

LAPORAN PENELITIAN



IMPLEMENTASI ALGORITMA *LEVENSHTEIN* DISTANCE UNTUK FITUR PENCARIAN KATA PADA PENGEMBANGAN KAMUS BAHASA ACEH BERBASIS WEB

Peneliti:

Rizal Ilham

NIM.180705027

Jenis Penelitian	Penelitian Inter Disipliner
Bidang Ilmu Kajian	Machine Learning
Dosen Peneliti	Bustami, M.Sc

UNIVERSITAS ISLAM NEGERI AR-RANIRY BANDA ACEH
FAKULTAS SAINS DAN TEKNOLOGI
PRODI TEKNOLOGI INFORMASI
MARET 2023

**LEMBARAN IDENTITAS DAN PENGESAHAN LAPORAN PENELITIAN PUSAT PENELITIAN DAN
PENERBITAN LP2M UIN AR-RANIRY TAHUN 2023**

1. a. Judul : Implementasi Algoritma *Levenshtein* Distance Untuk Fitur Pencarian Kata Pada Pengembangan Kamus Bahasa Aceh Berbasis Web
- b. Jenis Penelitian : Penelitian Inter Disipliner
- c. No. Registrasi : -
- d. Bidang Ilmu yang diteliti : Machine Learning
2. Peneliti
 - a. Nama Lengkap : Rizal Ilham
 - b. Jenis Kelamin : Laki Laki
 - c. NIM : 180705027
 - d. Fakultas/Prodi : Sains dan Teknologi/Teknologi Informasi
 - e. Anggota Peneliti 1
 - Nama Lengkap : Bustami, M.Sc.
 - Jenis Kelamin : Laki Laki
 - Fakultas/Prodi : Sains dan Teknologi/ Teknologi Informasi
 - f. Anggota Peneliti 2 *(Jika Ada)*
 - Nama Lengkap : Ima Dwitawati, M.B.A.
 - Jenis Kelamin : Perempuan
 - Fakultas/Prodi : Sains dan Teknologi/ Teknologi Informasi
3. Lokasi Kegiatan : Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh
4. Jangka Waktu Pelaksanaan : 6 (Enam) Bulan
5. Tahun Pelaksanaan : 2023
6. Jumlah Anggaran Biaya : -
7. Sumber Dana : Mandiri
8. Output dan Outcome : -

Mengetahui,
Dosen Pembimbing I



Bustami, M.Sc
NIDN. 2008048601

Banda Aceh, 01 Maret 2023

Pelaksana,



Rizal Ilham
NIM. 180705027

Menyetujui:

Ketua Prodi. Teknologi Informasi



Ima Dwitawati

NIP. 198210132014032002

ABSTRAK

Nama : Rizal Ilham
NIM : 180705027
Program Studi : Teknologi Informasi
Judul : Implementasi Algoritma Levenshtein Distance Untuk
Fitur Pencarian Kata Pada Pengembangan Kamus Bahasa
Aceh Berbasis Web
Tanggal Sidang :
Jumlah Halaman : 52 Halaman
Pembimbing I : Bustami, M. Sc
Pembimbing II : Ima Dwitawati, M.B.A
Kata Kunci : Aplikasi Kamus, Kamus Digital, Bahasa Aceh,
Autocomplete , Algoritma *Levenshtein Distance*

Bahasa Aceh merupakan bagian dari kebudayaan suatu masyarakat Aceh dan perlu dijaga dan dilestarikan keberadaannya. Salah satu cara untuk menjaga kelestarian bahasa Aceh adalah dengan membuat kamus digital. Namun, sebagian besar kamus digital bahasa Aceh yang sudah ada masih memiliki kekurangan pada fitur pencarian dan *database* kosakata yang terbatas. Penelitian ini bertujuan untuk memperbaiki aplikasi kamus digital bahasa Aceh yang sudah ada dengan fitur *Autocomplete* yang lebih baik dan berbasis website menggunakan Algoritma *Levenshtein Distance* untuk meningkatkan kualitas dan kemudahan akses pengguna dalam mencari dan memahami kosakata bahasa Aceh. Penelitian ini menggunakan metode pengumpulan data dengan melakukan *scanning* pada kamus fisik sebagai sumber data yang valid untuk kamus digital bahasa Aceh. Kemudian, dilakukan pengembangan codebase algoritma *Levenshtein Distance* dan implementasi Algoritma pada fitur pencarian kamus digital bahasa Aceh. Penelitian kamus digital bahasa Aceh dengan fitur *Autocomplete* dan Algoritma *Levenshtein Distance* berhasil dirancang dan diimplementasikan dengan baik. Pengguna dapat mencari kosakata bahasa Aceh dengan mudah dan cepat, serta

mendapatkan rekomendasi kata ketika salah mengetik atau lupa mengetik karakter kata yang ingin dicari. Selain itu, *database* kosakata dalam aplikasi ini lebih lengkap dan diharapkan dapat membantu pengguna dalam memahami bahasa Aceh secara lebih baik. Aplikasi kamus digital bahasa Aceh dengan fitur *Autocomplete* dan Algoritma *Levenshtein Distance* yang dirancang dalam penelitian ini berhasil meningkatkan kualitas dan kemudahan akses pengguna dalam mencari dan memahami kosakata bahasa Aceh pada aplikasi web yang sudah ada. Diharapkan aplikasi ini dapat membantu dalam menjaga dan melestarikan keberadaan bahasa Aceh sebagai bagian dari kebudayaan masyarakat Aceh. Penelitian ini menggunakan 100 data sebagai pengujian sistem, sehingga persentasi untuk pengujian jumlah data yang berbeda dapat berubah.

Kata kunci: Aplikasi Kamus, Kamus Digital, Bahasa Aceh, *Autocomplete*, Algoritma *Levenshtein Distance*.



ABSTRACT

Name : Rizal Ilham
Student ID : 180705027
Department : Information Technology
Title : Implementation of the Levenshtein Distance Algorithm for the Word Search Feature in the Development of a Web-Based Aceh Language Dictionary
Date :
Thesis Pages : 52 Pages
Supervisor I : Bustami, M. Sc
Supervisor II : Ima Dwitawati, M.B.A
Keywords : Dictionary Application, Digital Dictionary, Aceh Language, Autocomplete, Levenshtein Distance Algorithm.

The Acehnese language is part of the culture of an Acehnese people and needs to be maintained and preserved. One way to preserve the Acehnese language is to create a digital dictionary. However, most of the existing Aceh language digital dictionaries still lack search features and limited vocabulary databases. This study aims to design an Acehnese digital dictionary application with a better Autocomplete feature and website-based using the Levenshtein Distance Algorithm to improve the quality and ease of user access in finding and understanding Acehnese vocabulary. This study uses a data collection method by scanning the physical dictionary as a valid data source for the Aceh language digital dictionary. Then, the development of the Levenshtein Distance algorithm codebase was carried out and the implementation of the Algorithm on the search feature of the Aceh language digital dictionary. The Aceh language digital dictionary application with the Autocomplete feature and the Lenvenshtein Distance Algorithm has been successfully designed and implemented properly. Users can search for Acehnese vocabulary easily and quickly, as well as get word

recommendations when they make a typo or forget to type the character for the word they want to search for. In addition, the vocabulary database in this application is more complete and is expected to help users understand the Acehese language better. The Aceh language digital dictionary application with the Autocomplete feature and the Levenshtein Distance Algorithm designed in this study has succeeded in improving the quality and ease of user access in finding and understanding Acehese vocabulary on existing web applications. It is hoped that this application can assist in maintaining and preserving the existence of the Acehese language as part of the culture of the people of Aceh. This study uses 100 data as a test system, so that the percentages for testing different amounts of data can be changed.

Keywords: Dictionary Application, Digital Dictionary, Aceh Language, Autocomplete, Levenshtein Distance Algorithm.



KATA PENGANTAR

Bismillahirrahmanirrahim

Puji dan syukur kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan penulisan skripsi ini dengan baik. Shalawat dan salam tidak lupa kita sanjung sajikan kepada baginda Nabi besar Muhammad SAW yang telah membawa seluruh umatnya untuk menjadi generasi yang berilmu pengetahuan.

Puji dan syukur kehadiran Allah SWT yang memberikan rahmat serta hidayah sehingga peneliti mampu menyelesaikan Tugas Akhir ini dengan baik. Solawat serta salam pada baginda Nabi besar Muhammad SAW yang telah membawa seluruh umatnya menjadi berilmu pengetahuan seperti saat sekarang ini.

Dengan izin Allah SWT, peneliti dapat menyelesaikan penulisan Tugas Akhir dengan judul **“IMPLEMENTASI ALGORITMA LEVENSHTTEINDISTANCE UNTUK FITUR Pencarian Kata pada Pengembangan Kamus Bahasa Aceh Berbasis Web”** dengan harapan tulisan ini dapat memberikan manfaat bagi yang membutuhkan serta pada pembaca yang Budiman.

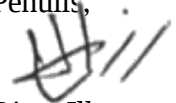
Dalam penulisan Tugas Akhir ini, tentunya banyak pihak yang berperan dalam proses penulisan, sehingga penulisan skripsi ini dapat diselesaikan, untuk itu peneliti ucapkan terima kasih banyak kepada:

1. Orang tua penulis Bapak Ridwan dan Ibu Aisyah dan keluarga yang telah mendo'akan, mendukung, memberikan semangat dan kasih sayang yang tidak

terhingga kepada keluarga. Dan kepada keluarga besar yang telah terlibat dalam proses pendidikan selama ini sehingga skripsi ini dapat diselesaikan.

2. Ibu Ima Dwitawati, M.B.A. dan Bapak Khairan Ar, M. Kom selaku Ketua dan Sekretaris Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Ar-Raniry, Banda Aceh.
3. Bapak Bustami, M. Sc. selaku pembimbing 1 dan Ibu Ima Dwitawati, M.B.A. selaku pembimbing 2 yang telah meluangkan waktu, fikiran untuk membantu dan membimbing peneliti demi kesempurnaan skripsi ini. Terima Kasih banyak penulis ucapkan. Semoga Bapak dan Ibu selalu mendapatkan rahmat dan Lindungan Allah SWT.
4. Khairan Ar, M. Kom. selaku Penasehat Akademik (PA) penulis selama menempuh pendidikan di Program Studi Teknologi Informasi. Terima kasih banyak telah memberi bimbingan dan saran selama ini kepada penulis.
5. Seluruh dosen yang mengajar pada Program Studi Teknologi Informasi yang telah memberikan ilmu pengetahuan yang sangat berguna bagi penulis selama proses belajar mengajar.
6. Ibu Cut Ida Rahmadiana S.Si. selaku staf prodi Teknologi Informasi yang telah membantu penulis dalam hal administrasi selama menempuh pendidikan di Program Studi Teknologi Informasi.
7. Sahabat serta teman-teman di Program Studi Teknologi Informasi dan Fakultas Sains dan Teknologi.

Banda Aceh, 01 Maret 2023
Penulis,



Rizal Ilham

DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR	iii
ABSTRAK	iv
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR PERSAMAAN	xiv
 BAB I PENDAHULUAN.....	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Batasan Masalah.....	5
 BAB II LANDASAN TEORI	 7
2.1 Penelitian Terdahulu	7
2.2 Bahasa Aceh.....	10
2.3 Kamus.....	11
2.4 Kamus Online	11
2.5 Website	12
2.6 Database.....	12
2.7 Algoritma.....	14
2.8 Algoritma Pencarian.....	14
2.9 Algoritma <i>Brute Force</i>	15
2.10 Algoritma <i>Kruth Morris Pratt</i>	15
2.11 Algoritma <i>Levenshtein Distance</i>	16
2.12 Bahasa Pemrograman Python.....	19
2.13 <i>HTML dan CSS</i>	19
2.14 Metode Evaluasi	20
 BAB III METODE PENELITIAN.....	 21
3.1 Jenis Penelitian	21
3.2 Tahapan Penelitian	22
3.3 <i>Select Data Resource</i>	22
3.4 <i>Collecting Data</i>	24
3.5 <i>Insert data to Database</i>	24
3.6 <i>Translate Algorithm Codebase</i>	25
3.7 <i>Waterfall Methodology</i>	25
3.8 <i>System Analysis</i>	26
3.9 <i>System Design</i>	29
3.10 <i>Coding and Algorithm Implementation</i>	32
3.11 <i>Testing</i>	32
3.12 <i>Maintenance</i>	33
3.13 <i>Tools dalam Penelitian</i>	33
3.14 <i>Tempat Penelitian</i>	34
3.15 <i>Jadwal Penelitian</i>	34

BAB IV HASIL DAN PEMBAHASAN	35
4.1 Hasil.....	35
4.2 Hasil Perbaikan Tampilan	35
4.3 Hasil Evaluasi <i>Similarity</i>	38
4.3 Hasil Evaluasi Akurasi	40
BAB V KESIMPULAN DAN SARAN.....	43
5.1 Kesimpulan.....	43
5.2 Saran.....	43
DAFTAR PUSTAKA	44
LAMPIRAN-LAMPIRAN	46



