

**IMPLEMENTASI METODE *RULE- BASED* PADA PROSES  
SILABIFIKASI DALAM BAHASA ACEH**

**TUGAS AKHIR**

**Diajukan oleh:**

**BAQIATUSH SHALIHAT**

**NIM. 190705011**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**



**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI AR-RANIRY  
BANDA ACEH  
2023 M/1444 H**

## LEMBAR PERSETUJUAN

### IMPLEMENTASI METODE *RULE-BASED* PADA PROSES SILABIFIKASI DALAM BAHASA ACEH

#### TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi  
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh  
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana  
pada Prodi Teknologi Informasi

Oleh:

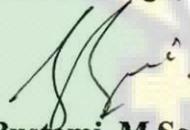
**BAQIATUSH SHALIHAT**

**NIM. 190705011**

**Mahasiswa Fakultas Sains dan Teknologi  
Program Studi Teknologi Informasi**

Disetujui untuk Dimunaqasyahkan Oleh:

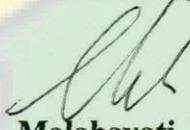
Pembimbing I,



**Bustami, M.Sc**

**NIP. 198604082014031001**

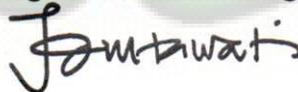
Pembimbing II,



**Malahayati, M.T**

**NIP. 198301272015032003**

Mengetahui,  
Ketua Program Studi Teknologi Informasi



**Ima Dwitawati, MBA**

**NIP. 198210132014032002**

## LEMBAR PENGESAHAN

### IMPLEMENTASI METODE *RULE-BASED* PADA PROSES SILABIFIKASI DALAM BAHASA ACEH

#### TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasah Tugas Akhir  
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus  
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)  
Pada Prodi Teknologi Informasi

Pada Hari/Tanggal: Selasa, 19 Desember 2023  
6 Jumadil Akhir 1445 H

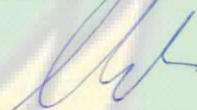
di Darussalam, Banda Aceh  
Panitia Ujian Munaqasah Tugas Akhir

Ketua,



Bustami, M.Sc  
NIP. 198604082014031001

Sekretaris,



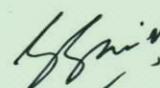
Malahayati, M.T  
NIP. 198301272015032003

Penguji I,



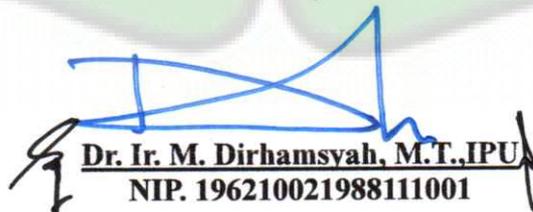
Hendri Ahmadian, S.Si., M.I.M  
NIP. 198301042014031002

Penguji II,



Khairan AR, M.Kom  
NIP. 198607042014031001

Mengetahui,  
Dekan Fakultas Sains dan Teknologi  
UIN Ar-Raniry Banda Aceh



Dr. Ir. M. Dirhamsyah, M.T., IPU  
NIP. 196210021988111001

## LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Baqiatush Shalihat

NIM : 190705011

Program Studi : Teknologi Informasi

Fakultas : Sains dan Teknologi

Judul Tugas Akhir : Implementasi Metode *Rule-Based* pada Proses Silabifikasi dalam Bahasa Aceh

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu mempertanggungjawab atas karya ini;

Bila kemudian hari ini ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat mempertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenakan sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 19 Desember 2023  
Yang Menyatakan



(Baqiatush Shalihat)

## ABSTRAK

Nama : Baqiatush Shalihat  
NIM : 190705011  
Program Studi : Teknologi Informasi  
Judul : Implementasi Metode *Rule-Based* pada Proses Silabifikasi dalam Bahasa Aceh  
Tanggal Sidang : 2023  
Jumlah Halaman : 68 Halaman  
Pembimbing I : Bustami, M.Sc  
Pembimbing II : Malahayati, M.T

Dalam era globalisasi dan interkoneksi antarindividu, komunikasi menjadi unsur penting dalam setiap interaksi manusia. Bahasa yang merupakan alat komunikasi utama juga berperan penting dalam membentuk identitas budaya setiap individu. Di Indonesia, terdapat berbagai bahasa daerah yang berkembang di samping Bahasa Indonesia sebagai bahasa nasional. Bahasa Aceh adalah bahasa yang digunakan sebagai sarana berkomunikasi dalam kehidupan sehari-hari bagi masyarakat Aceh. Namun, penggunaan Bahasa Aceh mengalami penurunan signifikan, terutama di kalangan generasi muda. Penurunan ini dipengaruhi oleh ketidakmampuan dalam berkomunikasi menggunakan Bahasa Aceh. Salah satu faktor yang berkontribusi terhadap ketidakmampuan ini adalah kurangnya pemahaman tentang tata cara membaca kata dalam Bahasa Aceh. Oleh karena itu, upaya yang dapat dilakukan untuk meningkatkan pemahaman tentang pengucapan kata berdasarkan tata cara membaca kata dalam Bahasa Aceh adalah dengan menerapkan metode *Rule-Based* dalam proses silabifikasi.

Penelitian ini menggunakan metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh. Dataset penelitian ini berisi kumpulan kata Bahasa Aceh yang berjumlah 6.225 baris data. Hasil penelitian menunjukkan bahwa implementasi metode *Rule-Based* dalam silabifikasi Bahasa Aceh berhasil dilakukan, dan model yang dihasilkan memperoleh akurasi yang sangat baik dalam memprediksi suku kata dengan tingkat WER sebesar 0,9%. Hasil ini menegaskan bahwa metode *Rule-Based* efektif dalam silabifikasi Bahasa Aceh dan memiliki potensi besar dalam meningkatkan pemahaman dan kemampuan berkomunikasi dalam Bahasa Aceh, serta berkontribusi pada pelestarian bahasa Daerah Aceh.

**Kata Kunci:** Silabifikasi, *Rule-Based*, Bahasa Aceh

## ABSTRACT

Name : Baqiatush Shalihah  
Student Number : 190705011  
Department : Information Technology  
Title : Implementation of the Rule-Based Method in the Syllabification Process in Acehese  
Date : 2023  
Number of Pages : 68 Pages  
Supervisor I : Bustami, M.Sc  
Supervisor II : Malahayati, M.T

In the era of globalization and interconnection between individuals, communication has become an important element in every human interaction. Language, which is the main means of communication, also plays an important role in forming the cultural identity of each individual. In Indonesia, there are various regional languages that have developed besides Indonesian as the national language. Acehese is a language used as a means of communication in everyday life for the people of Aceh. However, the use of Acehese has decreased significantly, especially among the younger generation. This decline was influenced by the inability to communicate using Acehese. One factor that contributes to this inability is a lack of understanding of how to read words in Acehese. Therefore, efforts that can be made to improve understanding of word pronunciation based on the procedures for reading words in Acehese are by applying the Rule-Based method in the syllabification process.

