasarkan Tabel 6, ditunjukkan bahwa *string* dengan nilai *similarity* tertinggi pada contoh kosakata yang diberikan untuk disarankan adalah “wedang”, dengan nilai *similarity* 85,72%. Sedangkan nilai *similarity* untuk tiga *string* lainnya berada pada rentang nilai 57% hingga 71%.

Pengujian Kinerja Algoritma *Levenshtein Distance*

Kinerja dari algoritma *Levenshtein Distance* yang diterapkan sebagai *spell checker* pada aplikasi penerjemah bahasa Jawa diukur berdasarkan jumlah kosakata yang disarankan dengan tepat dari 10 kali pengujian sebagai sampel yang dipilih dan diuji. Pada saat penelitian dilakukan, terdapat 90 pasang kosakata bahasa Indonesia - Jawa yang ditambahkan di basis data. Hasil pengujian kinerja algoritma *Levenshtein Distance* dirangkum ke dalam Tabel 7. Hasil pengujian menunjukkan bahwa terdapat dua saran yang tidak sesuai dengan kosakata yang dimaksud oleh pengguna. Hal ini dikarenakan terdapat beberapa *string* sasaran yang memiliki jarak *Levenshtein Distance* dan nilai *similarity* yang sama. Adapun tingkat akurasi algoritma *Levenshtein Distance* dihitung melalui persamaan (5).

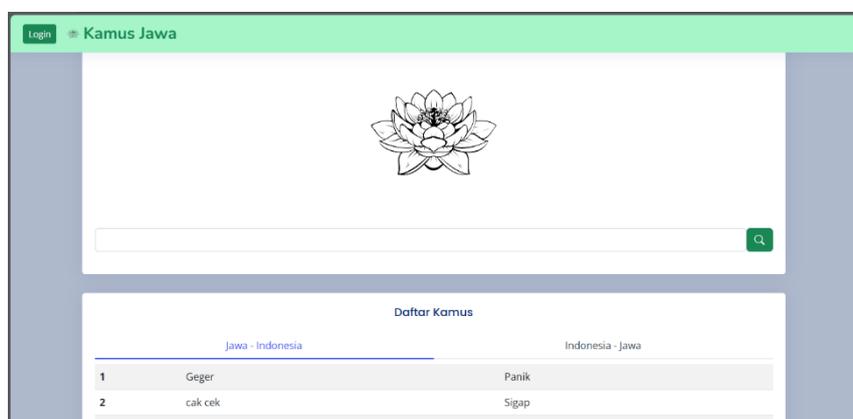
$$Akurasi = \left(\frac{10 - 2}{10} \right) \times 100\% = 80\% \quad (5)$$

Tabel 7. Hasil Pengujian Saran

No.	Input	Target	Jarak LD	Saran
1	Bantheng	Banteng	1	Banteng
2	Wedhang	Weding	1	Wedang
3	Jalan	Jalan	1	Dalan
4	Banyu	Banyu	0	Banyu
5	Sepur	Sepur	0	Sepur
6	Kutho	Kutha	1	Kutha
7	Balek	Balik	1	Balik
8	Ngombe	Ngombe	0	Ngombe
9	Katam	Katon	2	Katon
10	Loro	Lara	1	Lara

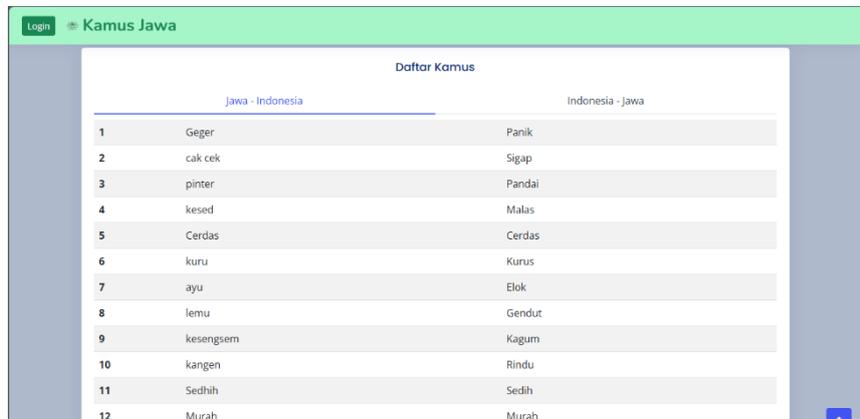
Implementasi Sistem

Pada tahap implementasi sistem, dilakukan penerapan sistem yang sudah dibuat sebelumnya dari tahap perancangan sistem, agar nanti aplikasi dapat dioperasikan tanpa kendala. Dalam aplikasi yang dibuat, terdapat sejumlah tampilan layar yang merupakan tampilan halaman dari aplikasi. Tampilan dari halaman menu utama diperlihatkan pada Gambar 2. Pada menu utama terdapat menu terjemahan bahasa Indonesia - Jawa.