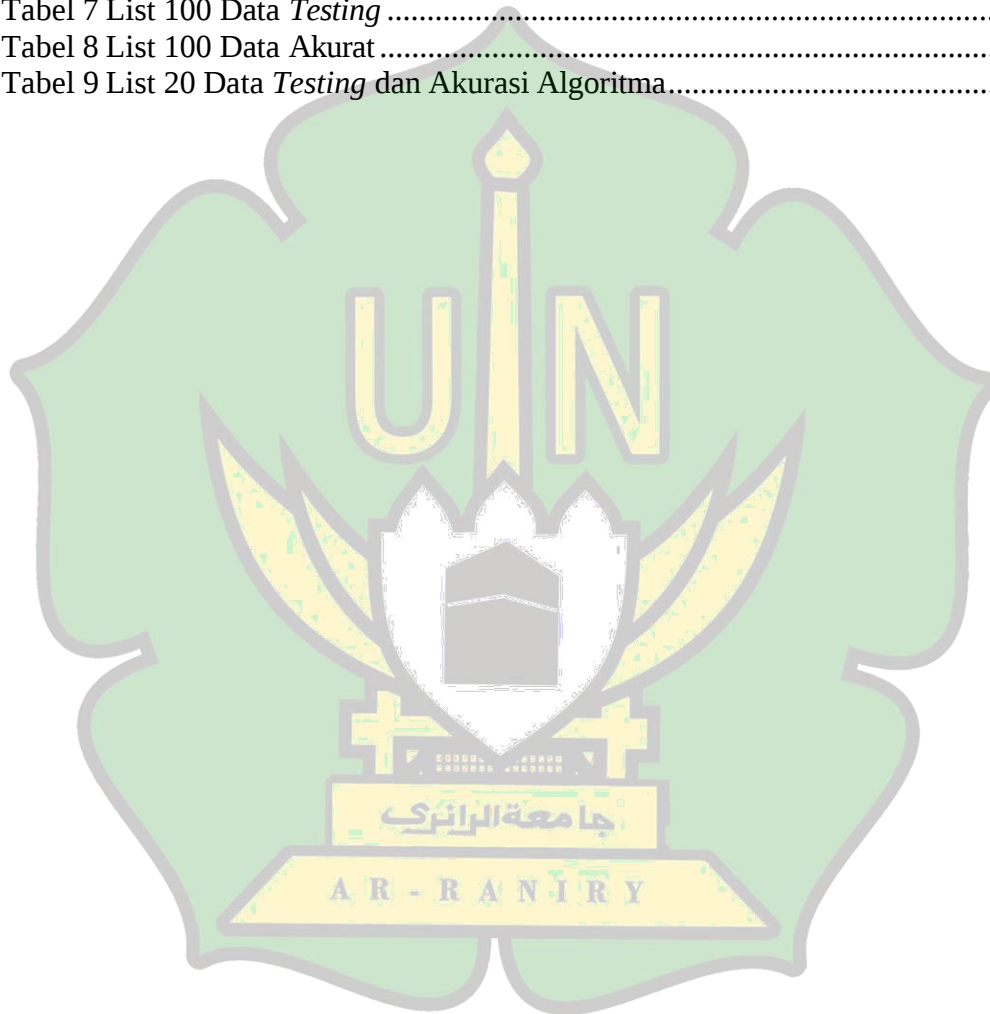
DAFTAR GAMBAR

Gambar 1 <i>Insertion</i>	17
Gambar 2 <i>Deletion</i>	17
Gambar 3 <i>Subtitution</i>	17
Gambar 4 Rumus <i>Levenshtein Distance</i>	18
Gambar 5 Tahapan Penelitian	22
Gambar 6 Cover Kamus Aceh Indonesia Inggris	23
Gambar 7 Halaman Depan Kamus.....	23
Gambar 8 <i>Flowchart</i> pengumpulan data.....	24
Gambar 9 <i>Flowchart</i> program untuk memasukkan data ke dalam database ...	25
Gambar 10 Alur Kerja Metode <i>Waterfall</i>	26
Gambar 11 <i>Activity Diagram</i> Alur Kegiatan Sistem.....	27
Gambar 12 ERD <i>Database</i>	27
Gambar 13 Struktur <i>Tabel Vocabularies</i>	30
Gambar 14 Struktur <i>Tabel Examples</i>	28
Gambar 15 Halaman Pencarian Aceh Indonesia.....	29
Gambar 16 Halaman pencarian Indonesia Aceh	29
Gambar 17 Tampilan rekomendasi Aceh Indonesia	29
Gambar 18 Tampilan Rekomendasi Indonesia Aceh.....	30
Gambar 19 Detail terjemahan Aceh Indonesia	30
Gambar 20 Detail terjemahan Indonesia Aceh	30
Gambar 21 Pencarian Aceh Indonesia	33
Gambar 22 Pencarian Indonesia Aceh.....	34
Gambar 23 Rekomendasi Aceh Indonesia	34
Gambar 24 Rekomendasi Indonesia Aceh	35
Gambar 25 Terjemahan Aceh Indonesia.....	36
Gambar 26 Terjemahan Indonesia Aceh.....	36



DAFTAR TABEL

Tabel 1 Matriks Perhitungan <i>Levenshtein Distance</i>	18
Tabel 2 Waktu Penelitian	32
Tabel 3 Persentasi kemiripan dua kata bahasa Aceh.....	37
Tabel 4 Detail Operasi yang terjadi 1.....	37
Tabel 5 Persentasi kemiripan dua kata bahasa Indonesia.....	40
Tabel 6 Detail operasi yang terjadi 2.....	40
Tabel 7 List 100 Data <i>Testing</i>	41
Tabel 8 List 100 Data Akurat	41
Tabel 9 List 20 Data <i>Testing</i> dan Akurasi Algoritma.....	42



DAFTAR PERSAMAAN

Persamaan 1 Rumus <i>Levenshtein Distance</i>	18
Persamaan 2 Rumus Mencari Similarity	32
Persamaan 3 Rumus Mendapatkan posisi kata	37
Persamaan 4 Rumus Mencari Akurasi	37



BAB I PENDAHULUAN

1.1 Latar Belakang

Bahasa adalah alat yang digunakan untuk berkomunikasi antar suatu kelompok komunitas tertentu, bahasa juga merupakan bagian inti dari kebudayaan komunitas tertentu. Bahasa yang kita gunakan sekarang merupakan hasil dari masyarakat masa lalu yang melakukan pengembangan dan pemeliharaan yang kemudian diwariskan kepada kita secara turun temurun, perkembangan bahasa sejalan dengan kebudayaan dari masyarakat pengguna bahasa itu sendiri (Herniti, 2010). Terdapat banyak sekali jenis bahasa yang digunakan untuk berkomunikasi, salah satu yang sering digunakan oleh masyarakat tradisional adalah bahasa daerah terkhusus di Indonesia.

Bahasa Daerah merupakan alat yang digunakan oleh suatu kelompok masyarakat tertentu untuk berkomunikasi dan saling berbagi informasi dengan sesama pengguna, sehingga keberadaan bahasa daerah sudah pasti erat kaitannya dengan daerah dan juga suku bangsa yang menggunakan bahasa tersebut. Namun, pada abad ke 21 ini masyarakat menganggap bahasa daerah kurang berpendidikan untuk digunakan sebagai bahasa utama jika dibandingkan bahasa asing dan bahasa Indonesia (Widianto & Pd, 2018). Salah satu penyebab terjadinya hal tersebut yaitu perubahan zaman, sehingga keberadaannya sedikit demi sedikit mulai pudar dan berdampak pada kepunahan (Kusumah & Cahyati, 2021). Salah satu bahasa daerah di Indonesia adalah bahasa Aceh.

Melihat pada hal yang dibahas pada paragraf sebelumnya, maka perlu upaya-upaya untuk menjaga dan melestarikan bahasa Aceh, salah satunya dengan

cara mendokumentasikan dalam bentuk fisik maupun dalam bentuk digital. Upaya dalam bentuk fisik sendiri sudah dilakukan oleh banyak pihak dengan cara membuat kamus dalam bentuk cetak. Namun, kamus dalam bentuk cetak memiliki banyak kekurangan seperti harus dibawa ke mana-mana yang tentunya akan menjadi beban bagi pengguna. Kamus bahasa Aceh dalam bentuk fisik juga sulit ditemukan di toko buku. Hal ini diperkuat setelah melakukan observasi di beberapa toko buku di kota Banda Aceh, belum ditemukan kamus berbahasa Aceh dalam bentuk fisik.

Cara lain untuk tetap menjaga kelestarian bahasa Aceh adalah dengan membuat kamus dalam bentuk digital. Kamus bahasa Aceh dalam bentuk digital memiliki beberapa keunggulan dibandingkan kamus cetak. Karena tidak perlu membawa kamus dalam bentuk fisik cukup hanya dengan memiliki akses internet.

Penelitian mengenai pengembangan kamus Bahasa Aceh digital telah banyak dilakukan, salah satunya adalah penelitian yang dilakukan oleh Mushaddiq yang berjudul “Perancangan Aplikasi Pembelajaran Bahasa Aceh Berbasis Android Untuk Guru PAUD Menggunakan Metode *User Centered Design (UCD)*”. Peneliti menjelaskan bahwa penggunaan metode *UCD* dalam perancangan aplikasi dapat memenuhi kebutuhan pengguna yaitu menampilkan data yang sudah disesuaikan dengan kurikulum PAUD Aceh, berupa gambar, kosakata dalam bahasa Aceh dan Indonesia serta contoh penggunaannya. Namun kekurangan pada penelitian ini tidak ada pembahasan mengenai *Autocomplete* pada fitur pencarian, tapi hanya membahas bagaimana metode *UCD* bekerja pada pengembangan aplikasi kamus bahasa Aceh berbasis Android (Firmansyah, 2020).

Selain itu penelitian yang dilakukan oleh Nuraini yang berjudul “Implementasi Algoritma *Kruth Morris Prath* Untuk Kamus Terjemahan Digital Aceh – Indonesia Berbasis Web”. Peneliti menjelaskan bahwa penggunaan Algoritma *Kruth Morris Prath* telah berhasil menerjemahkan bahasa Aceh ke dalam bahasa Indonesia begitu juga sebaliknya baik itu pencarian kata dasar maupun kalimat. Namun sistem yang diteliti tidak akan memberikan rekomendasi kata ketika data tidak di temukan di dalam *database* (Firmansyah, 2020).

Penelitian berikutnya berjudul “Perancangan dan Pembuatan Kamus Online Atraktif Bahasa Aceh Berbasis Website oleh” Riska Rahayu pada Tahun 2021. Penelitian ini sudah menghasilkan aplikasi kamus bahasa Aceh berbasis website yang baik dan sudah berhasil menerjemahkan kata yang dicari oleh pengguna. Namun terdapat beberapa kelemahan pada aplikasi yaitu, *database* yang digunakan hanya memiliki jumlah data 200 kosakata. Kelemahan lainnya adalah fitur pencarian yang terdapat pada aplikasi masih belum bisa bekerja secara optimal. Seperti tidak dapat memberikan rekomendasi kata ketika pengguna salah mengetik atau lupa mengetik karakter kata yang ingin dicari (Riska Rahayu, 2021). Kelemahan ini menjadi resiko pengguna tidak akan mendapatkan terjemahan dari kata yang diinginkan. Selain itu karena bahasa Aceh memiliki keunikan tersendiri pada penulisan beberapa karakter seperti “*é*” atau “*ô*”, tentunya dapat menjadi keterbatasan pengguna pada saat ingin mencari kata mengingat komputer saat ini tidak menyediakan karakter tersebut di *keyboard*.

Permasalahan ini dapat diselesaikan dengan satu metode yang dinamakan *Autocomplete*. Metode *Autocomplete* dapat mengoreksi, memperbaiki dan melengkapi kata pada fitur pencarian. Ada banyak algoritma yang bisa digunakan

untuk mengimplementasikan *Autocomplete* pada fitur pencarian, beberapa di antaranya seperti *Kruth Morris Prath* yaitu sebuah algoritma untuk mencari sebuah string dalam kumpulan string (Firmansyah, 2020), *Levenshtein Distance* yaitu sebuah algoritma yang digunakan untuk mencari jarak dua string (Bagus Ketut, 2018), *Brute Force* yaitu algoritma yang digunakan untuk melakukan pencocokan antara string (Sinaga, 2021).

Berdasarkan masalah yang sudah disebutkan, maka pada penelitian ini peneliti akan melakukan pengembangan aplikasi kamus bahasa Aceh berbasis website dengan penekanan pada penambahan jumlah kosakata dan perbaikan pada fitur pencarian kosakata bahasa Aceh yang baik dan benar. Fitur pencarian yang akan dikembangkan di sistem ini akan menggunakan konsep *Autocomplete* untuk melakukan pengoreksian kata secara otomatis dengan mengimplementasikan algoritma *Levenshtein Distance*.

Pemilihan algoritma ini ditentukan berdasarkan kebutuhan dari latar belakang bahasa yang memiliki keunikan pada beberapa karakter. Sehingga salah satu Teknik yang dapat dilakukan adalah dengan mencari jarak perbedaan antara dua buah string.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka peneliti dapat menguraikan beberapa rumusan masalah:

1. Bagaimana mengembangkan aplikasi kamus Bahasa Aceh berbasis Web?
2. Bagaimana mengimplementasikan Algoritma *Levenshtein Distance* untuk mendapatkan rekomendasi kata pada kamus bahasa Aceh berbasis Web?

1.3 Tujuan Penelitian

Berdasarkan latar belakang dan masalah yang sudah dituliskan di atas, maka tujuan dari penelitian ini antara lain:

1. Mengembangkan kamus Bahasa Aceh berbasis Web.
2. Mengimplementasikan Algoritma *Levenshtein Distance* pada fitur pencarian kamus Bahasa Aceh untuk mendapatkan rekomendasi kata.

1.4 Manfaat Penelitian

Berdasarkan latar belakang, rumusan masalah dan tujuan penelitian yang telah diuraikan maka diharapkan beberapa manfaat yang dapat diambil:

1. Memudahkan pengguna untuk mendapatkan kata yang diinginkan dengan memberikan saran beberapa kata yang sama atau hampir sama dengan kata yang diinginkan untuk dipilih.
2. Mencegah terjadinya kesalahan sistem yang tidak dapat menemukan kata yang tidak sesuai.
3. Memudahkan pengguna untuk alternatif tidak dapatnya mengetik karakter yang tidak ada di atas *keyboard*.

1.5 Batasan Masalah

Beberapa batasan masalah pada penelitian ini antara lain:

1. Data yang dikumpulkan hanya bekerja untuk 3 bahasa yaitu, Bahasa Aceh, Bahasa Indonesia dan Bahasa Inggris.

2. Penelitian ini membahas mengenai bagaimana penerapan metode *Levenshtein Distance* sebagai pencarian jarak kata tanpa melakukan perbandingan dengan metode lainnya.
3. Fokus utama pengembangan ini adalah pada sisi server untuk menghitung jarak antar kata yang dicari dengan seluruh data yang terdapat dalam *database* tanpa perlu menghawatirkan tampilan depan yang sederhana.



BAB II LANDASAN TEORI

2.1 Penelitian Terdahulu

2.1.1 Penelitian Kamus Bahasa Aceh

Penelitian mengenai pengembangan aplikasi kamus bahasa Aceh pernah dilakukan oleh Mushaddiq dengan judul “Perancangan Aplikasi Pembelajaran Bahasa Aceh Berbasis Android Untuk Guru PAUD menggunakan Metode *User Centered Design (UCD)*”. *UCD* merupakan metode perancangan yang menjadikan penggunanya sebagai pusat dari pengembangan sistem. Perancangan aplikasi dengan menggunakan metode *UCD* memiliki tiga tahapan yaitu *Prototyping*, *Design Database* dan yang terakhir implementasi perancangan menggunakan *Flutter Framework*. Peneliti menjelaskan bahwa penggunaan metode *UCD* dalam merancang aplikasi dapat memenuhi kebutuhan pengguna yaitu menampilkan data yang sudah disesuaikan dengan kurikulum Pendidikan Anak Usia Dini (PAUD) Aceh, berupa gambar, kosa kata dalam bahasa Aceh dan Indonesia serta contoh penggunaannya (Mushaddiq, 2020).

