
Implementasi Algoritma *Levenshtein Distance* pada Aplikasi Kamus Terjemahan Bahasa Indonesia – Bahasa Bangka

Adi Ardiansyah

Mega Mart Sungailiat

E-mail: megamart555@gmail.com

Abstrak

*Indonesia merupakan negara dengan banyak bahasa daerah dimana salah satunya adalah bahasa Bangka yang termasuk dalam bahasa rumpun Melayu yang digunakan sebagai media komunikasi oleh masyarakat Bangka. Cukup banyak masyarakat Bangka yang kurang memahami bahasa Bangka, khususnya yang tinggal di perkotaan dan pendatang dari luar Bangka. Selain itu, alat untuk menerjemahkan bahasa Indonesia ke bahasa Bangka atau sebaliknya yang beredar di masyarakat jumlahnya masih terbatas. Hal ini menyebabkan miskomunikasi dan sulitnya komunikasi terjalin antara masyarakat Bangka yang kurang fasih berbahasa Indonesia dengan masyarakat yang kurang memahami bahasa Bangka. Aplikasi Android kamus terjemahan bahasa Indonesia ke Bangka atau sebaliknya akan dibuat untuk mengatasi permasalahan ini. Aplikasi usulan dilengkapi fitur spell checker dengan algoritma *Levenshtein Distance* agar dapat menerima dan mengoreksi input kosa kata yang salah ketik atau tidak sama persis dengan yang ada di basis data. Dengan menggunakan aplikasi usulan, komunikasi antara masyarakat Bangka yang kurang fasih berbahasa Indonesia dengan masyarakat dari luar Bangka menjadi lebih mudah dan lancar. Adapun tingkat akurasi dari algoritma *Levenshtein Distance* dalam menyarankan kosa kata yang mungkin dimaksudkan oleh pengguna saat ada kesalahan pengetikan adalah sudah baik, yaitu sebesar 83,33%.*

Kata Kunci— *Bahasa Bangka, Kamus, Levenshtein Distance, Spell Checker, Terjemahan*

Abstract

*Indonesia is a country with many regional languages, where one of them is Bangka language which is part of the Malay language family which is used as a medium of communication by the Bangka society. Many people in Bangka lack of understanding of Bangka language, especially those who live in urban areas and immigrants from outside of Bangka. Moreover, the number of tools to translate Indonesian to Bangka or vice versa circulating among the public is still limited. This causes miscommunication and difficulty in establishing communication between Bangka people who are less fluent in Indonesian and people who do not understand Bangka language well enough. An Indonesian to Bangka or vice versa translation dictionary Android application will be created to overcome this problem. The proposed application is equipped with spell checker feature using *Levenshtein Distance* algorithm so that it can receive and correct vocabulary input that has typos or does not exactly match those in the database. By utilizing the proposed application, communication between Bangka people who are not fluent in Indonesian and people from outside of Bangka become easier and smoother. The accuracy level of *Levenshtein Distance* algorithm in suggesting vocabulary that might be intended by the user when there is a typing error is good, namely 83.33%.*

Keywords— *Bangka Language, Dictionary, Levenshtein Distance, Spell Checker, Translation*

1. PENDAHULUAN

Semua orang memerlukan komunikasi untuk menjalankan aktivitas sehari-hari mereka, salah satunya dengan memakai bahasa. Bahasa adalah sarana komunikasi yang paling efektif untuk menyampaikan ide, gagasan, maksud, serta tujuan ke pihak lain. Bahasa memungkinkan kerja sama terjalin antara pihak satu dengan lainnya dalam kehidupan bermasyarakat untuk menyelesaikan sebuah pekerjaan. Sebuah kosakata yang penulisannya sama dapat memiliki makna yang berbeda tergantung asal usul bahasa. Misalkan, kata “sarang” dalam bahasa Korea berarti cinta, sedangkan dalam bahasa Indonesia berarti kandang [1].

Indonesia pada tahun 2022 terdiri atas 38 provinsi yang meliputi ribuan pulau dengan ras, agama, suku, budaya, dan bahasa daerah dari penduduknya yang sangat beragam [2]. Provinsi Kepulauan Bangka Belitung adalah provinsi ke-31 di Indonesia sejak tahun 2000 dengan bahasa mayoritas daerahnya adalah bahasa Bangka yang termasuk dalam bahasa rumpun Melayu [3]. Berdasarkan asal wilayah penduduknya, bahasa Bangka terdiri dari beberapa dialek utama seperti dialek Pangkalpinang, Sungailiat, Toboali, Koba, dan Mentok. Dialek-dialek ini memiliki variasi yang sama namun dengan kosakata yang perbedaannya tidak signifikan karena terjadi hanya pada tataran fonem. Perbedaan kosakata secara signifikan hanya ditemukan di daerah pedesaan seperti Sadai, Air Gegas, dan Kelapa. Sebagai contoh kata “kamu” digantikan “ka” atau “ki” untuk daerah perkotaan dan “pok” untuk daerah pedesaan [4]. Adapun dialek yang digunakan pada penelitian ini adalah dialek Sungailiat.

Untuk menerjemahkan kosa kata dari suatu bahasa ke bahasa lainnya dibutuhkan kamus terjemahan. Proses pencarian kosa kata menurut abjad satu persatu di kamus yang berupa buku tebal membutuhkan waktu cukup lama serta kurang praktis untuk dibawa ke mana-mana [5]. Jumlah kamus bahasa daerah seperti

kamus bahasa Bangka yang tercetak masih sangat sedikit bahkan belum dipasarkan untuk masyarakat umum. Kamus tersebut biasanya hanya tersedia di balai bahasa dan perpustakaan wilayah [6]. Selain itu, terdapat juga kamus bahasa Bangka-Indonesia versi elektronik (*e-book*) yang bisa didapatkan secara gratis di *website* Kementerian Pendidikan Kebudayaan, Riset dan Teknologi Republik Indonesia (Kemendikbudristek) [7]. Fitur pencarian kosa kata di *e-book* namun masih memiliki kekurangan, yaitu kata kunci yang diketik harus mirip dan tidak boleh ada kesalahan pengetikan kosa kata [8]. Misalnya kosa kata “kededep” bisa dicari dengan mengetikkan “kede”, “edep”, atau “dede” namun pencarian akan gagal jika mengetikkan *typo* seperti “kehde”.