This research uses the Rule-Based method in the syllabification process in Acehese. This research dataset contains a collection of Acehese words totaling 6.225 lines of data. The research results show that the implementation of the Rule-Based method in Acehese syllabification was successful, and the resulting model obtained very good accuracy in predicting syllables with a WER level of 0,9%. These results confirm that the Rule-Based method is effective in syllabifying the Acehese language and has great potential in improving understanding and communication skills in the Acehese language, as well as contributing to the preservation of the Acehese regional language.

**Keywords: Syllabification, Rule-Based, Acehese**

## KATA PENGANTAR

*Bismillahirrahmanirrahim.*

Puji syukur penulis panjatkan atas kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul “**Implementasi Metode Rule-Based pada Proses Silabifikasi dalam Bahasa Aceh**”. Shalawat beserta salam semoga tersampaikan kepada Rasulullah SAW beserta keluarga dan sahabat beliau sekalian yang telah memperjuangkan umat Islam kepada jalan kebenaran dengan dibekali ilmu yang bermanfaat untuk dunia dan akhirat.

Penyusunan tugas akhir ini diajukan sebagai salah satu syarat untuk menyelesaikan tugas akhir pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, UIN Ar-Raniry. Dalam penulisan tugas akhir ini, penulis dengan segala kerendahan hati ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ibunda Mamni dan Ayahanda Shalahuddin serta abangnda tersayang Al-Hadid Qursani yang selalu memberikan doa dan dukungan yang tiada hentinya.
2. Bapak Bustami, M.Sc selaku Pembimbing I yang senantiasa memberikan arahan dan bimbingan dalam penyusunan tugas akhir ini.
3. Ibu Malahayati, M.T. selaku Pembimbing II dan sebagai Penasehat Akademik yang senantiasa memberikan arahan dan motivasi.
4. Ibu Ima Dwitawati, MBA dan Bapak Khairan AR, M.Kom selaku Ketua dan Sekretaris Prodi Teknologi Informasi, yang telah memberikan bimbingan dan arahan dalam menyelesaikan tugas akhir ini.
5. Ibu Cut Ida Rahmadiana, S.Si selaku *Staff* Prodi Teknologi Informasi, yang senantiasa membantu penulis dalam pemberkasan administrasi.
6. Bapak Dr. Ir. M. Dirhamsyah, M.T.,IPU selaku Dekan Fakultas Sains dan Teknologi UIN Ar-Raniry.
7. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah membekali penulis dengan ilmu pengetahuan di bidang Teknologi Informasi.

8. Sahabat dan teman-teman yang selalu memberikan dukungan moral dalam menyelesaikan penyusunan tugas akhir ini.
9. Pihak-pihak terkait yang lainnya yang membantu penulis dalam penyusunan tugas akhir ini.

Penulis menyadari sepenuhnya bahwa dalam penyusunan tugas akhir ini tidak cukup dikategorikan sempurna, untuk itu penulis dengan segala kerendahan hati menerima saran dan kritikan guna menyempurnakan penyusunan tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat bagi penulis dan pembaca dan semoga dicatat sebagai sebuah amal kebaikan oleh Allah SWT.

Amiin Ya Rabbal A'lamin.

The logo of UIN Ar-Raniry is a large, stylized emblem. It features a central shield with a white background, containing a dark grey silhouette of a mosque dome and minaret. Above the shield, the letters 'UIN' are written in a large, yellow, serif font. Below the shield, there is a yellow banner with the Arabic text 'جامعة الرانيري' (University of Ar-Raniry) and the name 'AR-RANIRY' in a yellow, sans-serif font. The entire emblem is set against a green, leaf-like background.

Banda Aceh, 27 September 2023

Penulis

Baqiatush Shalihat

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN .....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN .....</b>	<b>ii</b>
<b>LEMBAR PERNYATAAN KEASLIAN .....</b>	<b>iii</b>
<b>ABSTRAK .....</b>	<b>iv</b>
<b>ABSTRACT .....</b>	<b>v</b>
<b>KATA PENGANTAR.....</b>	<b>vi</b>
<b>DAFTAR ISI.....</b>	<b>viii</b>
<b>DAFTAR GAMBAR.....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian .....	2
1.4 Batasan Masalah .....	3
1.5 Manfaat Penelitian .....	3
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>4</b>
2.1 Tinjauan Pustaka.....	4
2.2 Teori Pendukung.....	7
2.2.1 Bahasa Aceh .....	7
2.2.2 Silabifikasi.....	8
2.2.3 <i>Artificial Intelligence</i> (AI).....	9
2.2.4 <i>Machine Learning</i> .....	10
2.2.5 Algoritma.....	11
2.2.6 <i>Finite State Machine</i> (FSM).....	12
2.2.7 <i>Rule-Based</i> .....	12
2.3 <i>Tools</i> .....	13
2.3.1 <i>Python</i> .....	13
2.3.2 <i>Google Colaboratory</i> .....	13
2.4 Model Evaluasi .....	14
2.5 Kerangka Pemikiran Penelitian .....	14

<b>BAB III</b>	<b>METODOLOGI PENELITIAN .....</b>	<b>16</b>
3.1	Tahapan Penelitian .....	16
3.2	Metode Pengumpulan Data.....	16
3.2.1	Identifikasi Masalah .....	17
3.2.2	Studi Literatur.....	17
3.3	Rancangan Model .....	17
3.3.1	Penentuan Konsep Penelitian .....	17
3.3.2	Pengumpulan <i>Dataset</i> .....	17
3.3.3	Pembangunan Model.....	18
3.4	Metode Analisis Hasil.....	33
3.4.1	Evaluasi Model.....	33
3.5	Alat Bantu Penelitian.....	33
<b>BAB IV</b>	<b>HASIL DAN PEMBAHASAN.....</b>	<b>34</b>
4.1	Pelatihan Model .....	34
4.2	Hasil Pengujian Model.....	35
4.3	Evaluasi Hasil Pengujian .....	36
<b>BAB V</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>39</b>
5.1	Kesimpulan .....	39
5.2	Saran .....	39
<b>DAFTAR PUSTAKA.....</b>		<b>40</b>
<b>LAMPIRAN.....</b>		<b>42</b>
<b>RIWAYAT HIDUP.....</b>		<b>55</b>