Penelitian kedua terkait pengembangan aplikasi kamus bahasa aceh dilakukan oleh Nuraini dengan judul “Implementasi Algoritma *Kruth Morries Prath* Untuk Kamus Terjemahan Digital Aceh – Bahasa Indonesia Berbasis Web”. Metode yang digunakan adalah *Kruth Morries Prath*. *Kruth Mories Prath* adalah algoritma yang berguna untuk mencari kata yang cocok dengan kata yang dicari oleh pengguna pada fitur pencarian kamus. Hasil dari penelitian ini adalah *Kruth Morries Prath* telah berhasil menerjemahkan bahasa Aceh ke dalam bahasa Indonesia begitu juga sebaliknya baik itu pencarian kata dasar maupun kalimat

(Firmansyah, 2020). Namun sistem yang diteliti tidak memberikan rekomendasi kata ketika tidak ditemukan kecocokan kata dengan data yang ada di dalam *database*.

Penelitian lainnya mengenai kamus bahasa Aceh pernah dikembangkan oleh Rizka Rahayu dalam penelitiannya yang berjudul “Perancangan dan Pembuatan Kamus Online Atraktif Bahasa Aceh Berbasis Website”. Pada penelitian tersebut peneliti memaparkan bagaimana membuat sistem kamus bahasa Aceh berbasis Website. Namun, sistem yang telah dikembangkan masih terbilang sederhana dan masih memerlukan fitur tambahan agar lebih baik lagi (Riska Rahayu, 2021). Salah satu fitur yang perlu ditambahkan adalah sistem yang dapat melakukan koreksi pada kata yang salah ketik dan menggantinya dengan karakter yang seharusnya, pelengkapan jumlah karakter kata jika pengguna lupa untuk mengetikkan karakter secara lengkap, menyediakan kata yang karakternya tidak tersedia di papan ketik.

2.1.2 Penelitian Algoritma *Levenshtein Distance*

Penelitian tentang penerapan *Levenshtein Distance* pernah diterapkan pada fitur pencarian sistem Informasi perpustakaan di Fakultas Kedokteran Universitas Palangka Raya untuk mencari data buku yang ada di perpustakaan. Hasil dari penelitian adalah sistem akan mengembalikan judul buku yang memiliki perbedaan karakter paling sedikit dengan kata yang diinputkan oleh pengguna untuk dijadikan rekomendasi yang ditampilkan kepada pengguna (Restiana, 2022).

Selain itu penerapan Algoritma *Levenshtein Distance* juga pernah dilakukan pada fitur pencarian kata kamus Aneka Bahasa. Sistem yang

dikembangkan bekerja untuk menerjemahkan satu bahasa ke bahasa Bali, Indonesia, Inggris, Prancis dan Jerman, hasil dari penerapan algoritma ini pengguna dapat menterjemahkan kata dengan hasil yang optimal (Bagus Ketut, 2018).

Penelitian lainnya mengenai penerapan algoritma *Levenshtein Distance* juga pernah dilakukan pada pengembangan aplikasi kamus digital berbasis Android. Output yang dihasilkan dari penelitian ini adalah aplikasi kamus terjemahan dari bahasa Minang ke Indonesia dan sebaliknya yang sudah menerapkan algoritma *Levenshtein Distance* pada fitur pencariannya untuk mengoreksi kesalahan karakter yang dimasukkan oleh pengguna (Sari & Umami, 2020).

Penelitian berikutnya mengenai penerapan *Levenshtein Distance* pernah dilakukan pada sistem yang mendeteksi plagiarisme pada dokumen. Latar belakang yang melandasi dari penelitian ini adalah banyaknya kaum intelektual yang melakukan plagiasi pada laporan ataupun dokumen yang digunakan untuk keperluan akademik. Kesimpulan dari penelitian ini yaitu, Algoritma *Levenshtein Distance* cocok digunakan untuk melakukan pemeriksaan plagiasi pada dokumen yang dimasukkan namun Algoritma *Levenshtein distance* harus diterapkan pada posisi yang tepat untuk memeriksa 2 dokumen, karena pemeriksaan dilakukan per karakter di dalam *string* dengan kata lain Algoritma tidak akan bekerja secara baik untuk dokumen yang panjang dengan struktur *string* yang tidak teratur.

Berdasarkan penelitian yang telah disebutkan, belum ada penelitian mengenai masalah pencarian pada kamus bahasa Aceh jika pengguna salah mengetikkan karakter dari suatu kata atau pengguna menulis kata dengan karakter

yang tidak sempurna. Dapat dilihat bahwa penelitian saat ini mengenai pemecahan masalah untuk pencarian dalam kamus bahasa Aceh hanya mampu mencari kata jika karakter yang diketikkan sama persis dengan kata yang ada di *database*. Oleh karena itu peneliti ingin melakukan penelitian mengenai bagaimana memecahkan masalah jika pengguna salah mengetikkan karakter pada kata yang ingin dicari atau pengguna tidak lengkap menulis karakter kata yang dicari, namun pengguna tetap bisa mendapatkan rekomendasi kata untuk dipilih dengan menggunakan algoritma *Levenshtein Distance* pada aplikasi yang sudah dikembangkan oleh Riska pada penelitian.

2.2 Bahasa Aceh

Bahasa Aceh merupakan bahasa yang banyak dituturkan oleh masyarakat yang tinggal di daerah Aceh untuk berkomunikasi antar sesama. Kelompok masyarakat yang berada luar Aceh sering menggunakannya sebagai alat berkomunikasi, karena bahasa Aceh juga merupakan bahasa dasar dan bahasa ibu dari masyarakat Aceh yang berada luar kota (Devi, 2021). Bahasa Aceh masih aktif digunakan sampai saat ini di kawasan provinsi Aceh. Bahasa Aceh memiliki penutur yang paling banyak jika di bandingkan dengan bahasa lainnya di Kawasan provinsi Aceh. Bahasa Aceh memiliki keunikan tersendiri dari bahasa daerah lainnya, mulai dari vokal penuturan, penulisan huruf vokal yang sebagian memiliki tanda petik di bagian atas huruf (Hifra & Razali, 2022). Tentu vokal unik tersebut tidak terdapat dalam *keyboard* komputer.

Bahasa Aceh penting untuk dilestarikan untuk mempertahankan eksistensi bahasa tersebut. Karena bahasa Aceh merupakan suatu kearifan lokal masyarakat

Aceh. Bahasa daerah memiliki resiko kepunahan yang lebih tinggi dibandingkan dengan bahasa nasional. Disebabkan karena penuturan bahasa daerah hanya berada di tempat suku penutur bahasa itu tinggal. Sedangkan bahasa nasional memiliki resiko kepunahan lebih rendah dikarenakan pengguna bahasa tersebut tersebar lebih luas dibandingkan dengan bahasa daerah itu sendiri (Hifra & Razali,2022).

2.3 Kamus

Kamus merupakan buku yang berisi pilihan kata-kata dari suatu bahasa yang sudah disusun berdasarkan alfabet. Kamus memberikan informasi makna, pengungkapan, etimologi, bentuk infleksi dan lain-lainnya yang dapat diungkapkan dalam bahasa yang sama atau bahasa lainnya (Khusnul Khotimah, 2021).

2.4 Kamus Online

Kamus online sendiri merupakan kamus bahasa yang dibuat dalam bentuk digital seperti Web, Desktop atau Mobile. Kelebihan dari kamus online adalah dapat diakses secara real-time, di manapun yang memiliki akses internet. Aplikasi seperti Play Store ataupun App Store sudah banyak tersedia aplikasi digital yang dapat diinstall pada perangkat Mobile. baik itu online maupun offline, versi online memiliki data yang terpusat dan dipegang oleh pengembang aplikasi. Sedangkan data untuk aplikasi offline biasanya jarang diupdate karena data langsung dimasukkan ke dalam aplikasi dan di simpan di perangkat pengguna aplikasi tersebut.

2.5 Website

Website merupakan media penyampaian informasi secara online yang diakses melalui internet dan dapat diakses kapanpun dan di manapun. Website juga sering didefinisikan sebagai kumpulan halaman yang menyampaikan informasi gambar, *text*, video, audio dan lainnya (Hani Saniatun, 2022). Aplikasi Web dapat dirancang untuk berbagai kegunaan dan dapat digunakan oleh siapapun tidak terbatas pada pengembangan aplikasi kamus.

2.6 Database

Database merupakan struktur data yang menyimpan informasi secara teorganisasi. Pada umumnya penggunaan *database* di susun dari beberapa tabel yang berbeda. Pada penelitian ini *database* digunakan untuk menyimpan data semua kosakata yang diambil dari salah satu kamus bahasa Aceh secara digital.

Terdapat banyak jenis *database*, berikut peneliti uraikan beberapa jenis *database*:

1. Database Terpusat

Ini adalah jenis *database* yang menyimpan data pada sistem *database* terpusat. Ini kenyamanan pengguna untuk mengakses data yang disimpan dari lokasi yang berbeda melalui beberapa aplikasi.

2. Database Terdistribusi

Tidak seperti sistem *database* terpusat, dalam sistem terdistribusi, data didistribusikan di antara sistem *database* yang berbeda dari suatu

organisasi. Sistem *database* terhubung melalui tautan komunikasi. Tautan semacam itu membantu pengguna untuk mengakses data dengan mudah.

3. *Relational Database*

Database ini didasarkan pada model data relasional, yang menyimpan data dalam bentuk baris (*tuple*) dan kolom (*atribut*), dan bersama-sama membentuk tabel (relasi). *Database* relasional menggunakan SQL untuk menyimpan, memanipulasi, serta memelihara data.

4. *NoSQL Database*

NoSQL / Not Only SQL adalah jenis *database* yang digunakan untuk menyimpan berbagai kumpulan data. Ini bukan *database* relasional karena menyimpan data tidak hanya dalam bentuk tabel tetapi dalam beberapa cara berbeda. Itu muncul ketika permintaan untuk membangun aplikasi modern meningkat. Dengan demikian, *NoSQL* menghadirkan berbagai macam teknologi *database* sebagai tanggapan atas tuntutan tersebut.

5. *Cloud Database*

Jenis *database* tempat data disimpan dalam lingkungan virtual dan dijalankan melalui platform komputasi awan.

6. *Database Berorientasi Objek*

Jenis *database* yang menggunakan pendekatan model data berbasis objek untuk menyimpan data dalam sistem *database*. Data direpresentasikan dan disimpan sebagai objek yang mirip dengan objek yang digunakan dalam bahasa pemrograman berorientasi objek.

Dari banyaknya jenis *database* yang telah diuraikan, pada pengembangan ini peneliti menggunakan *Relational Database* sesuai dengan kebutuhan untuk penyimpanan data yang akan digunakan untuk menjalankan sistem.

2.7 Algoritma

Algoritma adalah sekumpulan tindakan yang perlu dilakukan untuk menyelesaikan suatu permasalahan komputasi secara berurutan, logis dan sistematis. Algoritma pertama kali ditemukan pada tahun 770 M oleh seorang ilmuwan terkenal yang bernama Abu Ja'far Muhammad Ibnu Musa al-Khwarizmi.

Algoritma sangat erat kaitannya dengan pemrograman komputer. Namun penerapan algoritma tidak hanya sebatas pada komputer, tapi secara tidak sadar terimplementasikan dalam kehidupan sehari-hari. Contoh dasar penerapan algoritma dalam kehidupan nyata adalah proses memasak yang memiliki tahap-tahap terstruktur untuk membuat makanan (Khairani Aritonang, 2022).

2.8 Algoritma Pencarian

Algoritma pencarian merupakan suatu algoritma yang menerima suatu inputan data yang kemudian melakukan pencarian sesuai dengan data yang

dimasukkan, pencarian ini akan menghasilkan 2 kemungkinan yaitu data ditemukan atau data tidak ditemukan (Sinaga, 2021). Terdapat beberapa algoritma yang dapat digunakan untuk melakukan pencarian seperti Algoritma *Brute Force*, Algoritma *Kruth Morris Pratt*, Algoritma *Levenshtein Distance* dan masih banyak lagi yang tidak mungkin disebutkan dalam penelitian ini.

2.9 Algoritma *Brute Force*

Algoritma *Brute Force* merupakan salah satu algoritma pencarian yang bekerja dengan cara melakukan pengecekan setiap karakter yang ada dalam kata mulai dari karakter yang pertama sampai karakter yang terakhir. Setelah karakter pertama selesai dibandingkan dan ditemukan kecocokan maka algoritma secara otomatis akan pindah ke karakter selanjutnya. Begitu sampai semua karakter dalam *string* selesai dibandingkan (Sinaga, 2021). Kelebihan dari algoritma *Brute Force* adalah mampu memecahkan sebagian besar masalah sederhana dan juga mudah di mengerti. Algoritma *Brute Force* layak untuk beberapa masalah seperti pencarian kata, pengurutan kata dan juga pencocokan *string* (Rismayani, 2021). Namun kelemahan dari algoritma ini membutuhkan waktu yang cukup lama dan proses komputasi yang besar untuk membandingkan setiap karakter *string* (Sinaga, 2021).

2.10 Algoritma *Kruth Morris Pratt*

Algoritma *Kruth Morris Pratt* merupakan jenis algoritma pencarian *string* yang melakukan pengecekan karakter dari kiri ke kanan. Algoritma *Kruth Morris Pratt* bekerja lebih efisien dari pada algoritma *Brute Force*, karena algoritma

Kruth Morris Pratt bekerja dengan langsung melakukan pergeseran beberapa karakter jika karakter yang dipasangkan dengan karakter yang sudah cocok tidak sama dengan struktur kata yang dibandingkan. Kemudian memasukkan karakter yang tidak cocok tersebut ke dalam fungsi yang disebut *Failure Function*, perbandingan dilakukan hingga semua karakter yang dicari selesai diperiksa satu persatu. Algoritma *Kruth Morris Pratt* termasuk ke dalam jenis pencarian *Exact Matching* (Nababan & Jannah, 2019). Yang mana *string* yang dicari harus sama persis dengan data yang ada di dalam sistem tidak boleh ada perbedaan pada struktur kata.