Berbagai kelemahan dari kamus fisik dan elektronik yang diperburuk kurangnya sosialisasi oleh dinas terkait kepada masyarakat atas keberadaan kamus-kamus tersebut menimbulkan masalah [9]. Salah satu masalah tersebut adalah cukup banyak jumlah masyarakat Bangka yang tinggal di perkotaan dan pendatang dari luar Bangka yang kurang memahami bahasa Bangka. Hal ini menyebabkan kesalahpahaman dan sulitnya komunikasi terjalin antara masyarakat Bangka, khususnya yang ada di pedesaan dan kurang fasih berbahasa Indonesia dengan masyarakat yang kurang memahami bahasa Bangka. Akibatnya aktivitas sehari-hari yang terjadi di lingkungan tersebut dalam berbagai bidang akan terganggu seperti pendidikan, pekerjaan, perdagangan, dan lain-lain.

Untuk mengatasi permasalahan di atas, akan dibuatkan sebuah aplikasi Android kamus terjemahan bahasa Indonesia-Bangka untuk perangkat *mobile*. Pada saat penelitian ini dilakukan, rata-rata berat dari perangkat *mobile* seperti *smartphone* yang beredar di pasaran adalah kurang dari 200 gram dan muat di saku celana sehingga lebih fleksibel, praktis, dan dapat dibawa kemana-mana jika dibandingkan dengan laptop [10]. Lebih dari 90% sistem operasi dari *smartphone* yang digunakan penduduk Indonesia, khususnya Bangka adalah Android [11].

Hal ini dikarenakan harga perangkatnya lebih terjangkau, lebih banyak dukungan aplikasi pihak ketiga, dan *open source* daripada pesaingnya, yaitu iPhone [10].

Aplikasi usulan dilengkapi fitur *spell checker* dengan algoritma *Levenshtein Distance* agar aplikasi dapat menerima dan mengoreksi input kosa kata dari pengguna yang salah ketik atau tidak sama persis dengan yang ada di basis data. Algoritma *Levenshtein Distance* merupakan pengukuran untuk mengkalkulasikan jumlah perbedaan antara dua kata yang berdasarkan jumlah operasi minimum untuk mengganti satu kata ke kata lainnya [8].

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 kamus bahasa Bima-Indonesia dimana ia dapat memeriksa kesalahan pengetikan dan kemudian merekomendasikan kata yang mungkin dimaksudkan pengguna [12]. Arnawa mengoptimasi pencarian kata kunci pada kamus aneka bahasa dengan algoritma *Levenshtein Distance* agar dapat mengatasi masalah berupa kesalahan pengejaan [13].

Selain *Levenshtein Distance*, terdapat algoritma lain yang dapat digunakan untuk pencarian kata yang tidak sama persis. Salah satunya adalah algoritma *Jaro Winkler* dengan *quadratic runtime complexity* ($O(n^2)$) yang sangat efektif pada pencarian *string* yang pendek. Algoritma ini menghitung panjang *string*, menemukan jumlah karakter yang sama antara dua *string*, dan menemukan jumlah transposisi [14]. Sebuah aplikasi *web* kamus istilah basis data dengan algoritma *Soft Cosine Similarity* mampu menampilkan sinonim dari kata yang diketik pengguna berdasarkan kedekatan atau kesamaan maknanya. Misalkan jika kata “pecahan” diinput, maka kata yang

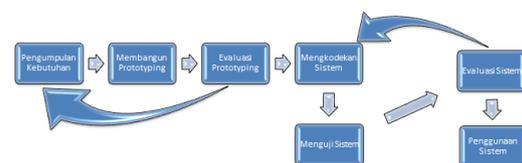
dianggap sinonim darinya adalah *double* dan *float* karena kedua kata tersebut mempunyai makna yang mendekati ataupun sama dengan kata “pecahan” [15].

Dengan menggunakan aplikasi usulan, diharapkan komunikasi antara masyarakat Bangka yang kurang fasih berbahasa Indonesia dengan masyarakat pendatang menjadi lebih mudah dan lancar. Diharapkan juga tingkat akurasi dari algoritma *Levenshtein Distance* dalam menyarankan kosa kata yang mungkin dimaksudkan oleh pengguna saat ada kesalahan pengetikan untuk berbagai skenario pengujian adalah di atas 60%.

2. METODE PENELITIAN

2.1. Model Prototype

Model *prototype* adalah salah satu metode siklus hidup sistem yang berdasarkan pada konsep model bekerja (*working model*). Pada *prototype* sebuah model akan dikembangkan menjadi sistem *final* lebih cepat dibandingkan model tradisional dengan biaya yang lebih rendah [16]. Model *prototype* digunakan dalam penelitian ini dengan harapan kepuasan pengguna akan meningkat dan aplikasi dapat diimplementasikan dengan baik. *Prototype* dari aplikasi mula-mula dipresentasikan ke pelanggan lalu pelanggan memberikan masukan untuk perbaikan *prototype* tersebut sebelum aplikasi jadi dibuat sehingga aplikasi tersebut nantinya sesuai dengan keinginan dan kebutuhan pelanggan. Adapun tahapan yang terdapat di model *prototype* ditunjukkan Gambar 1.



Gambar 1. Tahapan Model Prototype [16]

2.1.1. Pengumpulan Kebutuhan

Berdasarkan hasil pengumpulan data, secara garis besar dibutuhkan sebuah aplikasi kamus terjemahan bahasa

Indonesia-Bangka berbasis Android yang juga dapat merekomendasikan kosakata yang mungkin dimaksudkan pengguna jika terjadi kesalahan pengetikan. Aplikasi ini nantinya dapat membantu komunikasi antara masyarakat Bangka yang kurang fasih berbahasa Indonesia dengan masyarakat pendatang yang kurang memahami bahasa Bangka.