## DAFTAR GAMBAR

Gambar 2.1 <i>Artificial Intelligence, Machine Learning, dan Deep Learning</i> .....	10
Gambar 2.2 <i>Supervised Learning</i> .....	11
Gambar 2.3 <i>Unsupervised Learning</i> .....	11
Gambar 2.4 Gambaran Aturan <i>Rule-Based</i> .....	13
Gambar 2.5 Kerangka Pemikiran Penelitian .....	14
Gambar 3.1 Tahapan Penelitian .....	16
Gambar 3.2 Alur Kerja Model .....	18
Gambar 3.3 <i>Pseudocode</i> Algoritma Silabifikasi Vokal .....	22
Gambar 3.4 <i>Pseudocode</i> Algoritma Silabifikasi Konsonan .....	24
Gambar 3.5 <i>Pseudocode</i> Algoritma Silabifikasi Tambahan Vokal .....	26
Gambar 3.6 <i>Pseudocode</i> Algoritma Silabifikasi Tambahan Konsonan .....	29
Gambar 3.7 <i>Pseudocode</i> Algoritma Silabifikasi Tambahan Khusus .....	31
Gambar 4.1 Grafik Hasil Akurasi .....	37

## DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu.....	6
Tabel 2.2 Bahasa Arab – Bahasa Aceh.....	7
Tabel 2.3 Bahasa Inggris – Bahasa Aceh.....	7
Tabel 2.4 Bahasa Portugis - Bahasa Aceh.....	7
Tabel 2.5 Bahasa Tagalog - Bahasa Aceh.....	8
Tabel 2.6 Bahasa Champa - Bahasa Aceh.....	8
Tabel 2.7 Contoh Kata Bahasa Aceh.....	8
Tabel 2.8 Contoh Silabifikasi Kata dalam Bahasa Aceh.....	9
Tabel 3.1 Huruf Vokal.....	19
Tabel 3.2 Huruf Diftong atau Gabungan Vokal.....	19
Tabel 3.3 Huruf Konsonan.....	20
Tabel 3.4 Huruf Gabungan Konsonan.....	20
Tabel 3.5 Contoh Kata Aturan Silabifikasi.....	31
Tabel 4.1 Sampel Hasil Data <i>Train</i> .....	34
Tabel 4.2 Sampel Hasil Data <i>Test</i> .....	35
Tabel 4.3 Hasil Data <i>Test</i> yang Salah disilabifikasi.....	35

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam interaksi, komunikasi menjadi suatu hal yang sangat penting. Komunikasi antar satu individu dengan individu lainnya dapat dilakukan dengan adanya bahasa (Martono, 2011). Namun, bahasa tidak hanya sebagai sarana berkomunikasi tetapi juga sebagai pengenalan identitas individu (Santoso, 2006). Setiap wilayah dalam Negara Indonesia memiliki bahasa daerahnya masing-masing terlepas dari Bahasa Indonesia sebagai bahasa persatuan.

Mempelajari bahasa daerah merupakan suatu kewajiban bagi masyarakat pada suatu daerah yang ditempati demi melestarikan bahasa daerahnya. Selain bahasa daerah memiliki peranan penting sebagai pengenalan identitas individu, bahasa daerah juga berfungsi sebagai alat komunikasi dalam kehidupan sehari-hari dan menjadi bukti peninggalan nenek moyang terdahulu (Nacikit, 2020). Salah satu bahasa daerah yang ada dalam Negara Indonesia adalah Bahasa Aceh.

Bahasa Aceh adalah bahasa yang digunakan dalam kehidupan sehari-hari sebagai alat komunikasi bagi masyarakat Aceh (Subhayni et al., 2020). Bahasa Aceh juga merupakan bahasa pertama yang diajarkan kepada masyarakat Aceh sebagai bahasa ibu (Rahma, 2020). Namun, pemakaian Bahasa Aceh mengalami penurunan setiap tahunnya. Penurunan pemakaian Bahasa Aceh dapat disebabkan oleh para generasi muda yang kurang berpartisipasi memakai Bahasa Aceh sebagai bahasa sehari-hari (Rahma, 2020). Salah satu penyebabnya adalah karena ketidakmampuan berkomunikasi menggunakan Bahasa Aceh (Alamsyah et al., 2016). Untuk mempelajari suatu bahasa khususnya Bahasa Aceh, dapat dimulai dengan mengenal huruf abjad dalam Bahasa Aceh dan mempelajari cara membaca berdasarkan suku kata.

Peningkatan kemampuan dalam berbicara dapat dipengaruhi oleh penerapan cara membaca berdasarkan suku kata (Suyadi & Sari, 2021). Suku kata atau

silabifikasi adalah pembagian atau pemisahan sebuah kata menjadi suku kata sehingga terdiri dari penggalan huruf vokal dan konsonan dengan menerapkan aturan dan kaidah tertentu (Bartlett et al., 2008). Penerapan proses pemenggalan suku kata dapat memudahkan dalam memahami cara membaca suatu kata pada bahasa yang digunakan (Rahman et al., 2021). Proses pemenggalan suku kata dapat dilakukan dengan menerapkan metode *Rule-Based*. Hal tersebut dibuktikan dengan adanya penelitian terdahulu terkait penggunaan metode *Rule-Based* pada proses pemenggalan suku kata.

Penelitian terdahulu yang menerapkan metode *Rule-Based* pada proses silabifikasi telah dilakukan oleh (Putra et al., 2019), yang menghasilkan 244 kata yang benar dari 257 jumlah kata. Penelitian berikutnya juga dilakukan oleh (Rahman et al., 2021), yang menghasilkan 39.283 kata yang benar dari 40.456 jumlah kata. Pada penelitian yang lain, dilakukan juga oleh (Mahastama, 2022) yang menghasilkan 483 kata yang benar dari 500 jumlah kata.

Oleh sebab itu, penulis berinisiatif untuk melakukan sebuah penelitian dengan menggunakan metode *Rule-Based* pada proses silabifikasi, sehingga topik dari penelitian ini adalah "Implementasi Metode *Rule-Based* pada Proses Silabifikasi dalam Bahasa Aceh". Dari hasil penelitian ini diharapkan dapat memahami cara membaca suatu kata pada Bahasa Aceh sehingga pemahaman terkait Bahasa Aceh semakin meningkat.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah yang akan dibahas pada penelitian ini adalah:

1. Bagaimana proses implementasi metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh?
2. Bagaimana tingkat *Word Error Rate* yang diperoleh dengan menggunakan metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh?