2.11 Algoritma Levenshtein Distance

Algoritma *Levenshtein Distance* merupakan algoritma yang digunakan untuk mencari jarak perbedaan antara dua kata dalam *string* (Widiatry, 2019). Dengan menggunakan algoritma *Levenshtein Distance* sistem akan menghitung perbedaan antara 2 *string* yang mana *string* diambil dari dalam *database* satu persatu dan melakukan perbandingan. Kata dengan jarak perbedaan paling kecil merupakan kata yang paling akurat hitungannya dengan kata yang dicari oleh pengguna (Adawiyah & Saragih, 2022). Setelah mendapatkan hasil perbedaan jarak kata yang dicari, sistem akan mengembalikan 10 kata dengan jarak perbedaan terkecil untuk dijadikan rekomendasi kata yang bisa dipilih oleh pengguna di bagian bawah *form* pencarian.

Levenshtein Distance pertama kali ditemukan pada tahun 1965 oleh Vladimir Levenshtein. Untuk mengetahui perbedaan 2 buah *string* yang dibandingkan algoritma ini menggunakan 3 operasi yang harus dilakukan

berdasarkan dari keadaan kata yang dimasukkan oleh pengguna. Berikut adalah beberapa operasi yang dilakukan dalam Algoritma *Levenshtein Distance*:

2.11.1 *Insertion* (Penyisipan Karakter)

Operasi ini digunakan untuk memasukkan karakter yang tidak lengkap dimasukkan oleh pengguna berikut contoh kasus yang umum terjadi pada penggunaan kesalahan ketik pada umumnya pada kata 'ello:

KATA 1	H	E	L	L	O
KATA 2	H	E	-	L	O
KARAKTER KURANG			L		

Gambar 1 *Insertion*

Pada gambar di atas kata **“Hello”** tidak diketikan dengan lengkap yaitu kekurangan satu karakter yaitu **“l”**, maka pada kasus seperti ini operasi *insertion* dibutuhkan dalam algoritma ini.

2.11.2 *Deletion* (Penghapusan Karakter)

Operasi ini akan bekerja ketika pengguna mengetik kelebihan karakter dari yang seharusnya seperti pada gambar berikut:

KATA 1	H	E	L	L	O	W
KATA 2	H	E	L	L	O	
KARAKTER LEBIH						W

Gambar 2 *Deletion*

Pada gambar urutan karakter dari kata **“Hellow”** terdapat kelebihan karakter **“w”**, kata pada kasus seperti ini diperlukannya operasi penghapusan karakter yang lebih dari kata yang seharusnya.

2.11.3 *Subtitution* (Penukaran Karakter)

Operasi ini bekerja Ketika pengguna salah mengetikkan karakter yang berbeda dengan karakter seharusnya dari sebuah kata, berikut contoh kasus yang biasa terjadi:

KATA 1	H	E	L	L	Ô
KATA 2	H	E	L	L	O
KARAKTER LEBIH					Ô

Gambar 3 Subtitution

Pada tabel kata “*Gabôh*” pengguna salah memasukkan karakter “*o*” yang mana karakter yang seharusnya dimasukkan adalah karakter “*ô*”, maka pada kasus seperti inilah diperlukannya Operasi ini untuk menukar katakter yang salah dengan karakter yang seharusnya.

Persamaan 1 adalah rumus bagaimana *Levenshtein Distance* menghitung perbedaan jarak antara 2 *string*:

$$\text{lev}(a, b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \text{lev}(\text{tail}(a), \text{tail}(b)) & \text{if } a[0] = b[0], \\ 1 + \min \begin{cases} \text{lev}(\text{tail}(a), b) \\ \text{lev}(a, \text{tail}(b)) \\ \text{lev}(\text{tail}(a), \text{tail}(b)) \end{cases} & \text{otherwise,} \end{cases}$$

Levenshtein Distance
(Sumber wikipedia.org)

Cara kerja rumus tersebut dapat ilustrasikan seperti contoh pada tabel matriks berikut ini:

Tabel 1 Perhitungan *Levenshtein Distance*

		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

Secara sederhana, tabel di atas bekerja dengan cara membandingkan dua buah string kemudian menambahkan nilai 1 kepada setiap katakter yang berbeda antara dua string, hasil akhir jumlah perbedaan terletak pada kolom dan baris terakhir pada tabel.

2.12 Bahasa Pemrograman Python

Setiap sistem pada saat ini pasti dibangun menggunakan bahasa pemrograman. Python merupakan salah satu bahasa pemrograman yang memiliki sintak yang sederhana, mudah dipahami dan dipelajari (Sabri, 2021). Python memiliki banyak *library* yang bisa digunakan untuk membantu memecahkan masalah yang sudah pernah dipecahkan oleh Developer lain. Pada penelitian ini fokus utama peneliti adalah bagaimana menerapkan Algoritma *Levenshtein Distance* dalam bahasa pemrograman Python dan bagaimana menerapannya pada sistem Web.

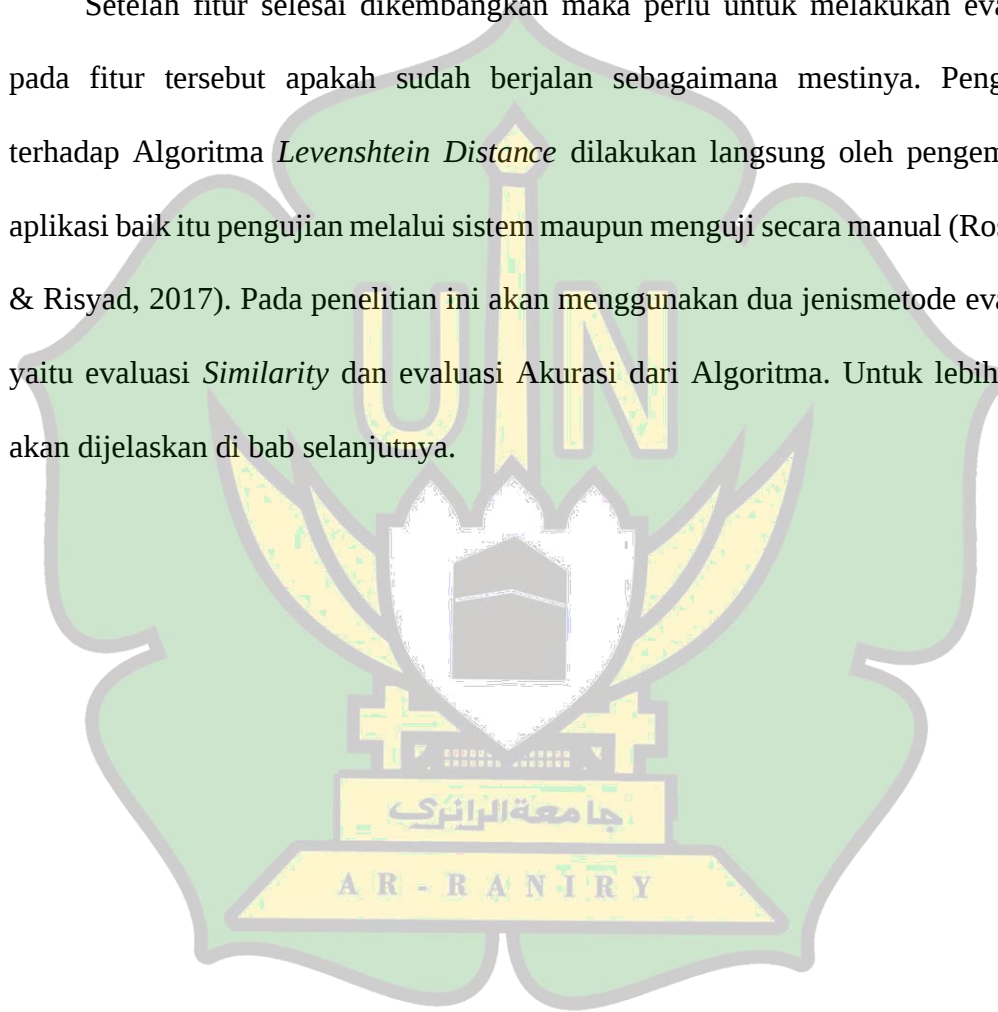
2.13 HTML dan CSS

HTML merupakan singkatan dari *Hypertext Markup Language*. Yang mana ini merupakan bahasa template yang digunakan untuk membuat halaman dasar

sedangkan CSS singkatan dari *Cascading Style Sheet* yang mana bahasa template yang satu ini digunakan untuk membuat tampilan halaman web menjadi lebih menarik ketika dipandang.

2.14 Metode Evaluasi

Setelah fitur selesai dikembangkan maka perlu untuk melakukan evaluasi pada fitur tersebut apakah sudah berjalan sebagaimana mestinya. Pengujian terhadap Algoritma *Levenshtein Distance* dilakukan langsung oleh pengembang aplikasi baik itu pengujian melalui sistem maupun menguji secara manual (Rosmala & Risya, 2017). Pada penelitian ini akan menggunakan dua jenis metode evaluasi yaitu evaluasi *Similarity* dan evaluasi Akurasi dari Algoritma. Untuk lebih jelas akan dijelaskan di bab selanjutnya.



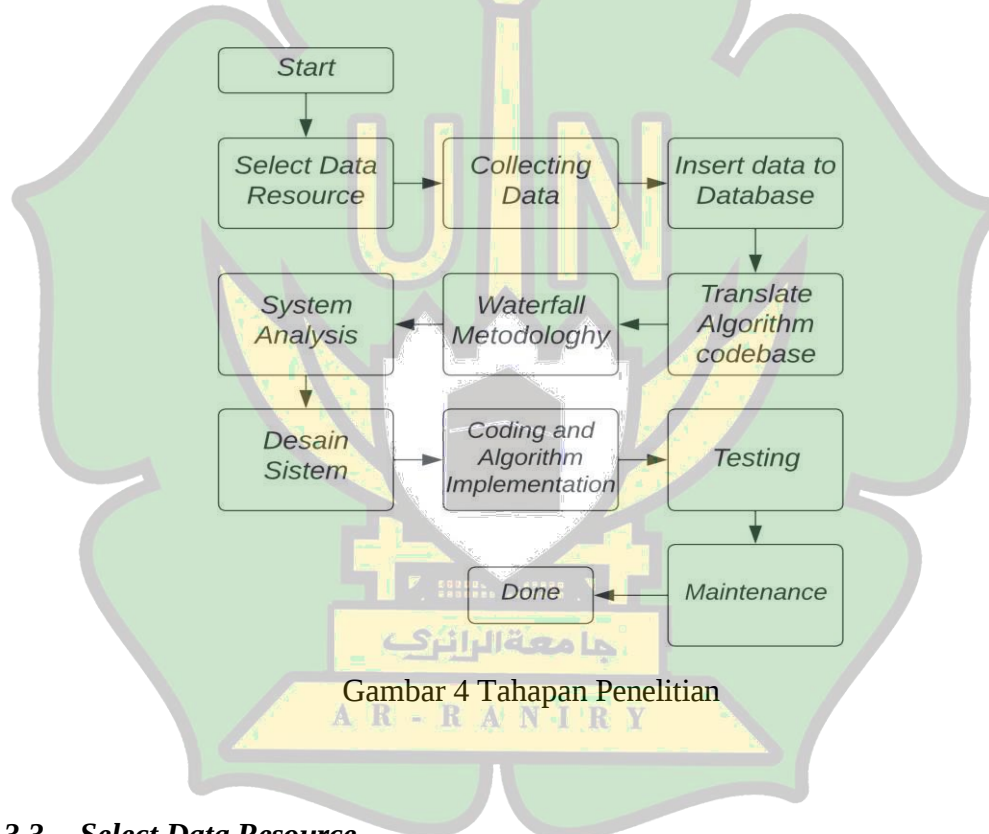
BAB III METODE PENELITIAN

3.1 Jenis Penelitian

Pada penelitian ini, peneliti menggunakan jenis penelitian kuantitatif, yaitu memecahkan yang sudah dipaparkan pada bab sebelumnya dengan menggunakan solusi yang sistematis dan perlu dilakukannya perhitungan untuk mengetahui jarak perbedaan antara kata yang dimasukkan dengan seluruh kata yang ada di dalam *database*. Pemilihan jenis penelitian ini didasarkan pada masalah yang masih terdapat pada penelitian yang sudah dilakukan oleh Riska Rahayu dengan judul *“Perancangan dan Pembuatan Kamus Online Atraktif Kamus Bahasa Aceh Berbasis Website”*. Sistem yang dikembangkan dalam penelitian tersebut belum memiliki algoritma pencarian kata yang dapat memberikan rekomendasi kata yang mirip ketika pengguna salah mengetikkan karakter kata, kelebihan karakter yang diketik atau bahkan pengguna mengetik kata yang tidak ada di dalam *database*. Sejauh ini sistem yang sudah dikembangkan akan mengembalikan *error* ketika tidak menemukan data yang dicari pengguna dari dalam *database*.