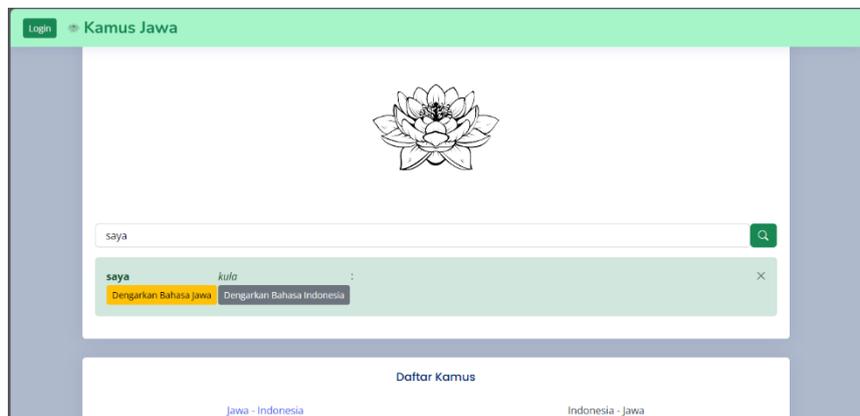


Gambar 2. Tampilan halaman menu utama

Tampilan dari halaman daftar kamus dapat diperlihatkan pada Gambar 3. Pada halaman ini, terdapat daftar kosakata jika ingin mencari secara manual. Tampilan dari halaman menu terjemahan diperlihatkan pada Gambar 4. Pada gambar tersebut ditunjukkan jika kosakata yang dimasukkan sama persis dengan yang ada di basis data, maka terjemahan akan ditampilkan.



Gambar 3. Tampilan Halaman Daftar Kata



Gambar 4. Tampilan Halaman Menu Terjemahan

Pengujian Beta (*Close Beta*)

Pengujian pada aplikasi ini dilakukan oleh 30 responden yang berasal dari ahli IT, ahli materi, serta pengguna akhir yang menggunakan aplikasi. Hasil pengujian didapatkan dengan menjawab 10 pertanyaan yang diberikan kepada responden. Hasil pengujian *Close Beta* untuk pertanyaan ke-1 hingga ke-10 masing-masing diuraikan pada Tabel 8 s/d Tabel 17.

Tabel 8. Skor Jawaban Pertanyaan Ke-1

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Secara keseluruhan, saya merasa puas dengan kemudahan penggunaan aplikasi	Sangat Setuju (SS)	5	1	5
	Setuju (S)	4	3	12
	Netral (N)	3	0	0
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				17

Tabel 9. Skor Jawaban Pertanyaan Ke-2

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Cara penggunaan aplikasi ini sangat mudah	Sangat Setuju (SS)	5	2	10
	Setuju (S)	4	2	8
	Netral (N)	3	0	0
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				18

Tabel 10. Skor Jawaban Pertanyaan Ke-3

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Saya dapat memenuhi kebutuhan belajar Bahasa jawa dengan lebih mudah Ketika menggunakan aplikasi ini	Sangat Setuju (SS)	5	1	5
	Setuju (S)	4	2	8
	Netral (N)	3	0	0
	Tidak Setuju (TS)	2	1	2
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				15

Tabel 11. Skor Jawaban Pertanyaan Ke-4

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Saya dapat memenuhi kebutuhan belajar Bahasa jawa dengan cepat Ketika menggunakan aplikasi ini	Sangat Setuju (SS)	5	1	5
	Setuju (S)	4	2	8
	Netral (N)	3	0	0
	Tidak Setuju (TS)	2	1	2
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				15

Tabel 12. Skor Jawaban Pertanyaan Ke-5

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Saya dapat memenuhi kebutuhan belajar Bahasa jawa dengan menggunakan aplikasi ini	Sangat Setuju (SS)	5	1	5
	Setuju (S)	4	2	8
	Netral (N)	3	1	3
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				16

Tabel 13. Skor Jawaban Pertanyaan Ke-6

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Saya merasa nyaman menggunakan aplikasi ini	Sangat Setuju (SS)	5	1	5
	Setuju (S)	4	2	8
	Netral (N)	3	1	3
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				16

Tabel 14. Skor Jawaban Pertanyaan Ke-7

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Aplikasi ini sangat mudah dipelajari	Sangat Setuju (SS)	5	1	5
	Setuju (S)	4	2	8
	Netral (N)	3	1	3
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				16