## **1.3 Tujuan Penelitian**

Berdasarkan rumusan masalah yang telah diuraikan, tujuan dari penelitian ini adalah untuk mengetahui:

1. Proses implementasi metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh.
2. Tingkat *Word Error Rate* yang diperoleh dengan menggunakan metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh.

#### 1.4 Batasan Masalah

Berdasarkan uraian tujuan pada penelitian ini, dibutuhkan batasan masalah agar tidak menyimpang dari pembahasan yang dibahas. Sehingga batasan masalah dari penelitian ini sebagai berikut:

1. Data yang digunakan adalah data yang berisi kumpulan kata dari Bahasa Aceh.
2. Teknik yang digunakan adalah proses silabifikasi atau pemenggalan kata menjadi suku kata.
3. Metode yang digunakan dalam penelitian ini adalah metode *Rule-Based* dengan menerapkan aturan silabifikasi dalam Bahasa Aceh.
4. Bahasa pemrograman yang digunakan adalah *python* dengan *google colab* sebagai media untuk editor teks.

#### 1.5 Manfaat Penelitian

Berdasarkan tujuan pada penelitian ini, maka manfaat yang didapatkan dari penelitian ini sebagai berikut:

1. Dapat menambah ilmu pengetahuan dan juga sebagai bahan evaluasi bagi peneliti selanjutnya pada penelitian yang berhubungan dengan implementasi menggunakan metode *Rule-Based* pada proses silabifikasi.
2. Dapat dijadikan sebagai referensi untuk mengetahui tingkat akurasi dengan menggunakan metode *Rule-Based* pada proses silabifikasi.
3. Dapat menambah ilmu pengetahuan bagi peneliti yang mengembangkan model *text to speech*, sehingga menjadikan model dapat berjalan dengan baik dan benar.
4. Keberhasilan dari model ini juga dapat dimanfaatkan untuk proses pembelajaran Bahasa Aceh sehingga penuturan dan pemahaman terkait Bahasa Aceh semakin meningkat.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka

Pada tahap penelitian, dibutuhkan penelitian terdahulu atau penelitian yang terkait untuk dijadikan referensi dalam pengembangan penelitian yang dilakukan. Adapun penelitian yang terkait yang dijadikan sebagai referensi yaitu:

Pertama, pada penelitian yang dilakukan oleh (Rahman et al., 2021), dengan judul "Silabifikasi Menggunakan Metode *Rule-Based* dalam Bahasa Indonesia". Penelitian ini menggunakan metode *Rule-Based* untuk melakukan silabifikasi pada kata bahasa Indonesia. Aturan yang digunakan dalam penelitian ini didapat dari proses pembelajaran *dataset*. Hasil pengujian yang didapatkan dari 40.456 jumlah kata, terdapat 1.173 kata yang salah disilabifikasi.

Kedua, pada penelitian (Humami, 2022), dengan judul "Aplikasi Penentuan Kedudukan Kata Bahasa Arab Menggunakan Metode *Rule-Based*". Pada penelitian ini menerapkan metode *Rule-Based* dalam menentukan identitas kata Bahasa Arab yang sesuai dengan aturan dan kaidah tertentu. Hasil pengujian dari 92 jumlah kata, terdapat 3 kata yang tidak sesuai pada uji coba jenis kata, 1 kata yang tidak sesuai pada uji coba jenis *i'rab*, 6 kata yang tidak sesuai pada uji coba kedudukan kata.

Ketiga, pada penelitian yang dilakukan oleh (Mahastama, 2022), dengan judul "Model Berbasis Aturan untuk Transliterasi Bahasa Jawa dengan Aksara Latin ke Aksara Jawa". Penelitian ini menggunakan model *Rule-Based* dalam proses pemisahan kata pada kata bahasa Jawa. Hasil pengujian untuk tahap pertama yang didapatkan dari 600 jumlah kata yang diuji, terdapat 17 kata yang salah. Sedangkan hasil pengujian untuk tahap kedua yang didapatkan dari 40 kalimat dengan 147 jumlah kata, terdapat 12 kalimat yang salah dalam proses pemisahan kata.

Keempat, pada penelitian (Mahastama, 2022), dengan judul "Aplikasi Bantu Belajar Menulis Aksara Jawa untuk Siswa Sekolah Dasar". Penelitian ini

menerapkan metode berbasis aturan (*Rule-Based syllabification*) dalam menentukan pemenggalan suku kata pada aksara jawa. Hasil pengujian yang didapatkan dari 500 jumlah kata, terdapat 17 kata yang salah.

Kelima, pada penelitian yang dilakukan oleh (Putra et al., 2019), dengan judul "*Syllabification of Balinese Words Using the Syllabification Algorithm*". Penelitian ini menggunakan metode pendekatan *Rule-Based* untuk proses pemisahan kata pada kata bahasa bali sesuai dengan kaidah-kaidah tertentu. Data yang digunakan berupa data yang sudah diberikan pelabelan serta terdapat data *testing* dan data yang menyimpan hasil pemisahan kumpulan kata tersebut. Hasil pengujian yang didapatkan dari 257 jumlah kata yang diuji, terdapat 13 kata yang salah dalam proses pemisahan kata pada kata bahasa bali.

Keenam, pada penelitian (Wibowo et al., 2021), dengan judul "*Implementasi Syllabification Algorithm dan String Matching Algorithm untuk Deteksi Kesalahan Pemisahan Suku Kata Pada E-Paper Artikel Berita*". Pada penelitian ini menerapkan metode *syllabification* dan *string matching* dalam melakukan silabifikasi pada kata bahasa indonesia dengan menggunakan data dalam bentuk dokumen digital. Hasil pengujian yang didapatkan dari 2.084 jumlah kata, terdapat 434 kata yang salah dalam proses silabifikasi.

Ketujuh, pada penelitian yang dilakukan oleh (Arifin et al., 2019), dengan judul "*Segmentasi Suku Kata Bahasa Indonesia Berbasis Aturan untuk Pengembangan Sistem Text-to-Audiovisual*". Penelitian ini menggunakan metode *Finite State Machine* (FSM) untuk melakukan segmentasi suku kata pada kata Bahasa Indonesia. Hasil pengujian dari penelitian ini, terdapat beberapa masalah dalam proses segmentasi kata seperti kata yang menyerupai kata diftong sehingga segmentasi kata yang didapat berupa kata diftong.