3.2 Tahapan Penelitian

Dalam melakukan penelitian ini, terdapat beberapa tahap yang harus dilakukan mulai dari memilih sumber kamus fisik yang akan digunakan sebagai main data pada sistem, mengoleksi kata ke dalam *file* digital, memasukkan data ke dalam *database*, menentukan metode pengembangan sistem, menentukan kebutuhan dari sistem yang akan dikembangkan, melakukan desain, implemementasi *coding*, pengujian sistem sampai siap untuk digunakan dan dilakukan pemeliharaan, tahapan pada penelitian ini dapat dilihat pada gambar 4:

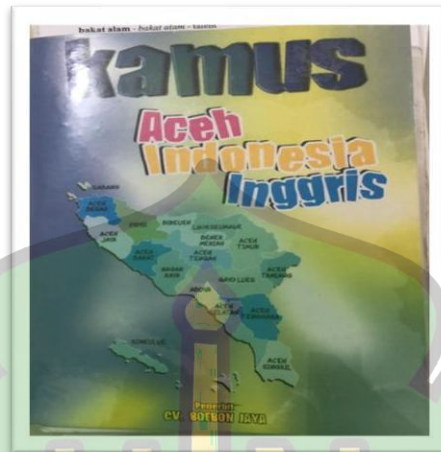


Gambar 4 Tahapan Penelitian

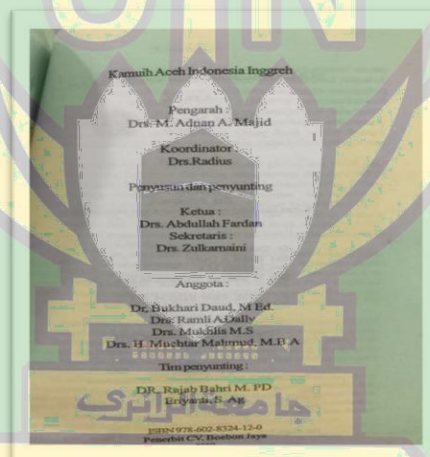
3.3 Select Data Resource

Pada tahapan pertama pada penelitian ini adalah menentukan sumber data yang akan digunakan oleh sistem sebagai data utama yang akan digunakan ditahap *production* (aplikasi rilis). Kamus yang digunakan sebagai sumber data pada penelitian ini berjudul *Kamus Aceh-Indonesia-Inggris* yang ditulis oleh Drs. Abdullah Fardan yang kemudian diterbitkan oleh CV. Boebon Jaya pada tahun

2019. Kata yang ada di dalam kamus ini terdiri dari 2 bahasa terjemahan dari bahasa Aceh yaitu Indonesia dan Inggris, bagian cover dan halaman depan kamus dapat dilihat pada gambar 5 dan gambar 6.



Gambar 5 Cover Kamus Aceh Indonesia Inggris

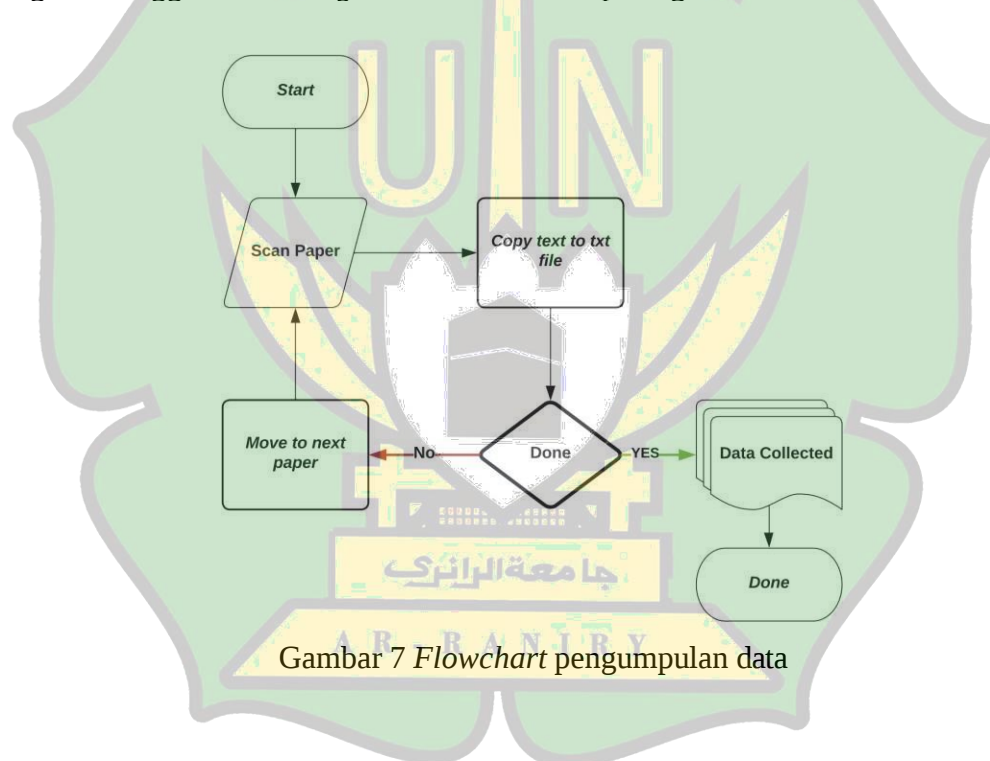


Gambar 6 Halaman Depan Kamus

Jenis data yang digunakan pada sistem ini merupakan data sekunder yang didapatkan dari memindai kamus fisik secara manual dan memindahkan data tersebut ke dalam bentuk digital, kosakata dalam kamus yang dipindai merupakan kosakata yang sudah dikumpulkan oleh pihak ataupun orang lain yang kemudian dicetak dalam bentuk fisik.

3.4 Collecting Data

Pada tahap pengoleksian data, kosakata kamus bahasa Aceh yang diterjemahkan ke dalam bahasa Indonesia dan bahasa Inggris, perlu dijadikan dalam bentuk digital sehingga dapat dimasukkan ke dalam *database*. Tools yang digunakan adalah Google Lens yang berfungsi untuk memindai setiap halaman kamus dan *meng-copy text* yang terdeteksi dan kemudian *text* tersebut dapat ditempelkan ke aplikasi text editor yang ada di komputer utama sebagai tempat penampung data. *Flowchart* pengumpulan data dari bentuk fisik ke dalam bentuk digital menggunakan *Google Lens* bisa dilihat pada gambar 7.

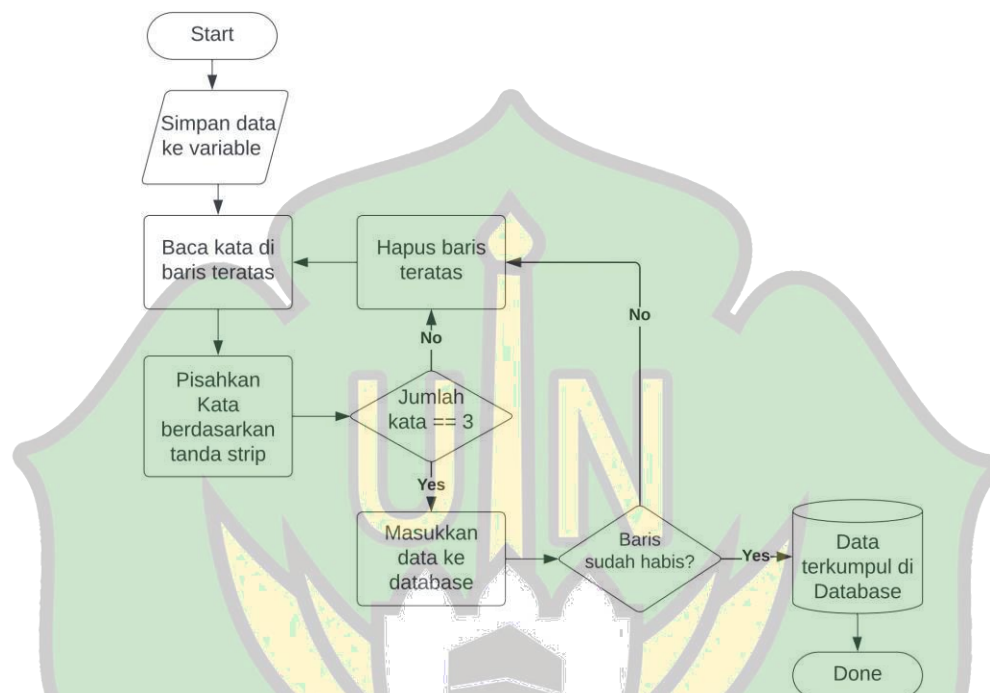


Gambar 7 *Flowchart* pengumpulan data

3.5 Insert data to Database

Setelah data terkumpulkan dalam bentuk digital dan sudah di pilah, maka data tersebut harus dimasukkan ke dalam *database* sehingga dapat diintegrasikan dan di olah dengan *script programming* yang akan dibuat. Teknik memasukkan data ke dalam *database* adalah dengan menjalankan satu program yang akan

peneliti bangun khusus yang berfungsi untuk mengeksekusi *file* yang sudah berisi kosakata bahasa Aceh yang sudah diterjemahkan ke dalam dua bahasa lainnya. Tujuan dimasukkan dalam ke dalam *database* agar data dapat diakses oleh sistem. *Flowchart* program dapat dilihat pada gambar 8.



Gambar 8 *Flowchart* program untuk memasukkan data ke dalam *database*

3.6 Translate Algorithm Codebase

Pada tahap ini Algoritma *Levenshtein Distance* yang digunakan dikembangkan menggunakan bahasa pemrograman *Python* yang kemudiannantinya akan dijalankan pada sisi *server*.

3.7 Waterfall Methodology

Metode yang digunakan untuk mengembangkan sistem dalam penelitian ini adalah "*Sistem Development Life Cicle*" Jenis *Waterfall*. *Waterfall* sendiri merupakan jenis pengembangan sistem yang bersifat sistematis serta berurutan,

tahapan penerapan *waterfall* dalam pengembangan sistem ini meliputi menentukan *requirements* (Analisis Kebutuhan Sistem), membuat *Design sistem*, mengimplementasikan (Penulisan Kode Sistem), Melakukan pengujian dan yang terakhir melakukan pemeliharaan sistem. Alur penerapan *waterfall* pada pengembangan sistem dapat dilihat pada gambar berikut.



Gambar 9 Alur Kerja Metode *Waterfall*.

3.8 System Analysis

Tahap pertama yang diperlukan sebelum mengembangkan adalah menentukan kebutuhan sistem. Terdapat dua jenis kebutuhan sistem yaitu Kebutuhan Functional dan Kebutuhan Non Functional. Berikut adalah hasil analisis pada sistem yang sudah ada saat ini setelah disesuaikan dengan kebutuhan pada penelitian.

3.8.1 Kebutuhan *Functional*

- a. User bisa memasukkan kata ke dalam *form* pencarian dan menekan tombol cari pada halaman web.

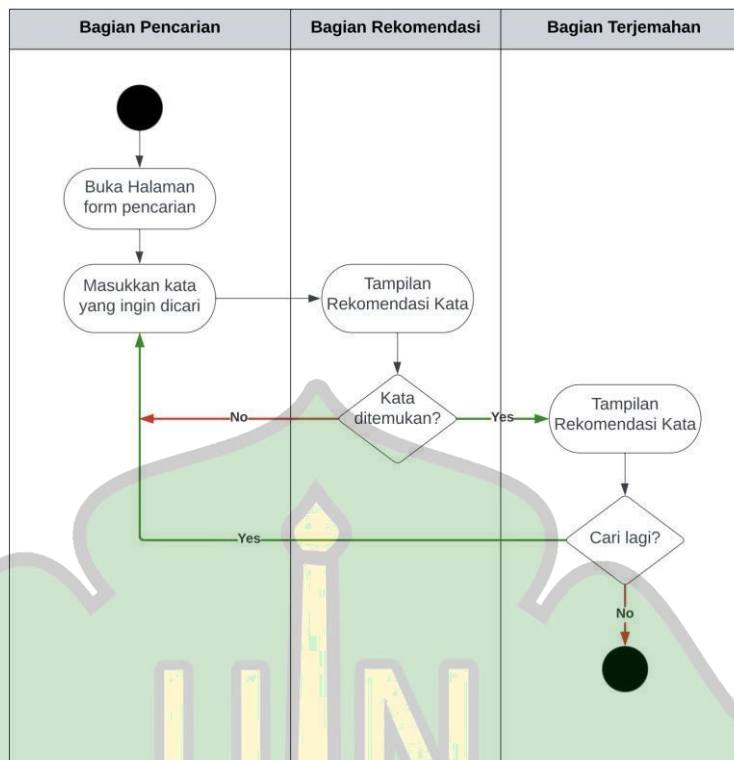
- b. Sistem harus bisa menghitung jarak perbedaan secara akurat setelah tombol cari ditekan.
- c. User bisa melihat beberapa rekomendasi kata yang memiliki jarak perbedaan karakter paling sedikit dengan kata yang dimasukkan setelah menekan tombol cari.
- d. User bisa mencari kata dari Bahasa Aceh ke Bahasa Indonesia.
- e. User bisa mencari kata dari Bahasa Indonesia ke Bahasa Aceh.
- f. User bisa melihat detail kata setelah menekan salah satu kata yang direkomendasikan.

3.8.2 *Kebutuhan Non functional*

- a. Dapat diakses menggunakan Google Chrome, Safari atau mesin pencari lainnya.
- b. Dapat diakses secara online dari mana pun.
- c. Tampilan yang sederhana guna untuk mengarahkan pengguna agar lebih fokus pada kegunaan sistem.

3.8.3 *Diagram Activity*

Activity diagram pada fitur pencarian yang sudah dikembangkan dapat dilihat pada gambar 10.



Gambar 10 Activity Diagram alur kegiatan sistem

3.8.4 Perancangan Database

a. Entity Relationship Diagram

Entity Relationship Diagram atau ERD adalah pemodelan struktur dan hubungan antar data bagaimana data tersebut tersimpan di dalam *database*.

Vocabularies		Examples	
Index	Integer	Index	Integer
Aceh	Varchar	aceh_id	Varchar
Indonesia	Varchar	contoh_aceh	Varchar
English	Varchar	contoh_indonesia	Varchar
Jawoe	Varchar	contoh_english	Varchar

Gambar 11 ERD database

b. *Structure Table Vocabularies*

Tabel Vocabularies digunakan untuk menyimpan data semua kosakata yang akan digunakan oleh sistem.

Column Name	#	Data Type	Length	Not Null	Auto Increment
123 index	1	INTEGER		[v]	[]
RBC aceh	2	VARCHAR	150	[]	[]
RBC indonesia	3	VARCHAR	150	[]	[]
RBC english	4	VARCHAR	150	[]	[]
RBC jawoe	5	VARCHAR	150	[]	[]

Gambar 12 Struktur *Tabel Vocabularies*

c. *Structure Tabel Examples*

Tabel Examples digunakan untuk menyimpan data contoh yang berhubungan dengan kosakata jika ada.

Column Name	#	Data Type	Length	Not Null	Auto Increment
123 index	1	INTEGER		[v]	[]
RBC aceh_id	2	VARCHAR	150	[]	[]
RBC contoh_aceh	3	VARCHAR	200	[]	[]
RBC contoh_indonesia	4	VARCHAR	200	[]	[]
RBC contoh_english	5	VARCHAR	200	[]	[]

Gambar 13 Struktur *Tabel Examples*

3.9 *System Design*

Setelah analisis selesai dilakukan maka tahap selanjutnya adalah membuat desain tampilan dari sistem yang akan dikembangkan, desain tampilan sistem dapat dilihat pada gambar berikut.