2.1.2. Membangun Prototyping

Pada tahap ini dibuat prototipe berupa desain halaman dan antarmuka dari aplikasi seperti halaman utama, halaman terjemahan, dan halaman manajemen kosakata. Selain itu akan dibuatkan juga *pseudocode* yang berupa langkah-langkah dan cara kerja dari algoritma *Levenshtein Distance* yang nantinya akan diterapkan sebagai *spell checker* dari aplikasi usulan.

2.1.3. Evaluasi Prototyping

Layout halaman dan *pseudocode* yang sudah didesain selanjutnya dievaluasi oleh beberapa pemangku kepentingan, seperti budayawan Bangka dan staf dari Kantor Bahasa Provinsi Kepulauan Bangka Belitung sebagai perwakilan penutur bahasa Bangka. Jika terdapat masukan dari mereka, tahap 1 dan 2 akan diulangi dimana prototipe diperbaiki berdasarkan masukan tersebut kemudian dievaluasi kembali. Jika tidak ada masukan, tahap mengkodekan sistem akan dilaksanakan.

2.1.4. Mengkode Sistem

Prototipe aplikasi yang sudah dievaluasi selanjutnya diberikan kode agar semua antarmuka dan fitur yang ada berjalan sesuai yang diharapkan. Selain itu, *pseudocode* dari algoritma *Levenshtein Distance* yang telah dibuat diterjemahkan ke dalam bahasa pemrograman Java dengan *Integrated Development Environment (IDE)* Android Studio. Adapun *database* yang digunakan untuk menyimpan sampel sebanyak 90 buah kosakata bahasa Indonesia-Bangka dan informasi *login* admin adalah Firebase.

2.1.5. Menguji Sistem

Sebelum dipresentasikan kembali ke pemangku kepentingan, aplikasi yang

telah dibuat akan diuji secara mandiri oleh pengembang dengan model pengujian *black box* agar tidak terdapat *bug* ataupun *error*. Selain itu akan dilakukan pengujian dengan 18 skenario berbeda untuk mengukur kinerja dari algoritma *Levenshtein Distance* dalam menyarankan kosakata yang mungkin dimaksudkan pengguna jika terdapat kesalahan pengetikan.

2.1.6. Evaluasi Sistem

Aplikasi kamus terjemahan bahasa Indonesia-Bangka yang sudah diuji kemudian dipresentasikan kembali ke pemangku kepentingan untuk dievaluasi. Jika ada masukan, maka tahap 4 dan 5 akan diulangi untuk memperbaiki kekurangan yang ada pada aplikasi berdasarkan masukan tersebut. Jika tidak ada masukan, maka tahap penggunaan sistem akan dilaksanakan.

2.1.7. Penggunaan Sistem

File Android Package File (APK) dari aplikasi jadi yang sudah dievaluasi oleh pemangku kepentingan kemudian disebarluaskan ke masyarakat umum, khususnya yang berdomisili di pulau Bangka. *File* APK tersebut kemudian diunggah di penyedia layanan penyimpanan *file* seperti Google Drive dan tautan *Uniform Resource Locator (URL)* untuk mengunduhnya dibagikan di *website* pihak terkait, sosial media, dan di grup WhatsApp. Dikarenakan aplikasi yang dibuat menggunakan Firebase sebagai *Database Management System (DBMS)*, maka penyewaan layanan *hosting* tidak dibutuhkan.

2.2. Teknik Pengumpulan Data

Tahap teknik pengumpulan data berkaitan dengan penentuan kebutuhan pengguna dan pengembang. Pada penelitian ini akan digunakan beberapa metode untuk mendapatkan data dan informasi yang dibutuhkan, yaitu wawancara, observasi, dan studi pustaka.

2.2.1. Wawancara

Wawancara dilakukan oleh peneliti dengan salah satu budayawan Bangka agar dapat diidentifikasi informasi penting

seperti masalah terkait komunikasi yang ada di lingkungan pulau Bangka dan fitur utama dari aplikasi kamus terjemahan yang akan dibuat sebagai solusi dari masalah-masalah tersebut. Selain itu akan ditanyakan juga tentang proses penerjemahan bahasa Indonesia ke Bangka atau sebaliknya oleh budayawan secara konvensional.

2.2.2. Observasi

Observasi dilakukan oleh peneliti dengan mengamati secara langsung komunikasi yang terjadi dalam kehidupan sehari-hari antara masyarakat Bangka yang kurang fasih berbahasa Indonesia dengan masyarakat dari luar Bangka yang kurang memahami bahasa Bangka misalkan dalam proses jual beli barang di pasar tradisional. Agar komunikasi dapat terlaksana dengan baik, mereka didampingi oleh kenalan yang berperan sebagai penerjemah sekaligus memahami bahasa Bangka dan bahasa Indonesia.

2.2.3. Studi Pustaka

Studi pustaka dilakukan oleh peneliti dengan mempelajari berbagai karya ilmiah yang relevan dengan aplikasi kamus terjemahan. Selain itu dipelajari juga sebuah *e-book* kamus bahasa Melayu Bangka-Indonesia dimana 90 dari keseluruhan kosakata yang ada pada *e-book* tersebut nantinya digunakan sebagai basis pengetahuan dari aplikasi kamus.

2.3. Tools Pengembangan Perangkat Lunak

Pada metode berorientasi objek, karakteristik dari perangkat lunak dimodelkan dengan menggunakan *tools* pengembangan perangkat lunak, dimana *tools* yang digunakan di penelitian ini adalah *Unified Modeling Language* (UML). Pada UML terdapat diagram-diagram pendukung yang terbagi menjadi 2 bagian, yaitu bagian analisis dan bagian rancangan. Pada penelitian ini, di bagian analisis ada *use case diagram* dan *activity diagram*, sedangkan di bagian rancangan ada *sequence diagram* dan *class diagram*.

1) *Use case diagram* digunakan untuk menggambarkan kebutuhan fungsional

dari setiap *actor* pada aplikasi kamus terjemahan bahasa Indonesia-Bangka.

- 2) *Activity diagram* digunakan untuk menggambarkan dan menganalisa proses penerjemahan kosakata bahasa Indonesia ke Bangka atau sebaliknya yang berjalan dan diusulkan sekaligus menjelaskan aliran fungsionalitas dari manajemen kosakata di aplikasi usulan.
- 3) *Sequence diagram* digunakan untuk menggambarkan rangkaian atau urutan eksekusi dari setiap proses / fitur yang ada di aplikasi kamus yang diusulkan.
- 4) *Class diagram* digunakan untuk menggambarkan struktur tabel yang ada di basis data Firebase yang terdapat pada aplikasi usulan.