Tabel 15. Skor Jawaban Pertanyaan Ke-8

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Saya merasa lebih produktif Ketika menggunakan aplikasi ini	Sangat Setuju (SS)	5	1	5
	Setuju (S)	4	2	8
	Netral (N)	3	0	0
	Tidak Setuju (TS)	2	1	2
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				15

Tabel 16. Skor Jawaban Pertanyaan Ke-9

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Jika terjadi error, aplikasi ini memberikan pesan kesalahan yang jelas dan memberitahu cara mengatasi	Sangat Setuju (SS)	5	0	0
	Setuju (S)	4	0	0
	Netral (N)	3	3	9
	Tidak Setuju (TS)	2	1	2
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				11

Tabel 17. Skor Jawaban Pertanyaan Ke-10

Pertanyaan	Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Informasi yang disediakan aplikasi ini sangat jelas	Sangat Setuju (SS)	5	0	0
	Setuju (S)	4	2	8
	Netral (N)	3	0	0
	Tidak Setuju (TS)	2	2	4
	Sangat Tidak Setuju (STS)	1	0	0
Total skor				12

Berdasarkan nilai keseluruhan, didapatkan bahwa nilai ideal seluruh pertanyaan adalah 200, dengan nilai persentase yang dihitung menggunakan persamaan (6). Sehingga didapatkan nilai persentase untuk masing-masing kategori jawaban seperti yang diperlihatkan pada Tabel 18.

$$\text{Nilai Persentase } (P) = \frac{\text{Total Nilai}}{200} \times 100\% \quad (6)$$

Tabel 18. Total Skor Seluruh Pertanyaan

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor
Sangat Setuju (SS)	5	9	45
Setuju (S)	4	17	68
Netral (N)	3	6	18
Tidak Setuju (TS)	2	6	12
Sangat Tidak Setuju (STS)	1	0	0
Total			145

Dengan menggunakan persamaan nomor (6), maka diperoleh nilai persentase sebesar 72,5% untuk hasil kuesioner Beta (*Close Beta*). Hal ini menunjukkan bahwa responden cukup puas dengan aplikasi yang telah dibuat.

$$\text{Nilai Persentase (P)} = \frac{145}{200} \times 100\% = 72,5\% \quad (6)$$

SIMPULAN DAN SARAN

Berdasarkan hasil analisis, perancangan, implementasi, dan pengujian yang telah dilakukan, aplikasi ini mampu menerjemahkan kosakata dari bahasa Indonesia ke bahasa Jawa dengan baik, sehingga bermanfaat dalam membantu komunikasi serta mendukung proses belajar siswa sekolah dasar yang belum memahami bahasa Jawa. Tingkat akurasi yang tinggi dari algoritma *Levenshtein Distance* dalam menyesuaikan kosakata menunjukkan efektivitasnya dalam menangani kesalahan kata yang dimasukkan. Aplikasi yang dibuat juga telah melalui pengujian Beta yang melibatkan berbagai pengguna dan mendapatkan nilai yang baik sehingga layak untuk diimplementasikan. Untuk penelitian selanjutnya, aplikasi dapat dikembangkan dengan memanfaatkan algoritma pencarian lain atau menggabungkannya dengan *Levenshtein Distance* untuk meningkatkan akurasi. Selain itu, aplikasi sebaiknya menyarankan kosakata dari basis data ketika terjadi kesalahan ketik, serta dapat menerima input berupa kalimat lengkap dan mampu menampilkan makna kata yang diterjemahkan secara kontekstual untuk mencegah miskomunikasi.