Gambar 14 adalah tampilan halaman pencarian Aceh-Indonesia

Gambar 14 Halaman pencarian Aceh – Indonesia

Gambar 15 adalah tampilan halaman pencarian Indonesia-Aceh

Gambar 15 Halaman pencarian Indonesia – Aceh

Gambar 16 adalah tampilan rekomendasi Aceh-Indonesia

No	Aceh	Detail
1	Semangat (Semangat)	Pilih
2	Gubdh (Gubuh)	Pilih
3	Mimah (Pindah)	Pilih
4	Nam (Nama)	Pilih

Copyright @ Indonesia Aceh 2022

Gambar 16 Tampilan Rekomendasi Aceh – Indonesia

Gambar 17 adalah tampilan rekomendasi Indonesia-Aceh

The screenshot shows a web interface for 'Kamus Indonesia Aceh'. At the top, there are navigation links: 'TRANSLATE', 'KAMUS', 'SARAN TERJEMAHAN', and 'REQUEST TERJEMAHAN'. Below this, the title 'Kamus Indonesia Aceh' is displayed. A search bar contains the word 'Semangat', and a 'Cari' button is next to it. Below the search bar, a message says 'Terjemahan akan ditampilkan disini'. Underneath, a section titled 'Rekomendasi untuk Semangat' contains a table with four rows of suggestions.

No	Indonesia	Detail
1	Semangat (Saumangat)	PSN
2	Gabus (Gaduh)	PSN
3	Pindah (Mudah)	PSN
4	Nama (Hari)	PSN

At the bottom, a footer reads 'Copyright @ Indonesia Aceh 2022'.

Gambar 17 Tampilan Rekomendasi Indonesia – Aceh

Gambar 18 adalah tampilan detail terjemahan Aceh-Indonesia

The screenshot shows a web interface for 'Kamus Aceh Indonesia'. At the top, there are navigation links: 'TRANSLATE', 'KAMUS', 'SARAN TERJEMAHAN', and 'REQUEST TERJEMAHAN'. Below this, the title 'Kamus Aceh Indonesia' is displayed. A search bar contains the word 'Pakon', and a 'Cari Kata' button is next to it. Below the search bar, a 'Cari' button is visible. Underneath, a box displays the following translations:

- Aceh : Pakon
- Indonesia : Mengapa
- English : Why
- Arab : -
- Jawoe : -

At the bottom, a footer reads 'Copyright @ Indonesia Aceh 2022'.

Gambar 18 Detail terjemahan Aceh – Indonesia

Gambar 19 adalah tampilan detail terjemahan Indonesia-Aceh

The screenshot shows a web interface for 'Kamus Indonesia Aceh'. At the top, there are navigation links: 'TRANSLATE', 'KAMUS', 'SARAN TERJEMAHAN', and 'REQUEST TERJEMAHAN'. Below this, the title 'Kamus Indonesia Aceh' is displayed. A search bar contains the word 'Mengapa', and a 'Cari Kata' button is next to it. Below the search bar, a 'Cari' button is visible. Underneath, a box displays the following translations:

- Indonesia : Mengapa
- Aceh : Pakon
- English : Why
- Arab : -
- Jawoe : -

At the bottom, a footer reads 'Copyright @ Indonesia Aceh 2022'.

Gambar 19 Detail terjemahan Indonesia – Aceh

3.10 Coding and Algorithm Implementation

Setelah menentukan kebutuhan sistem dan desain telah selesai dilakukan, maka pada tahap pengkodean ini akan melakukan implementasi pembuatan sistem berdasarkan hasil analisis kebutuhan dan desain sistem tersebut. Bahasa yang akan digunakan pada pengembangan sistem ini adalah *Python*.

3.11 Testing

Setelah sistem selesai dikembangkan, maka tahap selanjutnya adalah melakukan pengujian atau evaluasi pada sistem yang sudah selesai. Pengujian meliputi 2 hal penting terkait kinerja fitur pencarian:

3.11.1 Evaluasi Similarity

Pada evaluasi ini sistem akan menghitung kesamaan atau *similarity* kata yang dicari dengan setiap kata yang direkomendasikan oleh sistem kemudian mengembalikan hasil perhitungan dalam bentuk persentasi. Berikut adalah rumus yang digunakan untuk menghitung *similarity* antara kata yang dimasukkan dengan setiap baris rekomendasi:

$$\text{similarity} = \frac{\text{len_text} - \text{sum_diff}}{\text{len_text}} \times 100\% \quad \text{Similarity (Sumber: zenius.net)}$$

3.11.2 Evaluasi Akurasi

Pada evaluasi ini akan menguji ketepatan fitur dalam memberikan rekomendasi kata yang diinginkan pengguna. Fitur diuji dengan melakukan pencarian berulang terhadap 100 data secara acak yang memiliki penulisan yang

unik secara alpabet. Untuk menghitung akurasi dari Algoritma diperlukan dua rumus yaitu:

$$hasil = \frac{posisi_kata}{jumlah_list} \times 100\% \text{ Hasil (Sumber: id.quora.com)}$$

pada rumus di atas, **posisi_kata** diisi nilai dari **jumlah_list**. Untuk kata yang pertama kemudian dikurangi 1 untuk kata selanjutnya sampai rekomendasi terakhir.

$$akurasi = \frac{hasil_terjumlahkan}{jumlah_data_testing} \times 100\% \text{ Akurasi (Sumber: zenius.net)}$$

Pada rumus di atas semua nilai **hasil** pada rumus sebelumnya dijumlahkan untuk dijadikan nilai dari **hasil_terjumlahkan** kemudian dibagi dengan **jumlah_data_testing**.

3.12 Maintenance

Setelah sistem selesai dikembangkan tahapan terakhir adalah melakukan pemeliharaan pada sistem agar tetap bekerja sebagaimana mestinya.

3.13 Tools dalam Penelitian

Pada penelitian ini, peneliti menggunakan beberapa alat yang diperlukan untuk pengembangan sistem agar penelitian ini dapat diselesaikan dengan baik, beberapa *tools* yang akan digunakan adalah:

- a. Bahasa Pemrograman *Python*
- b. HTML dan CSS
- c. *Sqlite*
- d. *Visual Studio Code*

- e. *Git Version Control*
- f. MacBook Pro M1 2020, RAM 8 GB, SSD 512 GB

3.14 Tempat Penelitian

Penelitian ini dilaksanakan di Laboratorium Program Studi Teknologi Informasi, Universitas Islam Negeri Ar-Raniry, Banda Aceh.

3.15 Jadwal Penelitian

Penelitian akan dilakukan berdasarkan jadwal yang terlampir pada tabel 2.

Tabel 2 Waktu Penelitian

NO		KEGIATAN	JADWAL PENELITIAN 2022																											
			JULI				AGUSTUS				SEPTEMBER				OKTOBER				NOVEMBER				DESEMBER							
			I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV				
1	Identifikasi Masalah																													
2	Penyusunan dan Pengajuan Judul																													
3	Studi Literatur																													
4	Pengumpulan Data																													
5	Analisis Sistem																													
6	Perancangan Sistem																													
7	Penyusunan Proposal																													
8	Seminar Proposal																													
9	Pembuatan dan Pengujian Sistem																													
10	Penyusunan Skripsi																													
11	Sidang Skripsi																													

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil

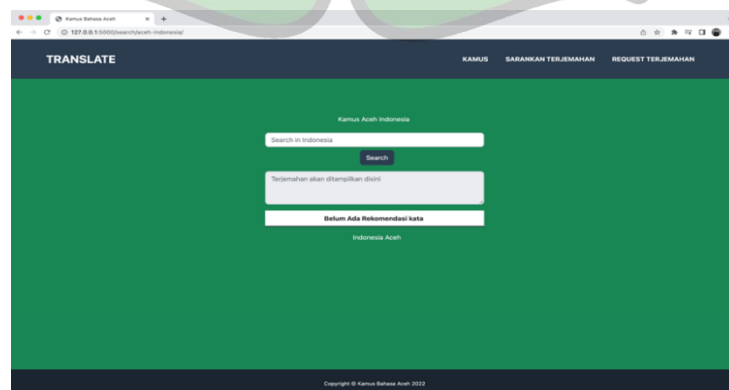
Setelah penelitian selesai dilakukan, hasil akhir akan meliputi dua bagian penting. Pertama memperbaiki tampilan halaman pencarian menjadi lebih sederhana. Kedua menghitung *similarity* dan akurasi dari kata yang dicari dengan kata yang diinginkan oleh pengguna.

4.2 Hasil Perbaikan Tampilan

Tampilan halaman yang diperbaiki terdapat pada 2 fitur pencarian aplikasi. yaitu tampilan *form* pencarian, tampilan rekomendasi kata setelah pengguna menekan tombol *Search* dan tampilan terjemahan setelah pengguna memilih salah satu kata yang direkomendasikan.

4.2.1 Halaman pencarian Aceh Indonesia

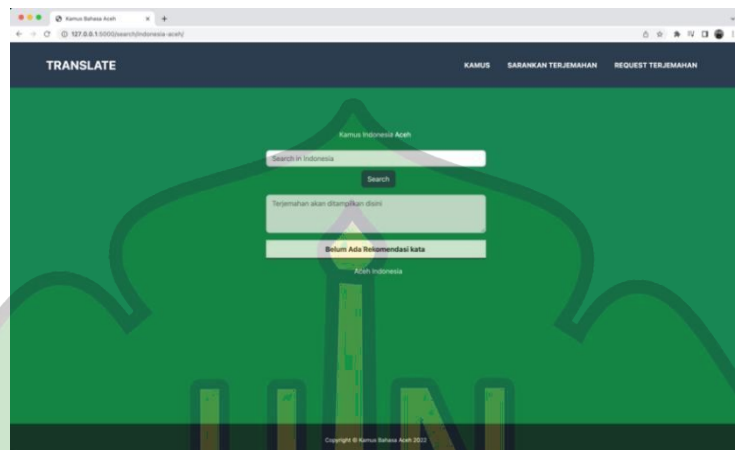
Halaman ini merupakan tampilan pertama yang akan dilihat oleh pengguna saat membuka website kamus bahasa aceh melalui mesin pencari. Gambar 20 menampilkan hasil perbaikan tampilan fitur pencarian pada aplikasi:



Gambar 20 Pencarian Aceh Indonesia

4.2.2 Halaman pencarian Indonesia Aceh

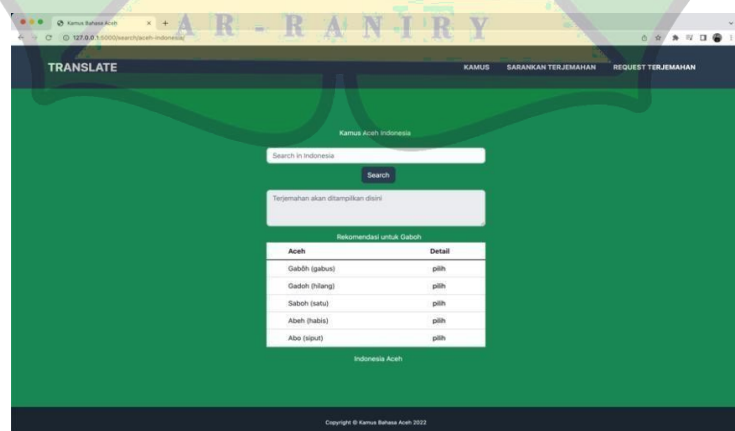
Pada halaman ini, akan muncul tampilan fitur pencarian kata bahasa Indonesia yang akan diterjemahkan ke dalam bahasa Aceh. Gambar 21 menampilkan hasil perbaikan tampilan pencarian kata Indonesia Aceh:



Gambar 21 Pencarian Indonesia Aceh

4.2.3 Halaman rekomendasi Aceh Indonesia

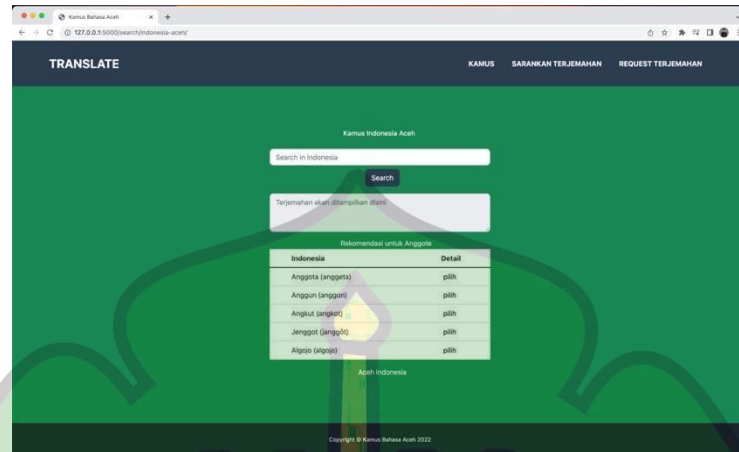
Setelah pengguna mencari kata yang diinginkan, maka website akan memberikan beberapa rekomendasi kata yang sudah dilakukan perhitungan menggunakan algoritma *Levenshtein Distance*. Gambar 22 adalah tampilan rekomendasi berdasarkan pencairan Aceh-Indonesia:



Gambar 22 Rekomendasi Aceh Indonesia

4.2.4 Halaman rekomendasi Indonesia Aceh

Gambar 23 adalah tampilan rekomendasi berdasarkan pencairan Indonesia-Aceh:



Gambar 23 Rekomendasi Indonesia Aceh

4.2.5 Halaman terjemahan Aceh Indonesia

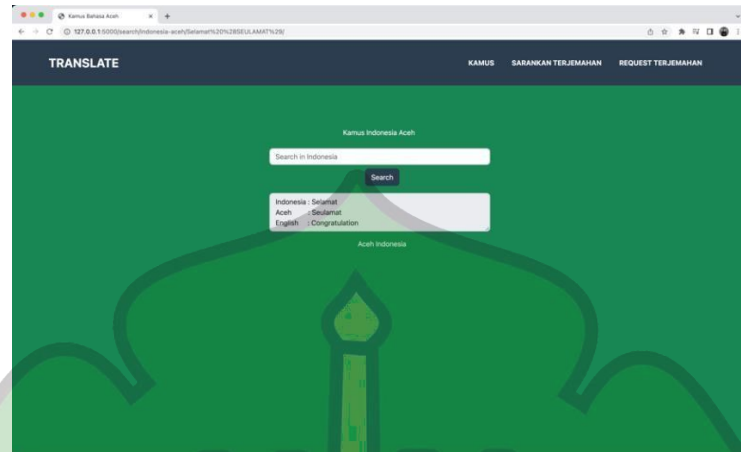
Setelah pengguna menemukan kata yang diinginkan, aplikasi akan menghilangkan rekomendasi kata yang lainya dan menuliskan kata yang dipilih ke dalam form terjemahan. Gambar 24 adalah tampilah terjemahan kata Aceh-Indonesia:



Gambar 24 Terjemahan Aceh Indonesia

4.2.6 Halaman terjemahan Indonesia Aceh

Gambar 25 sama seperti gambar 24, hanya saja terdapat perbedaan pencarian kata terjemahan antara Indonesia-Aceh:



Gambar 25 Terjemahan Indonesia Aceh

4.3 Hasil Evaluasi *Similarity*

4.3.1 Evaluasi *Similarity* Pencarian Aceh Indonesia

Ketika pengguna menekan tombol *Search* yang telah tersedia, maka sistem akan menghitung jarak setiap kata yang ada dengan kata yang dicari. Kemudian sistem akan mengembalikan 20 kata yang memiliki persentase kemiripan paling tinggi.