2.4. Algoritma Levenshtein Distance

Pada algoritma *Levenshtein Distance* terdapat matriks yang digunakan untuk menghitung jumlah perbedaan antara dua *string*. Jarak atau *distance* antara kedua *string* ditentukan oleh jumlah minimum operasi perubahan untuk mengubah dari *string* satu ke *string* lainnya yang dihitung dengan suatu tabel dimana nilai akhir dari jarak kedua *string* berada di pojok kanan bawah tabel.

String “bentuk” dan “bentik” berjarak 1 sebab dibutuhkan 1 operasi dalam mengubah “bentik” menjadi “bentuk”. Ada 3 jenis operasi utama yang dapat dilakukan oleh algoritma *Levenshtein Distance* yaitu pengubahan / penukaran, penambahan, dan penghapusan karakter.

Proses perhitungan jarak dari algoritma *Levenshtein Distance* (LD) dimulai dari pojok kiri atas sebuah *array* 2 dimensi / matriks yang sudah berisikan karakter *string* awal dan *string* sasaran serta diberikan nilai *cost* / biaya. Nilai di matriks adalah jumlah operasi penghapusan, penambahan, dan penukaran yang diperlukan dalam mengubah *string* awal ke *string* sasaran [17]. Operasi penghapusan, penambahan, penukaran, dan tanpa perubahan karakter berturut-turut dapat dilihat pada Persamaan (1) hingga (4) [18].

$$D(s, t) = \min D(s - 1, t) + 1 \quad (1)$$

$$D(s, t) = \min D(s, t - 1) + 1 \quad (2)$$

$$D(s, t) = \min D(s - 1, t - 1) + 1, sj \neq ti \quad (3)$$

$$D(s, t) = \min D(s - 1, t - 1), sj = ti \quad (4)$$

Keterangan :

- D = jarak LD
- s = string awal
- t = string sasaran
- s(j) = karakter string awal ke-j
- t(i) = karakter string sasaran ke-i

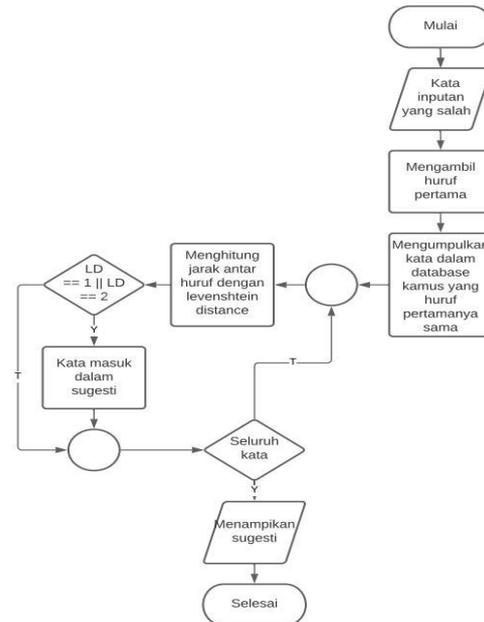
Saat terdapat kesalahan input kosakata oleh pengguna, algoritma LD akan berjalan dimana karakter pertama inputan pada kotak pencarian akan diambil. Setelah itu kosakata di basis data dengan karakter pertama yang sama dengan inputan dikumpulkan untuk perhitungan jarak mereka. Setelah itu LD1 (string awal atau kosakata yang diinput pengguna) dan LD2 (string sasaran atau kosakata di basis data) dibandingkan. Jika sama, kosakata tersebut disertakan sebagai saran yang akan ditampilkan. Jika berbeda, akan dilakukan pengecekan apakah yang tampil seluruh kosakata atau tidak. Jika tidak, langkah awal yaitu pencarian jarak dengan algoritma LD akan diulangi. Jika yang tampil adalah keseluruhan kosakata, saran atau prediksi kosakata akan ditampilkan dan proses algoritma LD berhenti. Untuk lebih jelasnya, flowchart cara kerja algoritma LD pada aplikasi usulan adalah seperti yang ditunjukkan Gambar 2.

Setelah nilai LD dari 2 kosakata yang dibandingkan diketahui, nilai kemiripan dari perbandingan kosakata tersebut kemudian dihitung menggunakan Persamaan (5).

$$Sim = 1 - \frac{d[m, n]}{\max[S, T]} \quad (5)$$

Keterangan :

- Sim = similarity / kemiripan
- d[m,n] = jarak LD
- max[S,T] = jumlah karakter maksimal dari 2 string



Gambar 2. Flowchart dari Algoritma LD

3. HASIL DAN PEMBAHASAN

3.1. Analisis Masalah

Masalah-masalah yang terjadi pada latar belakang penelitian dapat diidentifikasi dengan melaksanakan analisa masalah. Masih cukup banyak masyarakat Bangka yang tinggal di perkotaan dan masyarakat pendatang yang kurang memahami bahasa Bangka. Hal ini membuat komunikasi antara mereka dengan masyarakat Bangka yang tinggal jauh dari perkotaan dan kurang memahami bahasa Indonesia menjadi sulit tanpa kehadiran penerjemah. Selain itu, kamus bahasa Bangka cetak yang beredar di masyarakat Bangka jumlahnya masih sangat sedikit. Sekalipun ada, kamus cetak tidak praktis untuk dibawa kemana-mana karena ukurannya cukup besar dan cukup berat.

Sebenarnya sudah ada kamus bahasa Melayu Bangka-Indonesia dalam bentuk e-book yang dirilis secara online oleh instansi terkait, namun e-book ini kurang dikenal oleh masyarakat akibat minimnya publikasi. Selain itu, pencarian kosakata pada e-book juga memiliki keterbatasan seperti kosakata yang dicari tidak boleh memiliki kesalahan dalam pengetikan / typo agar kosakata tersebut dapat ditemukan.