Gambaran singkat tentang metode yang digunakan dalam penelitian sebelumnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Penelitian Terdahulu

NO	PENELITI	METODE	JUDUL	HASIL
1	(Rahman et al., 2021)	<i>Rule-Based</i>	Silabifikasi Menggunakan Metode <i>Rule-Based</i> dalam Bahasa Indonesia	Dari 40.456 jumlah kata, 2,9% kata yang salah
2	(Humami, 2022)	<i>Rule-Based</i>	Aplikasi Penentuan Kedudukan Kata Bahasa Arab Menggunakan Metode <i>Rule-Based</i>	Dari 92 jumlah kata, 3,2% kata yang tidak sesuai pada uji coba jenis kata, 1,1% kata yang tidak sesuai pada uji coba jenis i'rab, dan 6,5% kata yang tidak sesuai pada uji coba kedudukan kata
3	(Mahastama, 2022)	<i>Rule-Based</i>	Model Berbasis Aturan untuk Transliterasi Bahasa Jawa dengan Aksara Latin ke Aksara Jawa	Tahap 1: Dari 600 jumlah kata, 2,8% kata yang salah Tahap 2: Dari 40 kalimat dengan 147 jumlah kata, 12 kalimat yang salah
4	(Mahastama, 2022)	<i>Rule-Based Syllabification</i>	Aplikasi Bantu Belajar Menulis Aksara Jawa untuk Siswa Sekolah Dasar	Dari 500 jumlah kata, 3,4% kata yang salah
5	(Putra et al., 2019)	Pendekatan <i>Rule-Based</i>	<i>Syllabification of Balinese Words Using the Syllabification Algorithm</i>	Dari 257 jumlah kata, 5% kata yang salah
6	(Wibowo et al., 2021)	<i>Syllabification dan String Matching</i>	Implementasi <i>Syllabification Algorithm</i> dan <i>String Matching Algorithm</i> untuk Deteksi Kesalahan Pemisahan	Dari 2.084 jumlah kata, 21% kata yang salah

NO	PENELITI	METODE	JUDUL	HASIL
			Suku Kata Pada <i>E-Paper</i> Artikel Berita	
7	(Arifin et al., 2019)	<i>Finite State Machine</i> (FSM)	Segmentasi Suku Kata Bahasa Indonesia Berbasis Aturan untuk Pengembangan Sistem <i>Text-to-Audiovisual</i>	Terdapat masalah dalam proses segmentasi pada kata yang menyerupai kata diftong

## 2.2 Teori Pendukung

### 2.2.1 Bahasa Aceh

Bahasa Aceh berasal dari gabungan bahasa yang terdapat dalam beberapa wilayah di dunia. Hal tersebut dibuktikan dengan adanya beragam bahasa yang mirip dengan Bahasa Aceh. Beberapa contoh bahasa yang mempunyai kemiripan dengan Bahasa Aceh seperti pada Bahasa Arab, Inggris, Portugis, Filipina (Tagalog), dan Champa dapat dilihat pada tabel di bawah ini (Asyi, 2019).

Tabel 2.2 Bahasa Arab – Bahasa Aceh

Arab	Aceh	Indonesia
<i>Salam</i>	<i>Saleum</i>	Salam
<i>Khabar</i>	<i>Haba</i>	Kabar
<i>Sikin</i>	<i>Sikin</i>	Pisau

Tabel 2.3 Bahasa Inggris – Bahasa Aceh

Inggris	Aceh	Indonesia
<i>Film</i>	<i>Pilem</i>	Film
<i>Artist</i>	<i>Areutih</i>	Artis
<i>Ballon</i>	<i>Balon</i>	Balon

Tabel 2.4 Bahasa Portugis - Bahasa Aceh

Potugis	Aceh	Indonesia
<i>Bandeira</i>	<i>Beundera</i>	Bendera
<i>Janella</i>	<i>Jeundela</i>	Jendela
<i>Bola</i>	<i>Bola</i>	Bola

Tabel 2.5 Bahasa Tagalog - Bahasa Aceh

<b>Tagalog</b>	<b>Aceh</b>	<b>Indonesia</b>
<i>Pinto</i>	<i>Pinto</i>	Pintu
<i>Apoy</i>	<i>Apui</i>	Api
<i>Hukom</i>	<i>Hukom</i>	Hukum

Tabel 2.6 Bahasa Champa - Bahasa Aceh

<b>Champa</b>	<b>Aceh</b>	<b>Indonesia</b>
<i>Cim</i>	<i>Cicem</i>	Burung
<i>Hu</i>	<i>Hu</i>	Nyala
<i>Piyoh</i>	<i>Piyoh</i>	Singhah

Secara umum, Bahasa Aceh adalah bahasa yang digunakan sebagai sarana berkomunikasi dalam kehidupan sehari-hari bagi masyarakat Aceh (Subhayni et al., 2020). Bahasa Aceh juga merupakan bahasa ibu atau bahasa pertama yang diajarkan kepada masyarakat Aceh (Rahma, 2020). Beberapa contoh kata dalam Bahasa Aceh beserta arti dapat dilihat pada Tabel 2.7 (Rizki & Junaidi, 2020).

Tabel 2.7 Contoh Kata Bahasa Aceh

<b>Kata</b>	<b>Arti</b>
<i>Galak</i>	Suka
<i>Gata</i>	Kamu
<i>Giduek</i>	Duduk di atas sesuatu
<i>Haba</i>	Kabar
<i>Lumpoe</i>	Mimpi
<i>Meujaga</i>	Bergadang
<i>Padum</i>	Berapa
<i>Pajan</i>	Kapan
<i>Seumangat</i>	Semangat
<i>Tanyeung</i>	Tanya

### 2.2.2 Silabifikasi

Silabifikasi adalah sebutan untuk proses dalam melakukan pemisahan atau segmentasi pada sebuah kata menjadi suku kata sesuai dengan aturan dan kaidah tertentu (Wibowo et al., 2021). Silabifikasi merupakan salah satu proses yang

penting dilakukan untuk mengembangkan sistem *text-to-speech* sehingga dari hasil pemenggalan kata yang terdiri dari huruf vokal dan konsonan dapat mengetahui bagaimana cara membaca suatu kata pada bahasa yang digunakan (Rahman et al., 2021).

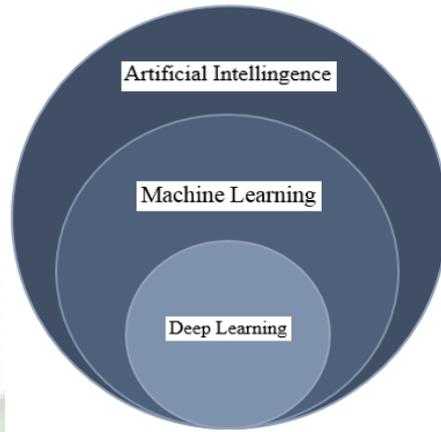
Penentuan huruf vokal, konsonan, diftong, dan juga gabungan konsonan terlebih dahulu dilakukan sebelum memulai proses silabifikasi. Penerapan cara membaca maupun berbicara yang berdasarkan suku kata atau silabifikasi dapat mempengaruhi peningkatan kemampuan berbicara (Suyadi & Sari, 2021). Beberapa contoh kata yang disilabifikasi dalam Bahasa Aceh dapat dilihat pada Tabel 2.8.