Hasil perhitungan persentasi kemiripan kata akan dimasukkan ke dalam *file CSV* untuk 15 kata teratas seperti terlihat pada Tabel 3.

Tabel 3 Persentasi Kemiripan dua kata bahasa Aceh

Keyword	Rekomendasi	Panjang	Persen Jarak	Karakter Beda
selamat	seulamat	7	85%	1
selamat	alamat	7	71%	2
selamat	lambat	7	71%	2
selamat	seumat	7	71%	2
selamat	kalimat	7	57%	3
selamat	ladat	7	57%	3
selamat	lalat	7	57%	3
selamat	peuaman	7	57%	3
selamat	plakat	7	57%	3
selamat	sama	7	57%	3
selamat	sapat	7	57%	3
selamat	sarat	7	57%	3
selamat	seulala	7	57%	3
selamat	seulat	7	57%	3
selamat	seumak	7	57%	3

Tabel 4 menampilkan detail operasi yang terjadi untuk 10 kata yang paling mirip seperti yang sudah dijelaskan pada Bab 2 point 11.

Tabel 4 Detail Operasi yang terjadi 1

PENCARIA	REKOMENDA	INSERTIO	DELETIO	SUBSTITUTIO	JARAK
Selamat	Seulamat	u			1
	Alamat		's	e > a	2
	Seumat		l	a > u	2
	Kalimat			s > k e > a a > l	3
	Ladat		s, e	m > d	3
	Lalat		s, e	m > l	3
	Peuaman			s > p l > u t > n	3
	Plakat		e	s > p m > k	3
	Sama		e, l, t		3

4.3.2 Evaluasi *Similarity* Pencarian Indonesia Aceh

Pada bagian ini sama seperti point 4.3.1 namun pencarian dilakukan berdasarkan bahasa Indonesia. Tabel 5 menunjukkan persentasi kemiripan antara kata yang dicari dengan yang direkomendasikan.

Tabel 5 Persentasi kemiripan dua kata bahasa Indonesia

Keyword	Rekomendasi	Panjang	Persen Jarak	Karakter Beda
kenapa	kelapa	6	83%	1
kenapa	berapa	6	66%	2
kenapa	berapa	6	66%	2
kenapa	jenaka	6	66%	2
kenapa	kecap	6	66%	2
kenapa	kepala	6	66%	2
kenapa	menara	6	66%	2
kenapa	mengapa	6	66%	2
kenapa	menyapa	6	66%	2
kenapa	tenaga	6	66%	2
kenapa	apa	6	50%	3
kenapa	apa	6	50%	3
kenapa	apa	6	50%	3
kenapa	benalu	6	50%	3
kenapa	benang	6	50%	3

Tabel 6 menampilkan detail operasi yang terjadi untuk 10 kata yang paling mirip seperti yang sudah dijelaskan pada Bab 2 point 11.

Tabel 6 Detail operasi yang terjadi 2

PENCARIAN	REKOMEND	INSERTION	DELETION	SUBSTITUTION	JARAK
Kenapa	Kelapa			n > l	1
	Berapa			k > b n > r	2
	Jenaka			k > j p > k	2
	Kecap		a	n > c	2
	Kepala		n > p p > l		2
	Menara		k > m p > r		2
	Mengapa	g		k > m	2
	Menyapa	y		k > m	2
	Tenaga			k > t g > p	2

4.3 Hasil Evaluasi Akurasi

Evaluasi akurasi Algoritma dilakukan berdasarkan pencarian yang dilakukan secara berulang untuk 100 kata yang sengaja ditulis secara acak dan

tidak sempurna. Tabel 7 dan tabel 8 adalah list data testing dan list data yang akurat.

Tabel 7 List 100 Data *Testing*

Ap	Ikle	Kapai jelajah	Rampeng	Tinggong
Abejad	Ikot	Karonya	Ranteng	Trompet
Abeuk	Ile	Lisek	Reflektor	Tuleh
Gaboh	Ipoh	Loteng	Reubong	Tumpok
Euntong	Itek	Lungkee	Reuntek ie ujeun	Turonan
Engkot	Jagong	Majelih	Rham	Ubah bentuk
Gabong	Jamprong	Mata kawé	Rhap	Udep
Gadeng	Janggot	Meuah	Sagop	Ujong
Gambe	Jareung	Meuhoi	Saket gigoo	Uke
Gamet	Jeum	Nyap	Salah tapeuse	Ulee
Gameum	Jeut peuteunte	On kayee	Sanggoi	Uree
Gamlomlam	Jehuak	Piep	Sanjung	Urong
Gampong	Jeupak	Pade	Sate	Urosan
Ganje that	Jeunerop	Pageue rinyeun	Sep that carong	Utom
Gunteng	Jeurelang	Pajoh	Serbaro	Wajeb
Guree	Jeuh	Pake	Temasok	Wajek
Han	Jram	Pre	Temipee	Wareh
Iek	Kabaret	Pancopet	Tengoh thon	Wase
Idong	Kadet	Rabon mata	Teuniree	Weng
Ie bobah	Kama eh	Rak kebak bu	Teurupong	Yah cek

Tabel 8 List 100 Data Akurat

'ap	Iklém	Kapai jeulajah	Rampéng	Tinggóng
Abeujad	Ikôt	Karónya	Ranténg	Trompèt
Abeuek	Ilé	Lisék	Réfléktor	Tuléh
Gabôh	Ipôh	Lotèng	Reubông	Tumpôk
Euntông	Iték	Lungkée	Reuntek ie ujeun	Turônán
Eungkôt	Jagông	Majeulih	Rh'am	Ubah beuntók
Gabông	Jamprông	Mata kawé	Rh'ap	Udép
Gadéng	Janggôt	Meu'ah	Sagôp	Ujông
Gambé	Jaréuceng	Meuhôi	Sakét gigoe	Uké
Gamèt	Jeuem	Ny'ap	Salah tapeusé	Ulèe
Gameuem	Jeuet peuteuntée	Ôn kayée	Sanggôi	Urée
Gamlômlam	Jeuh'ak	P'iep	Sanjông	Urông
Gampông	Jeuleupak	Padé	Saté	Urôsan
Ganjé that	Jeuneurop	Pageue rinyeuen	Sép that carong	Utôm
Gunténg	Jeureulang	Pajôh	Seubarô	Wajéb
Gurée	Jeuh	Paké	Teumasôk	Wajék
H'an	Jr'am	Pré	Teumipée	Waréh
'iek	Kabarèt	Pancopèt	Teungoh thôn	Wasé
Idông	Kadèt	Rabôn mata	Teunirée	Wéng
Ie bôbah	Kama éh	Rak keubah bu	Teurupông	Yah cék

Tabel 9 adalah list 20 dari 100 kata yang digunakan untuk menguji sistem:

Tabel 9 List 20 Data *Testing* dan Akurasi Algoritma

Keyword	Prediksi	Posisi Rekomendasi	Posisi / Jumlah	Hasil
Ap	'ap	1	10/10	1
Abejad	Abeujad	1	10/10	1
Abeuk	Abeuek	1	10/10	1
Gaboh	Gabôh	1	10/10	1
Euntong	Euntông	1	10/10	1
Engkot	Eungkôt	4	7/10	0.7
Gabong	Gabông	1	10/10	1
Gadeng	Gadéng	1	10/10	1
Gambe	Gambé	2	9/10	0.9
Gamet	Gamèt	1	10/10	1
Gameum	Gameuem	1	10/10	1
Gamlomlam	Gamlômlam	1	10/10	1
Gampong	Gampông	2	9/10	0.9
Ganje that	Ganjé that	1	10/10	1
Gunteng	Gunténg	1	10/10	1
Gurée	Gurèe	2	9/10	0.9
Han	H'an	5	6/10	0.6
Iek	'iek	1	10/10	1
Idong	Idông	3	8/10	0.8
Ie bobah	Ie bôbah	2	9/10	0.9

Berdasarkan tabel 9 dapat dilihat beberapa kolom yang memiliki nilai yang berbeda, kolom **Keyword** berisi inputan yang dimasukan, kolom **Prediksi** adalah kata yang diinginkan pengguna, kolom **Posisi Rekomendasi** adalah letak kata yang diinginkan pengguna dimulai dari 1, kolom **Posisi / Jumlah** berisi nilai posisi kata yang sesuai dimulai dari 10 untuk yang teratas 1 untuk yang paling bawah dan jumlah list rekomendasi yang ditampilkan, kolom **Hasil** merupakan hasil dari posisi dibagi jumlah list.

Setelah melakukan evaluasi akurasi terhadap 100 acak, dihasilkan bahwa Algoritma *Levenshtein Distance* memiliki akurasi sebesar 94% dalam memberikan rekomendasi kata yang tetap.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian Implementasi Algoritma *Levenshtein Distance* Untuk Fitur Pencarian Kata Pada Pengembangan Kamus Bahasa Aceh Berbasis Web. Data yang digunakan berjumlah 6387 yang terdiri dari 3 bahasa yaitu Aceh-Indonesia- Inggris untuk setiap baris kata yang tersimpan di dalam *database*. Kesimpulan dari penelitian ini antara lain adalah:

1. Data yang awalnya berjumlah 200 kata sekarang sudah menjadi 6387 kata.
2. Pengembangan website ini dikembangkan menggunakan bahasa pemrogramman Python dengan *Framework Flask* untuk kode yang berjalan disisi server.
3. *Levenshtein Distance* merupakan Algoritma yang tepat untuk menghitung jarak antar kata agar menghasilkan rekomendasi kata untuk ditampilkan ke pengguna.

5.2 Saran

Dari penelitian yang telah dilakukan, maka peneliti memberikan beberapa saran untuk bisa dijadikan penelitian selanjutnya dalam objek yang sama, antara lain:

1. Penambahan jenis bahasa dan contoh yang dapat diterjemahkan oleh sistem
2. Pengembangan pada fitur pencarian agar sistem dapat melakukan pencarian berdasarkan contoh dari suatu kata.

DAFTAR PUSTAKA

- Arsyad (2017), *Bahasa Sebagai Cermin Kebudayaan*. Fakultas Ilmu Tarbiyah dan Keguruan UIN Sumatera Utara Medan.
- Ening Herniti (2010), *Bahasa Dan Kelahirannya*, Jl. Marsda Adisucipto Yogyakarta 55281. Fakultas Adab UIN Sunan Kalijaga Yogyakarta
- Eko Widiyanto (2018), *Pemertahanan Bahasa Daerah Melalui Pembelajaran dan Kegiatan di Sekolah*. Universitas Mulia Kudus.
- Kusumah dan Cahyati (2021), *Kolerasi Orang Tua Terhadap Pemertahanan Bahasa Sunda Sebagai Bahasa Ibu Di Daerah Kuningan*. STKIP Muhammadiyah Kuningan.
- Mushaddiq (2021), *Perancangan Aplikasi Pembelajaran Bahasa Aceh Berbasis Android Untuk Guru PAUD Menggunakan Metode user Centered Design (UCD)*. Banda Aceh. Universitas Islam Negeri Ar-Raniry.
- Nuraini (2020), *Implementasi Algoritma Kruth Morris Prath Untuk Kamus Terjemah Digital Aceh - Bahasa Indonesia Berbasis Web*. Institut Bisnis dan Informatika (IBI) Kosgoro 1957.
- Riska Rahayu (2021), *Perancangan dan Pembuatan Kamus Online Atraktif Bahasa Aceh Berbasis Website*. Universitas Islam Negeri Ar-Raniry.
- Restiana (2019), *Penerapan Algoritma Levenshtein Distance Untuk Pencarian Pada Sistem Informasi Perpustakaan Pada Fakultas Kedokteran Universitas Palangka Raya*. Jl. Yos Sudarso Palangka. Universitas Palangka Raya.
- Ida bagus K.S.A (2016). *Optimasi Pencarian Kata Pada Kamus Aneka Bahasa Menggunakan Algoritma Levenshtein Distance*. Jl.Raya Puputan No. 86 Renon, Denpasar-Bali. STIKOM Bali.
- Sari dan Ummi (2020). *Perancangan Aplikasi Kamus Bahasa Minang Indonesia dan Indonesia Minang Menggunakan Algoritma Levenshtein*. Universitas Potensi Utama, K.L Yos Sudarso KM 6,5 No. 3A Tj.Mulia-Medan.
- Ayu Siregar (2021), *Aplikasi Kamus Bahasa Daerah Telukkuantan Berbasis Android*. Jl. Gatot Subroto KM. 7 Kebun Nenas, Desa Jake, Kab. Kuantan Singingi. Universitas Islam Kuantan Singingi.
- Rahmanto (2021), *Penerapan Algoritma Sequential Search pada Aplikasi Kamus Bahasa Ilmiah Tumbuhan*. Jl. ZA. Pagar Alam No.9-11, Labuhan Ratu, Bandarlampung 35142, Lampung, Indonesia. Universitas Teknokrat Indonesia.
- Sarmila Devi (2020), *Morfologi Bahasa Aceh Gampung Leuge Peurelak Dengan Gampung Keude peureulak Tahun 2021*. FKIP Universitas Samudra.