3.2. Analisis Kebutuhan

Analisis kebutuhan dilaksanakan agar kebutuhan yang diperlukan dalam proses pembuatan aplikasi kamus terjemahan bahasa Indonesia-Bangka dapat dikenali. Kebutuhan fungsional memuat proses-proses yang dapat dilakukan oleh setiap pengguna dengan peran berbeda pada aplikasi usulan. Gambar 3 adalah *use case diagram* yang menggambarkan kebutuhan fungsional dari aplikasi usulan dimana pengguna masyarakat dapat menterjemahkan kosakata bahasa Bangka ke Indonesia atau sebaliknya serta admin dapat melakukan manajemen data kosakata.



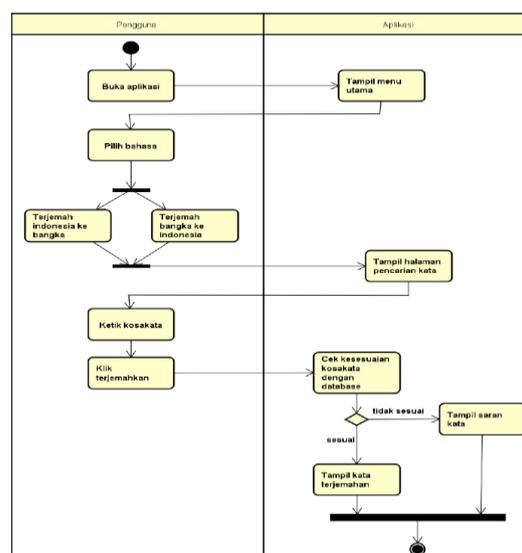
Gambar 3. Use Case Diagram Aplikasi Kamus Terjemahan yang Diusulkan

3.3. Identifikasi Sistem Usulan

Sebuah aplikasi kamus terjemahan bahasa Indonesia-Bangka nantinya akan diusulkan untuk memperbaiki kekurangan dari proses penerjemahan yang berjalan. Perangkat sasaran dari aplikasi usulan adalah *smartphone* berbasis Android dimana layaknya aplikasi Google Translate, aplikasi usulan memungkinkan pengguna memasukkan kosakata dalam bahasa Bangka dan kemudian aplikasi akan menterjemahkan kosakata tersebut ke bahasa Indonesia atau sebaliknya. Terdapat juga fitur *spell checker* di aplikasi agar pengguna tidak harus memasukkan kosakata yang sama persis dengan kosakata yang terdapat di *database*.

Proses penerjemahan bahasa Bangka ke bahasa Indonesia atau sebaliknya dengan menggunakan aplikasi

usulan dimulai dari pengguna menjalankan aplikasi usulan yang sudah terpasang di *smartphone*-nya. Aplikasi kemudian menampilkan halaman *menu* utama dan pengguna dapat memilih untuk menterjemahkan kosakata bahasa Bangka ke Indonesia atau sebaliknya. Pengguna selanjutnya dialihkan ke halaman pencarian kosakata untuk memasukkan kosakata yang ingin diterjemahkan. Aplikasi lalu mencocokkan kosakata yang dimasukkan dengan kosakata yang ada di *database*. Jika cocok, aplikasi menampilkan terjemahan dari kosakata tersebut. Jika tidak cocok, aplikasi merekomendasikan kosakata yang paling mungkin dimaksudkan oleh pengguna. *Activity diagram* aplikasi usulan dalam menterjemahkan kosakata dapat dilihat pada Gambar 5.



Gambar 5. Activity Diagram Penerjemahan Kosakata Aplikasi Usulan

3.4. Rancangan Sistem

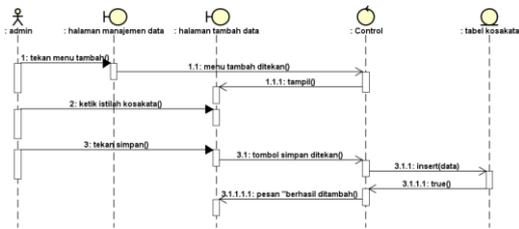
3.4.1. Sequence Diagram

Sequence diagram menjelaskan aliran fungsionalitas pada *use case* sekaligus menggambarkan urutan interaksi objek yang disusun dalam urutan waktu. Beberapa *sequence diagram* yang terdapat pada penelitian ini adalah sebagai berikut:

1) *Sequence diagram* tambah kosakata

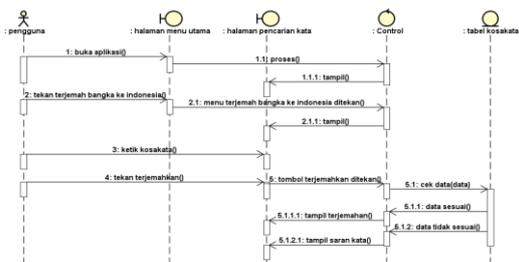
Proses yang terjadi ketika admin menambahkan kosakata bahasa Bangka-Indonesia ke basis data dengan

aplikasi usulan dapat dilihat pada Gambar 6.



Gambar 6. Sequence Diagram Tambah Kosakata

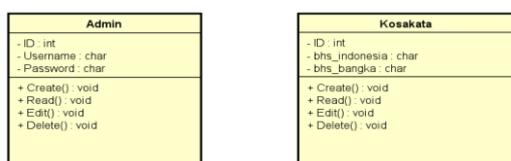
2) Sequence diagram terjemah bahasa Proses yang terjadi ketika masyarakat menerjemahkan kosakata bahasa Bangka ke Indonesia dengan aplikasi usulan dapat dilihat pada Gambar 7. Adapun proses penerjemahan bahasa Indonesia ke Bangka hanya berbeda pada menu yang digunakan.



Gambar 7. Sequence Diagram Terjemah Kosakata Bahasa Bangka ke Indonesia

3.4.2. Perancangan Basis Data

Pemetaan model konseptual ke bentuk model basis data yang akan digunakan aplikasi terjadi pada tahap perancangan basis data. Basis data dari aplikasi usulan dirancang dengan menggunakan class diagram seperti yang ditunjukkan Gambar 8. Pada class diagram, relasi antar class yang ada pada aplikasi usulan dapat dijelaskan. Adapun pada basis data aplikasi usulan hanya terdapat 2 class, yaitu admin dan kosakata yang tidak berelasi satu sama lain.