Tabel 2.8 Contoh Silabifikasi Kata dalam Bahasa Aceh

Kata	Silabifikasi	Arti
<i>Galak</i>	<i>Ga-lak</i>	Suka
<i>Gata</i>	<i>Ga-ta</i>	Kamu
<i>Giduek</i>	<i>Gi-duek</i>	Duduk di atas sesuatu
<i>Haba</i>	<i>Ha-ba</i>	Kabar
<i>Lumpoe</i>	<i>Lum-poe</i>	Mimpi
<i>Meujaga</i>	<i>Meu-ja-ga</i>	Bergadang
<i>Padum</i>	<i>Pa-dum</i>	Berapa
<i>Pajan</i>	<i>Pa-jan</i>	Kapan
<i>Seumangat</i>	<i>Seu-ma-ngat</i>	Semangat
<i>Tanyeung</i>	<i>Ta-nyeung</i>	Tanya

### 2.2.3 Artificial Intelligence (AI)

*Artificial Intelligence* (AI) adalah kecerdasan buatan yang dimasukkan ke dalam suatu sistem sehingga dapat menyelesaikan pekerjaan tertentu seperti yang dilakukan oleh manusia (Siahaan et al., 2020). *Artificial Intelligence* (AI) merupakan cabang ilmu komputer dalam membangun suatu perangkat lunak dan perangkat keras sehingga dapat menirukan beberapa fungsi dari pemikiran atau cara berfikir manusia. Basis pengetahuan sangat penting dalam *Artificial Intelligence* atau kecerdasan buatan sehingga dapat diterapkan pada pengambilan keputusan dan dapat menyelesaikan masalah (Ratama & Munawaroh, 2019). Ilustrasi hubungan dari *Artificial Intelligence*, *Machine Learning*, dan *Deep Learning* dapat dilihat pada Gambar 2.1.



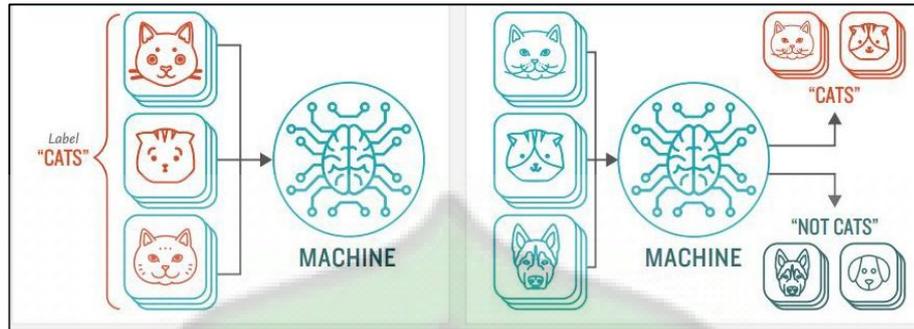
Gambar 2.1 *Artificial Intelligence, Machine Learning, dan Deep Learning*

Sumber: levity.ai

#### 2.2.4 *Machine Learning*

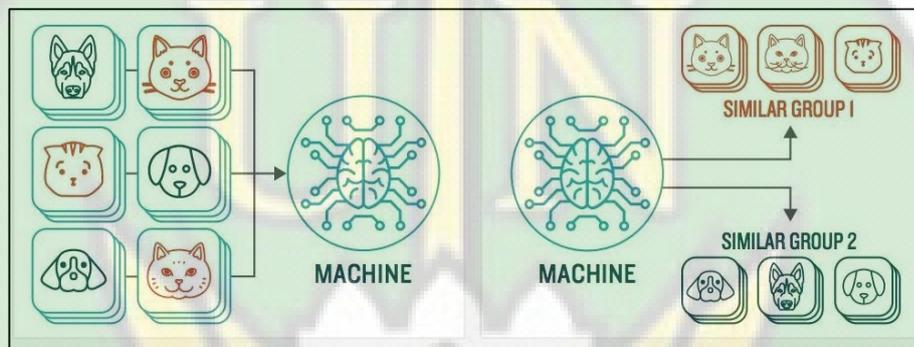
*Machine learning* atau pembelajaran mesin adalah salah satu bagian dari kecerdasan buatan yang bertujuan untuk menyelesaikan masalah dengan menggunakan algoritma. Kemampuan belajar *machine learning* dapat meningkat secara otomatis berdasarkan data-data yang tersedia (Kusuma, 2020).

Berdasarkan cara belajarnya, *machine learning* terbagi menjadi 3 kelompok yaitu: *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. *Supervised learning* atau pembelajaran yang terarah merupakan suatu proses pembelajaran pada mesin dengan menggunakan data *training* yang berlabel. Sehingga dengan menggunakan data *training* yang berlabel, mesin dapat belajar dari data yang didalamnya berisi solusi dan kesimpulan sebagai petunjuk. Sedangkan *Unsupervised Learning* kebalikan dari *supervised learning*. *Unsupervised learning* merupakan proses pembelajaran yang dilakukan tanpa adanya arahan yang berarti pembelajaran dengan menggunakan data yang tidak berlabel. Dalam proses ini mesin hanya mengelompokkan atau membagi isi data berdasarkan kategorinya masing-masing. Berbeda dengan *reinforcement learning*, kategori ini melakukan pembelajaran mesin melalui sebuah algoritma dengan cara berinteraksi dengan lingkungan sehingga dapat mengambil keputusan atau kesimpulan dari suatu permasalahan (Daqiqil, 2021). Gambaran perbedaan antara *Supervised Learning* dan *Unsupervised Learning* dapat dilihat pada Gambar 2.2 dan 2.3.



Gambar 2.2 *Supervised Learning*

Sumber: Boozalen.com



Gambar 2.3 *Unsupervised Learning*

Sumber: Boozalen.com

*Machine learning* juga merupakan suatu sistem yang dapat belajar secara otomatis dengan menggunakan data yang sudah berlabel (*supervised learning*) maupun data yang tidak memiliki label (*unsupervised learning*) dalam menyelesaikan suatu permasalahan (Retnoningsih & Pramudita, 2020). Pada penelitian ini menggunakan data *supervised learning* yang menggunakan data *training* sebagai pembelajaran pola atau aturan yang diterapkan dalam sistem.