- Hifra dan Razali (2022), *Analisis Kesalahan Penulisan Bahasa Aceh Dalam Lirik Lagu Aceh*. Universitas Syiah Kuala.
- Putri Khusnul Khotimah (2021), *Penyusunan kamus Saku Anak Bergambar Tiga Bahasa (Arab-Inggris-Indonesia)*. Universitas Islam Negeri Syarif Hidayatullah.
- Tsania Khairani Aritonang (2022), *Pengenalan Algoritma Pada Pembelajaran Pemrograman Komputer*. UIN Sumatera Utara.
- Aditya Sinaga (2021), *Implementasi Algoritma Brute Force Dalam Pencarian Menu Pada Aplikasi Pemesanan Coffee (Studi Kasus : Tanamera Coffee)*. STMIK Pelita Nusantara Jl. St. Iskandar Muda No. 1 Medan, Indonesia.
- Rismayani (2021), *Rancang Bangun Pencarian Kata Pada Kamus Istilah Komputer dan Informatika Menggunakan Algoritme Brute Force Berbasis Android*. Jl. Perintis Kemerdekaan Km.9 Makassar, Makassar, Indonesia 90000. STMIK Dipanegara Makassar.
- Nababan dan Jannah (2019). *Algoritma String Matching Brute Force dan Kruth Morris Pratt Sebagai Search Engine Berbasis Web Pada Kamus Istilah Jaringan Komputer*. STMIK Pelita Nusantara, Jl. Iskandar Muda No.1, Medan, Indonesia.
- Robiatul Adawiyah (2022), *Implementasi Algoritma Levenshtein Distance Dalam Mendeteksi Plagiarisme*. Universitas Potensi Utama.
- Aulia Sabri (2022), *Analisis Perbandingan Model Arsitektur CNN Dalam Pendeteksian Coronavirus Disease Menggunakan Citra X-Ray Paru-Paru (Studi Kasus : Baseline CNN, InceptionResnetV2, VGG19, dan Xception)*. Universitas Islam Negeri Ar-Raniry.
- Shinta Permata Sari (2022), *Analisis Technology Acceptance Model terhadap Minat Penggunaan Alat Pembayaran Non Tunai melalui Pemanfaatan Google Trends*. Mataram. Universitas Muhammadiyah Mataram.
- Nur Laeli Rachmawati (2021), *Evaluasi Kepuasan Pengguna Sistem E-Learning Menggunakan Metode End User Computing Satisfaction (Studi Kasus : Universitas Amikom Purwokerto)*. Sistem Informasi Universitas Amikom Purwokerto.
- Hani Saniatun Istiqomah (2022), *Use of Website as Information Media and Learning Media for Madrasah Students in Ngawi Regency*. Universitas Islam Negeri Raden Mas Said Surakarta.

LAMPIRAN-LAMPIRAN

Lampiran 1 Code

Insert Data to Database

```
# Method to insert to database
from app.controller.kosakata import buat_kosakata_baru

file = open('kosa_kata.txt','r')
# Read data from file line by line
data = file.readlines()

for d in data:
    # Split data Aceh - Indonesia - English
    kata = d.split(' - ')
    if len(kata) < 2:
        continue
    if len(d.split(' - ')) > 3:
        print('lebih dari 3')
    if len(d.split(' - ')) < 3:
        print('kurang dari 3')

    print(f'{kata[0]}, {kata[1]}, {kata[2]}')
    buat_kosakata_baru(kata[0], kata[1], kata[2])
```

Levenshtein Codebase

```
import numpy

def levenshteinDistanceMatrix(token1, token2):
    distances = numpy.zeros((len(token1) + 1, len(token2) + 1))
    for t1 in range(len(token1) + 1):
        distances[t1][0] = t1

    for t2 in range(len(token2) + 1):
        distances[0][t2] = t2

    a, b, c = 0, 0, 0
    for t1 in range(1, len(token1) + 1):
        for t2 in range(1, len(token2) + 1):
            if (token1[t1-1] == token2[t2-1]):
                distances[t1][t2] = distances[t1 - 1][t2 - 1]
            else:
                a = distances[t1][t2 - 1]
                b = distances[t1 - 1][t2]
                c = distances[t1 - 1][t2 - 1]
                if (a <= b and a <= c):
                    distances[t1][t2] = a + 1
```

```

elif (b <= a and b <= c):
    distances[t1][t2] = b + 1
else:
    distances[t1][t2] = c + 1

```

```

return int(distances[len(token1)][len(token2)])

```

Calculate Levenshtein Distance

```

from levenshtein import levenshteinDistanceMatrix
from app.utilities import calculate_percentage_accuracy, insert_distance_percentage

```

```

def calcDictDistance(word, numWords, search_type, csv_filename=None):
    file = open('kosa_kata.txt', 'r')
    lines = file.readlines()
    file.close()

```

```

dictWordDist = []
wordIdx = 0
perbedaan = []

```

```

for line in lines:
    value = line.split(' - ')
    if len(value) != 3:
        continue
    if search_type == 'aceh_indonesia':
        wordDistance = levenshteinDistanceMatrix(word, value[0].strip())
        perbedaan.append(wordDistance)
    elif search_type == 'indonesia_aceh':
        wordDistance = levenshteinDistanceMatrix(word, value[1].strip())
        perbedaan.append(wordDistance)

```

```

if wordDistance >= 10:
    wordDistance = 9

```

```

if search_type == 'aceh_indonesia':
    dictWordDist.append(str(int(wordDistance)) + "-" + value[0].strip())
    perbedaan.append(wordDistance)

```

```

elif search_type == 'indonesia_aceh':
    dictWordDist.append(str(int(wordDistance)) + "-" + value[1].strip())
    perbedaan.append(wordDistance)

```

```

wordIdx += 1

```

```

closestWords = []
wordDetails = []
currWordDist = 0

```

```

dictWordDist.sort()
alldata = []
total_data = 0
for i in range(numWords):
    currWordDist = dictWordDist[i]
    wordDetails = currWordDist.split("-")
    percentage_accuracy = calculate_percentage_accuracy(int(wordDetails[0]), len(word))
    data = [word, wordDetails[1], len(word), f'{str(int(percentage_accuracy))}%', wordDetails[0]]
    total_data += 1
    alldata.append(data)
    closestWords.append(wordDetails[1])

if csv_filename is not None:
    insert_distance_percentage(alldata, csv_filename)
perbedaan.sort()
return [closestWords]

```

Calculate Acuration Codebase

```

def calculate_percentage_accuracy(distance_character, total_character):
    return abs((distance_character - total_character) / total_character) * 100

def hitung_akurasi_levenshtein_distance(akurasi, numWords):
    return (akurasi/numWords) * 100

```

Data Testing untuk 100 kali pencarian

ap - 'ap
 abejad - abeujad
 abeuk - abeuek
 gaboh - gabôh
 euntong - euntông
 engkot - eungkôt
 gabong - gabông
 gadeng - gadéng
 gambe - gambé
 gamet - gamèt
 gameum - gameuem
 gamlomlam - gamlômlam
 gampong - gampông
 ganje that - ganjé that
 gunteng - gunténg
 guree - gurèe
 han - h'an
 iek - 'iek
 idong - idông
 ie bobah - ie bôbah
 iklem - iklém
 ikot - ikôt
 ile - ilé

ipoh - ipôh
 itek - iték
 jagong - jagông
 jamprong - jamprông
 janggot - janggôt
 jareung - jaréueng
 jeum - jeuem
 jeut peuteunte - jeuet peuteuntèe
 jehuak - jeuh'ak
 jeleupak - jeuleupak
 jeunerop - jeuneurop
 jeurelang - jeureulang
 jeuh - jeuôh
 jram - jr'am
 kabaret - kabarèt
 kadet - kadèt
 kama eh - kama éh
 kapai jelajah - kapai jeulajah
 karonya - karônnya
 lisek - lisék
 loteng - lotèng
 lungkee - lungkèe
 majelih - majeulih
 mata kawé - mata kawé
 meuah - meu'ah
 meuhoi - meuhôi
 nyap - ny'ap
 on kayee - ôn kayèe
 piep - p'iep
 pade - padé
 pageue rinyeun - pageue rinyeuen
 pajoh - pajôh
 pake - paké
 pre - pré
 pancopet - pancopèt
 rabon mata - rabôn mata
 rak kebak bu - rak keubah bu
 rampeng - rampéng
 ranteng - ranténg
 reflektor - réflèktor
 reubong - reubông
 reuntek ie ujeun - reunték ie ujeuen
 rham - rh'am
 rhap - rh'ap
 sagop - sagôp
 saket gigoo - sakét gigoe
 salah tapeuse - salah tapeusé
 sanggoi - sanggôi



sanjung - sanjông
 sate - saté
 sep that carong - sép that carong
 serbaro - seubarô
 temasok - teumasôk
 temipée - teumipèe
 tengoh thon - teungoh thôn
 teuniree - teunirée
 teurupong - teurupông
 tinggong - tinggông
 trompet - trompèt
 tuleh - tuléh
 tumpok - tumpôk
 turonan - turônân
 ubah bentuk - ubah beuntôk
 udep - udép
 ujong - ujông
 uke - uké
 ulee - ulèe
 uree - urée
 urong - urông
 urosan - urôsan
 utom - utôm
 wajib - wajéb
 wajak - wajék
 wareh - waréh
 wase - wasé
 weng - wéng
 yah cek - yah cék

Hasil Testing 100 Data dan Akurasi Algoritma

Keyword	Prediksi	Posisi Rekomendasi	Posisi / Jumlah	Hasil
Ap	'ap	1	10/10	1
Abejad	Abeujad	1	10/10	1
Abeuk	Abeuek	1	10/10	1
Gaboh	Gabôh	1	10/10	1
Euntong	Euntông	1	10/10	1
Engkot	Eungkôt	4	7/10	0.7
Gabong	Gabông	1	10/10	1
Gadeng	Gadéng	1	10/10	1
Gambe	Gambé	2	9/10	0.9

Gamet	Gamèt	1	10/10	1
Gameum	Gameuem	1	10/10	1
Gamlomlam	Gamlômlam	1	10/10	1
Gampong	Gampông	2	9/10	0.9
Ganje that	Ganjé that	1	10/10	1
Gunteng	Gunténg	1	10/10	1
Guree	Gurée	2	9/10	0.9
Han	H'an	5	6/10	0.6
Iek	'iek	1	10/10	1
Idong	Idông	3	8/10	0.8
Ie bobah	Ie bôbah	2	9/10	0.9
Ikem	Iklém	1	10/10	1
Ikot	Ikôt	2	9/10	0.9
Ile	Ilé	2	9/10	0.9
Ipoh	Ipôh	1	10/10	1
Itek	Iték	2	9/10	0.9
Jagong	Jagông	2	8/10	0.8
Jamprong	Jamprông	1	10/10	1
Janggot	Janggôt	1	10/10	1
Jareung	Jaréueng	5	6/10	0.6
Jeum	Jeuem	2	9/10	0.9
Jeut peuteunte	Jeuet peuteuntée	1	10/10	1
Jehuak	Jeuh'ak	1	10/10	1
Jeuleupak	Jeuleupak	1	10/10	1
Jeunerop	Jeuneurop	1	10/10	1
Jeurelang	Jeureulang	1	10/10	1
Jeuoh	Jeuôh	1	10/10	1
Jram	Jr'am	1	10/10	1
Kabaret	Kabarèt	1	10/10	1
Kadet	Kadèt	1	10/10	1
Kama eh	Kama éh	1	10/10	1
Kapai jelajah	Kapai jeulajah	1	10/10	1
Karonya	Karônya	1	10/10	1
Lisek	Lisék	1	9/10	0.9

Loteng	Lotèng	1	10/10	1
Lungkee	Lungkèe	1	10/10	1
Majelih	Majeulih	1	10/10	1
Mata kawé	Mata kawé	1	10/10	1
Meuah	Meu'ah	1	10/10	1
Meuhoi	Meuhôi	2	9/10	0.9
Nyap	Ny'ap	1	10/10	1
On kayee	Ôn kayèe	2	9/10	0.9
Piep	P'iep	1	10/10	1
Pade	Padé	2	9/10	0.9
Pageue rinyeun	Pageue rinyeuen	1	10/10	1
Pajoh	Pajôh	1	10/10	1
Pake	Paké	3	8/10	0.8
Pre	Pré	1	10/10	1
Pancopet	Pancopèt	1	10/10	1
Rabon mata	Rabôn mata	1	10/10	1
Rak kebak bu	Rak keubah bu	1	10/10	1
Rampeng	Rampéng	1	10/10	1
Ranteng	Ranténg	1	9/10	0.9
Reflektor	Réflèktor	1	10/10	1
Reubong	Reubông	1	10/10	1
Reuntek ie ujeun	Reunték ie ujeuen	1	10/10	1
Rham	Rh'am	3	8/10	0.8
Rhap	Rh'ap	2	9/10	0.9
Sagop	Sagôp	2	9/10	0.9
Saket gigoo	Sakét gigoe	1	10/10	1
Salah tapeuse	Salah tapeusé	1	10/10	1
Sanggoi	Sanggôi	1	10/10	1
Sanjung	Sanjông	1	10/10	1
Sate	Saté	4	7/10	0.7
Sep that carong	Sép that carong	1	10/10	1
Serbaro	Seubarô	1	10/10	1
Temasok	Teumasôk	1	10/10	1
Temipee	Teumipée	1	10/10	1

Tengoh thon	Teungoh thôn	1	10/10	1
Teuniree	Teunirée	1	10/10	1
Teurupong	Teurupông	1	10/10	1
Tinggong	Tinggông	1	10/10	1
Trompet	Trompèt	1	10/10	1
Tuleh	Tuléh	1	10/10	1
Tumpok	Tumpôk	2	9/10	0.9
Turonan	Turônán	1	10/10	1
Ubah bentuk	Ubah beuntôk	1	10/10	1
Udep	Udép	1	10/10	1
Ujong	Ujông	1	10/10	1
Uke	Uké	2	9/10	0.9
Ulee	Ulèe	4	7/10	0.7
Uree	Urèe	3	8/10	0.8
Urong	Urông	3	8/10	0.8
Urosan	Urôsan	1	10/10	1
Utom	Utôm	3	8/10	0.8
Wajeb	Wajéb	1	10/10	1
Wajek	Wajék	1	10/10	1
Wareh	Waréh	3	8/10	0.8
Wase	Wasé	2	9/10	0.9
Weng	Wéng	2	9/10	0.9
Yah cek	Yah cék	2	9/10	0.9
Hasil		Jumlah semua value kolom hasil		94.7
Akurasi		(hasil / jumlah pencarian) * 100 %		94.7%