Gambar 8. Class Diagram Aplikasi Usulan

3.5. Implementasi Sistem

Tahap implementasi sistem adalah tahap penerapan sistem yang sudah dibuat pada tahap perancangan sistem agar nanti aplikasi dapat dioperasikan.

3.5.1. Tampilan Layar

Tampilan layar merupakan tampilan halaman dari aplikasi yang diusulkan. Beberapa halaman serta fungsinya yang ada di aplikasi usulan adalah sebagai berikut.

1) Tampilan layar menu utama

Tampilan dari halaman menu utama dapat dilihat pada Gambar 9 dimana pada halaman ini terdapat 4 menu yang dapat dipilih, yaitu terjemah Indonesia ke Bangka, terjemah Bangka ke Indonesia, informasi aplikasi, dan manajemen data.



Gambar 9. Tampilan Layar Menu Utama

2) Tampilan layar menu terjemah Indonesia ke Bangka

Tampilan dari halaman terjemah Indonesia ke Bangka atau sebaliknya ditunjukkan Gambar 10. Jika kosakata yang dimasukkan sama persis dengan yang ada di basis data, terjemahannya akan ditampilkan. Jika tidak, sebuah kosakata yang menyerupai akan ditampilkan.

3) Tampilan layar login admin

Tampilan dari halaman login admin ditunjukkan Gambar 11 dimana seorang admin harus memverifikasi dirinya terlebih dahulu sebelum dapat mengelola kosakata yang terdapat di basis data aplikasi yang diusulkan.



Gambar 10. Tampilan Layar Terjemahan (kiri) Sama (kanan) Tidak Sama

3.5.2. Perhitungan Jarak Levenshtein

Algoritma LD diterapkan di menu pencarian kosakata dari aplikasi kamus terjemahan usulan sehingga aplikasi dapat menoleransi kesalahan pengetikan dan menyarankan kosakata dari basis data yang paling mirip dengan kosakata yang diinput.



Gambar 12. Tampilan Layar Manajemen Kosakata

Nilai jarak dimulai dari 0 dan dihitung dengan membandingkan karakter di posisi indeks yang sama. Nilai jarak tetap jika kedua karakter sama. Operasi yang sesuai akan dilakukan jika kedua karakter berbeda kemudian nilai jarak bertambah 1. Proses perhitungan jarak LD terdiri dari perhitungan jarak *string* awal dengan *string* sasaran dilanjutkan perhitungan nilai *similarity*.

- 1) Menghitung jarak *string* awal dengan *string* sasaran
Misalkan *string* awal (s) adalah “kecaklor” dan *string* sasaran (t) adalah

“kecalok”, mula-mula perhitungan matriks dimulai dari inialisasi urutan karakter pada masing-masing *string*. Untuk mengubah “kecaklor” menjadi “kecalok” nantinya membutuhkan 2 operasi, yaitu menghapus “k” kedua di “kecaklor” lalu mengubah “r” di “kecaklor” menjadi “k”.

Berdasarkan Gambar 14, “kecaklor” memiliki 8 karakter dan “kecalok” memiliki 7 karakter. Karakter ke-1 dari keduanya bernilai sama sehingga nilai matriks yang diberikan sesuai Persamaan (4), yaitu :

$$D(1,1) = D(1 - 1, 1 - 1) = 0$$

	t	K	E	C	A	L	O	K
s	D	1	2	3	4	5	6	7
K	1	0						
E	2							
C	3							
A	4							
K	5							
L	6							
O	7							
R	8							

Gambar 14. Nilai Matriks setelah Jarak D(1,1) Dihitung

Jarak karakter ke-1 (s) dengan karakter ke-2 (t) kemudian dihitung dan diketahui bahwa operasi penambahan karakter “e” pada (s) diperlukan. Nilai matriks yang diberikan sesuai Persamaan (2), yaitu :

$$D(1,2) = D(1, 2 - 1) + 1 = 0 + 1 = 1$$

Jarak karakter ke-1 (s) dengan karakter ke-3 (t) kemudian dihitung dan diperlukan operasi penambahan karakter “e” dan “c” pada (s). Nilai matriks yang diberikan sesuai Persamaan (2), yaitu :

$$D(1,3) = D(1, 3 - 1) + 1 = 1 + 1 = 2$$

Perhitungan jarak karakter ke-1 (s) dengan karakter ke-4 hingga ke-7 (t) selanjutnya dilakukan untuk menambahkan karakter “e”, “c”, “a”, “l”, “o”, dan “k” pada (s) dimana hasilnya dapat dilihat pada Gambar 15

$$D(1,4) = D(1,4 - 1) + 1 = 2 + 1 = 3$$

$$D(1,5) = D(1,5 - 1) + 1 = 3 + 1 = 4$$

$$D(1,6) = D(1,6 - 1) + 1 = 4 + 1 = 5$$

$$D(1,7) = D(1,7 - 1) + 1 = 5 + 1 = 6$$

	t	K	E	C	A	L	O	K
s	D	1	2	3	4	5	6	7
K	1	0	1	2	3	4	5	6
E	2							
C	3							
A	4							
K	5							
L	6							
O	7							
R	8							

Gambar 15. Nilai Matriks setelah Jarak D(1,7) Dihitung

Jarak LD terus dilakukan hingga semua nilai pada matriks terisi. Jarak LD berada di pojok kanan bawah matriks, yaitu sebesar 2 pada D(8,7) seperti yang ditunjukkan Gambar 16.

	t	K	E	C	A	L	O	K
s	D	1	2	3	4	5	6	7
K	1	0	1	2	3	4	5	6
E	2	1	0	1	2	3	4	5
C	3	2	1	0	1	2	3	4
A	4	3	2	1	0	1	2	3
K	5	4	3	2	1	1	2	3
L	6	5	4	3	2	1	2	3
O	7	6	5	4	3	2	1	2
R	8	7	6	5	4	3	2	2

Gambar 16. Nilai Matriks setelah Jarak D(8,7) Dihitung

Selanjutnya perhitungan dilakukan pada kata kunci kedua hingga terakhir sebagai *string* sasaran, sehingga diketahui masing-masing jarak LD-nya seperti pada Tabel 1.