DAFTAR PUSTAKA

- [1] I. W. Budiarta, "Eksistensi Bahasa Lokal Terkait Pindahan Ibu Kota Negara Ke Kalimantan Timur: Ancaman dan Strategi Pemertahanannya," *J. Ilmu Sos. Dan Hum.*, no. 1, pp. 1–9, 2019.
- [2] E. S. Utami and D. Zustiyanoro, "Lanskap Pembelajaran Bahasa Jawa Masa Pandemi Covid-19," *Piwulang J. Pendidik. Bhs. Jawa*, vol. 10, no. 1, pp. 44–57, 2022, doi: 10.15294/piwulang.v10i1.54176.
- [3] M. I. Maulana, "Menurunnya Minat Tulisan Aksara Jawa di Lingkup Mahasiswa Sejarah Peradaban Islam UIN Sunan Ampel Surabaya," *HEURISTIK J. Pendidik. Sej.*, vol. 3, no. 2, pp. 62–76, 2024, doi: 10.31258/hjps.3.2.62-76.
- [4] S. Ratnawati, "Penerapan Komunikasi bahasa jawa pada Anak Usia Dini," *J. Educ.*, vol. 06, no. 03, pp. 17164–17171, 2024.
- [5] V. N. Erika and A. N. Azizah, "Peran Pengajaran Bahasa Jawa Dalam Mempertahankan Warisan Kebudayaan Bahasa Jawa," *Wijayakusuma Natl. Conf.*, November 2023, pp. 191–195, 2023.
- [6] T. Aprilianto and A. Badawi, "Sistem Koreksi Kata dan Pengenalan Struktur Kalimat Berbahasa Indonesia dengan Pendekatan Kamus Berbasis Levenshtein Distance," *SPIRIT*, vol. 9, no. 1, pp. 48–61, 2018, doi: 10.53567/spirit.v9i1.72.
- [7] M. B. Ramadhan, I. Magdalena, K. Setyawati, G. Tiranita, and M. Rohmadi, "Kajian Sosiopragmatik Wacana Lisan Bahasa Jawa di Provinsi Yogyakarta dalam Upaya Menjaga Kearifan Bahasa Lokal (Indigenous Language)," *Kode J. Bhs.*, vol. 13, no. 1, pp. 5162–5174, 2024, doi: 10.24114/kjb.v13i1.56849.
- [8] A. A. Ghirrid, R. T. K. Sari, and R. T. Aldisa, "Algoritma Natural Language Processing Untuk Aplikasi Penerjemah (Indonesia–Jawa) Menggunakan Metode Speech Processing," *Jurnal JTik (Jurnal Teknologi*

Informasi dan Komunikasi, vol. 8, no. 3, pp. 746-759, 2024.

- [9] M. R. Batisya and K. Jevanda BS, "Implementasi Algoritma Levenshtein Distance Untuk Misspelled Word Pada Pencarian Lagu Melayu," *J. Inform.*, vol. 10, no. 1, pp. 72-78, 2023, doi: 10.31294/inf.v10i1.15208.
- [10] I. G. Handika and A. Purbasari, "Pemanfaatan Framework Laravel Dalam Pembangunan Aplikasi E-Travel Berbasis Website," *Konf. Nas. Sist. Inf.*, pp. 1329-1334, 2018.
- [11] Z. Subecz, "Web-development with Laravel framework," *Gradus*, vol. 8, no. 1, pp. 211-218, 2021, doi: 10.47833/2021.1.csc.006.
- [12] Y. P. Sari, G. A. Pradnyana, and I. M. A. Wirawan, "Pengembangan Aplikasi Kamus Bahasa Bima - Bahasa Indonesia Menggunakan Algoritma Levenshtein Distance Sebagai Spell Checker Berbasis Android," *Kumpul. Artik. Mhs. Pendidik. Tek. Inform.*, vol. 8, no. 2, pp. 86-95, 2019, doi: 10.23887/karmapati.v8i2.17964.
- [13] B. K. S. Arnawa, "Optimasi Pencarian Kata Pada Kamus Aneka Bahasa Menggunakan Algoritma Levenshtein Distance," *J. Sist. dan Inform.*, vol. 12, no. 2, pp. 149-157, 2018.
- [14] N. A. Sholikhah and H. Subrata, "Pengembangan Media Javanese Monopoly (Javanopoly) untuk Pembelajaran Unggah-Ungguh Bahasa Jawa Peserta Didik Kelas IV Sekolah Dasar," *J. Penelit. Pendidik. Guru Sekol. Dasar*, vol. 12, no. 4, pp. 486-500, 2024.
- [15] U. Nadhiroh, "Peranan Pembelajaran Bahasa Jawa Dalam Melestarikan Budaya Jawa," *JISABDA J. Ilm. Sastra dan Bhs. Daerah*, vol. 3, no. 1, pp. 1-10, 2021, doi: 10.26877/jisabda.v3i1.9223.
- [16] Okpatrioka Okpatrioka, "Research And Development (R&D) Penelitian Yang Inovatif Dalam Pendidikan," *Dharma Acariya Nusant. J. Pendidikan, Bhs. dan Budaya*, vol. 1, no. 1, pp. 86-100, 2023, doi: 10.47861/jdan.v1i1.154.
- [17] N. Aprilia, "Implementasi Algoritma Levenshtein Distance untuk Mengoreksi Kesalahan Kata pada Aplikasi Arsip Surat," Thesis, Universitas Mercu Buana Jakarta, 2023. [Online]. Available: <https://repository.mercubuana.ac.id/82338/>