### 2.2.5 Algoritma

Algoritma adalah bidang ilmu yang mempelajari tentang strategi dan teknik penyelesaian pada suatu permasalahan melalui rangkaian langkah-langkah tertentu yang diatur secara sistematis. Tujuannya adalah untuk mencapai hasil yang diinginkan atau menyelesaikan suatu tugas dengan efisien. Dalam prosesnya, algoritma memanfaatkan bahasa yang logis sebagai sarana untuk menyusun

instruksi-instruksi yang jelas dan terdefinisi dengan baik (Aritonang, 2022). Algoritma tidak hanya memfokuskan pada solusi matematis, tetapi juga dapat diterapkan pada berbagai bidang, termasuk ilmu komputer, dan kecerdasan buatan. Dengan adanya rangkaian langkah-langkah yang disusun secara jelas, dapat membantu menyelesaikan permasalahan dengan lebih efisien dan terstruktur.

#### **2.2.6 *Finite State Machine (FSM)***

*Finite State Machine (FSM)* adalah suatu metode yang terdiri dari sejumlah keadaan (*state*), kejadian (*event*), dan aksi (*action*) yang didapatkan dari proses input dan output. Jika terdapat inputan sistem pada saat sistem sedang berada dalam suatu *state* yang sedang aktif, maka sistem beralih ke *state* yang lainnya sesuai dengan inputan yang didapat.

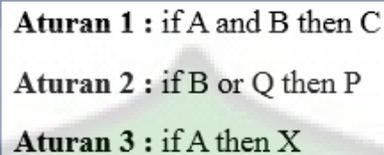
Metode FSM merupakan salah satu metode yang dapat diimplementasikan pada sistem untuk digunakan dalam proses pemenggalan atau segmentasi kata. Penerapan metode FSM dalam proses segmentasi suku kata sudah dilakukan oleh (Arifin et al., 2019). Hasil pengujian berhasil dilakukan, namun masih terdapat beberapa kata yang salah pada proses segmentasi kata yang terdeteksi sebagai kata diftong.

#### **2.2.7 *Rule-Based***

*Rule-Based* adalah suatu metode yang menyelesaikan masalah sesuai dengan aturan (*rule*) yang sudah ditulis (Shabihah, 2019). *Rule-Based* juga dikenal dengan sebutan *knowledge engineering*, yang berarti rekayasa pengetahuan berdasarkan pengetahuan yang diperoleh. *Rule-Based* sistem atau sistem berbasis aturan merupakan suatu metode sederhana yang dapat diaplikasikan untuk menyelesaikan berbagai masalah (Laksono, 2020).

Metode *Rule-Based* merupakan metode paling dasar yang dapat menyelesaikan suatu permasalahan berdasarkan pola dari hasil pembelajaran *dataset* yang diperoleh (Rahman et al., 2021). Secara keseluruhan, aturan dalam sistem *Rule-Based* terdiri dari kondisi (*if*) dan kesimpulan (*then*). Semua aturan yang berisi *if* dan *then* yang terdapat dalam *Rule-Based* tersebut diperiksa oleh sistem (Mulyani et al., 2020). Sehingga jika pada suatu kondisi sudah terpenuhi sesuai aturan yang ditulis maka akan menghasilkan kesimpulan. Pada aturan *Rule-Based* juga bisa ditambahkan dengan kata kunci *AND* dan *OR* sebagai

penghubungnya (Sulistiyo et al., 2020). Aturan (*rule*) yang diterapkan dalam sistem menggunakan metode *Rule-Based* ini adalah aturan yang berdasarkan pengetahuan dari kaidah dalam penuturan Bahasa Aceh. Gambaran aturan pada *Rule-Based* dapat dilihat pada Gambar 2.4.



**Aturan 1 : if A and B then C**  
**Aturan 2 : if B or Q then P**  
**Aturan 3 : if A then X**

Gambar 2.4 Gambaran Aturan *Rule-Based*

Sumber: Sulistiyo et al., 2020

## 2.3 *Tools*

*Tools* yang digunakan dalam penelitian ini adalah bahasa pemrograman *Python* dan menggunakan *Google Colaboratory* dalam melakukan proses pengolahan data.

### 2.3.1 *Python*

*Python* adalah salah satu bahasa pemrograman yang dapat beroperasi di berbagai platform dan digunakan dalam pengembangan perangkat lunak. Bahasa pemrograman *Python* banyak diminati karena struktur bahasa yang mudah dimengerti (Enterprise, 2019). *Python* merupakan *interpreted language*, yang berarti kode program (*source code*) yang ditulis akan dijalankan dari satu baris ke baris selanjutnya (Kurniawan, 2022).

### 2.3.2 *Google Colaboratory*

*Google Colaboratory* adalah suatu platform yang dapat digunakan dengan menggunakan bahasa pemrograman *Python*. Penggunaan *Google Colaboratory* atau disebut juga dengan *Google Colab* dapat membantu dalam pengolahan data. Tempat penyimpanan *Google Colab* adalah *Google Drive* yang berjalan dengan sistem *Cloud* dengan memanfaatkan media penyimpanan *Google*. Penerapan platform *Google Colab* memiliki kemiripan dengan platform *Jupyter Notebook*, salah satu perbedaannya adalah pada platform *Google Colab* hanya dapat dijalankan secara *online*, sedangkan platform *Jupyter Notebook* dapat dijalankan secara *offline* (Rahma et al., 2021).