Kosakata dengan jarak LD terkecil dipilih sebagai kosakata yang disarankan, yaitu “kecalok” untuk *string* awal “kecaklor”.

Tabel 1. Jarak LD *String* Awal dan Sasaran

<i>String</i> Awal	<i>String</i> Sasaran	Jarak LD
Kecaklor	Kecalok	2
	Kecalep	3
	Kededep	6
	Keceker	3

2) Menghitung nilai *similarity*
Similarity / kemiripan dari 2 *string* dihitung menggunakan Persamaan (5) dengan menghitung jarak *levenshtein* dan panjang terbesar dari kedua *string*. Berikut nilai *similarity* antara “kecaklor” (s) dengan “kecalok” (t) dimana nilai jaraknya (d[m,n]) adalah 2 dan panjang “kecaklor” serta “kecalok” masing-masing adalah 8 (S) dan 7 (T):

$$Sim = \left(1 - \frac{2}{\max[8,7]}\right) \times 100\% = 75\%$$

Dengan menggunakan Persamaan (5), nilai *similarity* dari 4 *string* sasaran (t) untuk panjang *string* awal adalah 8 (S) dirangkum ke dalam Tabel 2.

Tabel 2. Hasil Perhitungan *Similarity*

t	T	d[m,n]	max[S,T]	Sim
Kecalok	7	2	8	75%
Kecalep	7	3	8	62,5%
Kededep	7	6	8	25%
Keceker	7	3	8	62,5

Berdasarkan Tabel 2, *string* dengan nilai *similarity* tertinggi diperoleh dari nilai jarak yang terendah. Pada contoh ini, kosakata yang memiliki nilai jarak terendah dan nilai similaritas tertinggi untuk disarankan adalah “kecalok”.

3.5.3. Pengujian Kinerja Algoritma LD

Kinerja dari algoritma LD yang diterapkan sebagai *spell checker* pada aplikasi kamus terjemahan bahasa Bangka-Indonesia diukur berdasarkan jumlah kosakata yang disarankan dengan tepat dari 18 kali pengujian dengan skenario berbeda. Saran dinyatakan tepat jika sesuai (S) dengan kosakata yang dimaksudkan oleh pengguna dan tidak tepat jika tidak sesuai (TS). Saat penelitian ini dilakukan, ada 90 pasang kosakata bahasa Bangka-Indonesia yang ditambahkan di basis data. Hasil pengujian kinerja algoritma LD dirangkum ke dalam Tabel 3 dan Tabel 4 dimana ada 3 saran yang tidak sesuai dengan kosakata yang dimaksud oleh pengguna. Hal ini dikarenakan ada beberapa *string* sasaran yang memiliki jarak LD dan nilai *similarity* yang sama. Adapun tingkat akurasi algoritma LD adalah sebagai berikut :

$$\text{Akurasi} = \frac{18 - 3}{18} \times 100\% = 83,33\%$$

Tabel 3. Hasil Pengujian Saran Bahasa Bangka

No	Skenario	Input	Maksud	Saran	Ket
1	Pergantian 1 huruf	Kecek	Kacek	Kacek	S
2	Penambahan 1 huruf	Galeh	Galek	Gale	TS
3	Penyisipan 1 huruf	Kadhep	Kadep	Kadep	S
4	Penggantian 1 huruf	Ikal	Ikan	Ikak	TS
5	Penghapusan 1 huruf	Jwet	Juwet	Juwet	S
6	Penukaran 1 huruf	Emphe	Empes	Empes	S
7	Penukaran 2 huruf	Palebuk	Pelabuk	Pelabuk	S
8	Penggantian 2 huruf	Sekanak	Sekenek	Sekenek	S
9	Kesalahan > 2 huruf	Tiapou	Taipau	Taipau	S

Tabel 4. Hasil Pengujian Saran Bahasa Indonesia

No	Skenario	Input	Maksud	Saran	Ket
1	Pergantian 1 huruf	Kakik	Kakak	Kakek	TS
2	Penambahan 1 huruf	Manjah	Manja	Manja	S
3	Penyisipan 1 huruf	Hanantu	Hantu	Hantu	S
4	Penggantian 1 huruf	Ibi	Ibu	Ibu	S
5	Penghapusan 1 huruf	Dpan	Depan	Depan	S
6	Penukaran 1 huruf	Bantinj	Banting	Banting	S
7	Penukaran 2 huruf	Soreng	Sering	Sering	S
8	Penggantian 2 huruf	Kelihetin	Kelihatan	Kelihatan	S
9	Kesalahan > 2 huruf	Kaliem	Kalian	Kalian	S

4. KESIMPULAN

Berdasarkan analisis, perancangan, implementasi, dan pengujian yang sudah dilaksanakan dalam implementasi algoritma *Levenshtein Distance* di aplikasi kamus terjemahan Bahasa Indonesia – Bahasa Bangka, berikut kesimpulan-kesimpulan yang didapat :

- 1) Aplikasi kamus yang diusulkan dapat digunakan untuk menerjemahkan kosakata bahasa Bangka ke Indonesia atau sebaliknya dengan baik sehingga dapat membantu komunikasi antara masyarakat Bangka yang kurang fasih berbahasa Indonesia dengan masyarakat pendatang yang kurang menguasai bahasa Bangka.

- 2) Algoritma *Levenshtein Distance* yang diterapkan pada fitur *spell checker* dari aplikasi kamus usulan memungkinkan aplikasi menerima dan mengoreksi input kosa kata yang salah ketik atau tidak sama persis dengan yang ada di basis data.
- 3) Tingkat akurasi dari algoritma *Levenshtein Distance* dalam menyarankan kosa kata yang mungkin dimaksudkan oleh pengguna saat ada kesalahan pengetikan adalah sudah baik, yaitu sebesar 83,33%.

5. SARAN

Adapun saran-saran yang dapat menjadi masukan untuk penelitian berikutnya yang terkait dengan penelitian ini adalah sebagai berikut :

- 1) Penggunaan algoritma pencocokan *string* lain yang lebih baik atau memadukannya dengan *Levenshtein Distance* untuk meningkatkan tingkat akurasi algoritma.
- 2) Aplikasi sebaiknya merekomendasikan semua kosakata dari basis data dengan skor pengukuran melebihi ambang batas yang ditentukan jika terdapat *typo* oleh pengguna.
- 3) Aplikasi sebaiknya dapat menerima input berupa kalimat dan juga menampilkan makna dari kosakata yang diterjemahkan agar tidak terjadi miskomunikasi
- 4) Penambahan algoritma *stemming* teks agar aplikasi dapat menghilangkan imbuhan, awalan, dan akhiran dari kosakata yang diinput pengguna menjadi bentuk kata dasar sebelum dibandingkan dengan kosakata yang ada di basis data.

DAFTAR PUSTAKA

- [1] Mailani, O., Nuraeni, I., Syakila, S. A., dan Lazuardi, J., 2022, Bahasa Sebagai Alat Komunikasi dalam Kehidupan Manusia, *Kampret J.*, vol. 1, no. 1, hal 1–10.
- [2] Rahman, M. F., Najah, S., Furtuna, N. D., dan Anti, 2020, *Bhinneka Tunggal*

- Ika sebagai Benteng terhadap Risiko Keberagaman Bangsa Indonesia, *Al-Din J. Dakwah dan Sos. Keagamaan*, vol. 6, no. 2, hal 1–16.
- [3] Hadi, N. R. P., dan Kusumaningrum, S., 2020, Korespondensi Fonemis Bahasa Ogan dan Bahasa Bangka, *J. Cult. (Culture, Lang. Lit. Rev.)*, vol. 7, no. 2, hal 191–203.
- [4] Saputra, P. P., dan Afifulloh, M., 2020, Pemetaan Penggunaan Bahasa Melayu Bangka, *KREDO J. Ilm. Bhs. dan Sastra*, vol. 3, no. 2, hal 307–321.
- [5] Danur, H., Tute, K. J., dan Bhae, B. Y., 2022, Aplikasi Kamus Bahasa Daerah Manggarai Berbasis Android, *SATESI J. Sains Teknol. dan Sist. Inf.*, vol. 2, no. 2, hal 140–146.
- [6] Rahmat, N. F., Ibrahim, F., dan Rizaldy, A., 2023, Rancang Bangun Aplikasi Kamus Bahasa Daerah Menggunakan Flutter (Bahasa Daerah Bugis Kabupaten Pinrang), *SHIFT J. Software, Hardware, Inf. Technol.*, vol. 3, no. 2, hal. 1–10.
- [7] Kantor Bahasa Kepulauan Bangka Belitung, 2018, *Kamus Bahasa Melayu Bangka-Indonesia*, Ed. 1, Kantor Bahasa Kepulauan Bangka Belitung, Pangkalpinang.
- [8] Lihawa, M., Hartanto, A. D., Norhikmah, Prabowo, D., Fajri, I. N., dan Widayani, W., 2023, Implementasi Algoritma Levenshtein Distance dan Metode Regular Search Expression dalam Mendeteksi Salah Ketik pada JavaScript, *Sist. J. Sist. Inf.*, vol. 12, no. 2, hal 452–464.
- [9] Zalwia, Moita, S., dan Upe, A., 2018, Modernisasi dan Diskontinuitas Bahasa Daerah (Studi Kasus Penggunaan Bahasa Daerah GU di Kelurahan Lakudo Kecamatan Lakudo Kabupaten Buton Tengah), *JNS J. Neo Soc.*, vol. 3, no. 2, hal 494–502.
- [10] Adityawan, O., Rahadi, P. F., Pratama, S. D., dan Fachrurozi, R., 2023, Perancangan Media Interaktif Cerita Nabi-Nabi Ulul Azmi untuk Anak-Anak dalam Bentuk Aplikasi Berbasis Android sebagai Media Alternatif, *Wacadesain*, vol. 4, no. 1, hal. 1–9.
- [11] Ramadhan, F. Z., dan Adhitama, R., 2023, Perancangan Aplikasi Kantin Elektronik Berbasis Android Menggunakan Metode Scrum, *LEDGER J. Inform. Inf. Technol.*, vol. 2, no. 1, pp. 58–73.
- [12] Sari, Y. P., Pradnyana, G. A., dan Wirawa, I M. A., 2019, 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, hal 86–95.
- [13] Arnawa, I. B. K. S., 2018, Optimasi Pencarian Kata pada Kamus Aneka Bahasa Menggunakan Algoritma Levenshtein Distance, *J. Sist. dan Inform.*, vol. 12, no. 2, hal 149–157.
- [14] Sahfitri, V., dan Zarizal, I. B., 2022, Approximate String Matching untuk Pencarian Kata dalam Kamus Bahasa Indonesia menggunakan Algoritma Jaro Winkler, *J. Ilm. Matrik*, vol. 24, no. 3, hal 248–259.
- [15] Nuristiqomah, R. P., dan Anistyasari, Y., 2021, Pengembangan Kamus Istilah Basis Data Berbasis Website Menggunakan Algoritma Cosine Similarity untuk Meningkatkan Hasil Belajar Siswa, *J. IT-EDU*, vol. 05, no. 02, hal 621–630.
- [16] Ardiansyah, F., dan Munawaroh, 2023, Pengembangan Sistem Informasi Keanggotaan Online Berbasis Web Menggunakan Framework Laravel dengan Metode Prototype pada Asosiasi Inkindo, *JORAPI J. Res. Publ. Innov.*, vol. 1, no. 2, hal 266–271.
- [17] Zulkarnain, A., Poerbaningtyas, E., dan Risqoni, D. O., 2021, Sistem Pengamanan Ponsel Melalui Penyaringan Kata dengan Metode Levenshtein Distance, *Prosiding Seminar Nasional Sistem Informasi dan Teknologi (SISFOTEK) ke-5*, Padang, September 25.
- [18] Abdurahman dan Ismawan, F., 2022, Model Machine Learning Klasifikasi Data Sekolah TK Berdasarkan Status dan Kabupaten/Kota Administrasi Provinsi DKI Jakarta, *Fakt. Exacta*, vol. 15, no. 2, hal 127–138.