## 2.4 Model Evaluasi

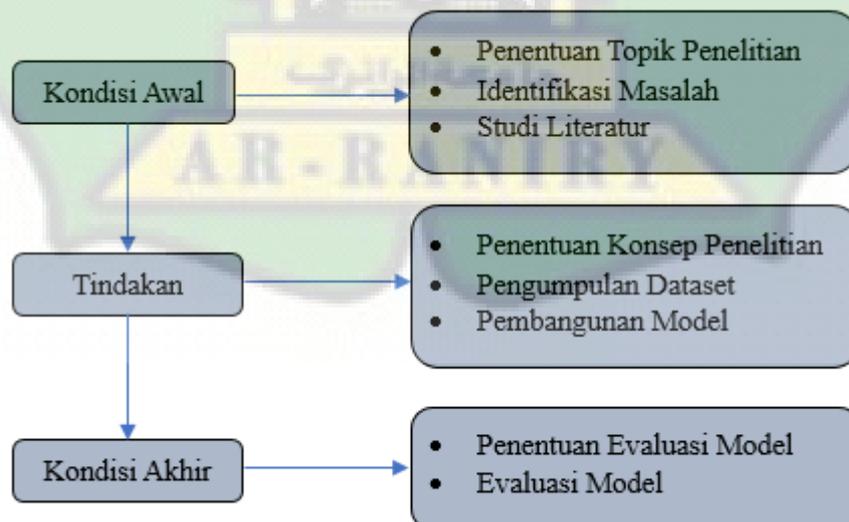
Pada penelitian ini memerlukan model evaluasi untuk mengukur tingkat *Word Error Rate*. Model evaluasi yang digunakan adalah *Word Error Rate* (WER), yang menghitung tingkat akurasi berdasarkan jumlah kata yang salah dibagi dengan keseluruhan jumlah kata. Model evaluasi ini selain dapat digunakan dalam menghitung tingkat WER pada kasus silabifikasi kata, juga digunakan pada kasus speech recognition dan sistem terjemahan. Pada model evaluasi WER, semakin mendekati nol hasil nilai yang didapatkan, maka tingkat keberhasilan model yang dibangun semakin bagus. Dikutip dari penelitian sebelumnya, Rumus *Word Error Rate* (WER) dapat dilihat pada Persamaan 1.

$$\text{WER} = \frac{\text{Jumlah kata yang salah}}{\text{Total jumlah kata}} \quad (1)$$

Sumber: Rahman et al., 2021

## 2.5 Kerangka Pemikiran Penelitian

Dalam melakukan penelitian diperlukan kerangka pemikiran pada suatu penelitian. Kerangka pemikiran penelitian terdiri dari gambaran umum secara keseluruhan pada suatu penelitian yang dimulai dari penentuan topik, pengumpulan data, dan pembangunan model. Gambaran kerangka pemikiran penelitian dapat dilihat pada Gambar 2.5.



Gambar 2.5 Kerangka Pemikiran Penelitian

Berikut uraian dari kerangka pemikiran pada penelitian ini berdasarkan pada Gambar 2.5.

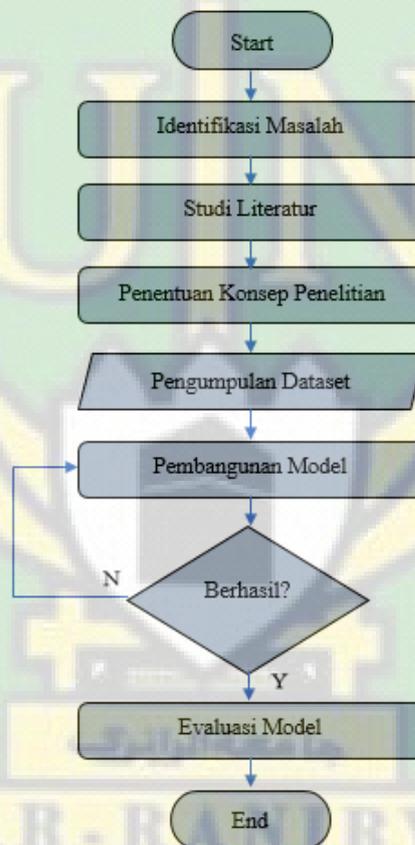
- Penentuan topik penelitian yang dipilih adalah yang berkaitan dengan *machine learning*.
- Pengumpulan data dilakukan berdasarkan permasalahan yang diteliti, yaitu proses silabifikasi pada kata Bahasa Aceh. Sehingga diperlukan beberapa penelitian terdahulu sebagai studi literatur.
- Penentuan konsep penelitian dipilih berdasarkan kesesuaian antara permasalahan yang diteliti dengan metode yang digunakan sehingga metode yang digunakan adalah metode *Rule-Based*.
- Pengumpulan *dataset* dilakukan dengan tujuan untuk keperluan proses pembangunan model. *Dataset* yang dikumpulkan dibagi menjadi 2, yaitu data *training* dan data *testing*.
- Pembangunan model dilakukan pada platform *Google Colab* dengan menggunakan bahasa pemrograman *python*.
- Penentuan model evaluasi dilakukan dengan tujuan untuk keperluan proses mendapatkan hasil keakuratan model. Model evaluasi yang digunakan adalah *Word Error Rate* (WER).
- Evaluasi model dilakukan berdasarkan model evaluasi yang dipilih sehingga mendapatkan hasil keakuratan dari model yang dibangun.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

Pada pembahasan tahapan penelitian akan menjelaskan terkait alur penelitian yang dilakukan. Secara umum tahapan penelitian seperti terlihat pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

#### 3.2 Metode Pengumpulan Data

Metode pengumpulan data dilakukan berdasarkan informasi yang didapatkan dari hasil proses identifikasi masalah dan penjelasan lebih lanjut pada studi literatur.

### **3.2.1 Identifikasi Masalah**

Tahapan ini merupakan tahapan pertama yang dilakukan untuk menemukan suatu permasalahan dan dari permasalahan tersebut diangkat menjadi topik penelitian. Pada tahapan ini peneliti menemukan suatu permasalahan yaitu ketidakmampuan dalam berkomunikasi menggunakan Bahasa Aceh yang disebabkan oleh kurangnya pengetahuan terkait pengucapan kata dalam Bahasa Aceh.

### **3.2.2 Studi Literatur**

Tahapan studi literatur berfungsi untuk menemukan solusi yang berkaitan dengan permasalahan yang terjadi. Tahapan ini dilakukan dengan cara membaca referensi dari penelitian terdahulu yang berkaitan dengan permasalahan yang terjadi. Dari tahapan ini didapatkan sebuah solusi yaitu dengan menerapkan metode *Rule-Based* atau metode lain seperti metode *Finite State Machine* (FSM) pada proses silabifikasi.

## **3.3 Rancangan Model**

Pada rancangan model akan dijelaskan terkait tahapan penentuan konsep penelitian, pengumpulan *dataset*, dan pembangunan model. Tahapan-tahapan tersebut menjadi peran penting dalam menganalisis data dan berfungsi untuk mendapatkan gambaran lebih lanjut tentang model yang dibangun.

### **3.3.1 Penentuan Konsep Penelitian**

Penentuan konsep penelitian didapatkan dari hasil tahapan identifikasi masalah dan solusi dari penelitian terdahulu yang sesuai dengan permasalahan yang terjadi. Tahapan ini berfungsi untuk menentukan metode yang diterapkan pada pembangunan model. Berdasarkan tahapan-tahapan sebelumnya, maka metode yang digunakan dalam pembangunan model pada kasus silabifikasi adalah metode *Rule-Based*.

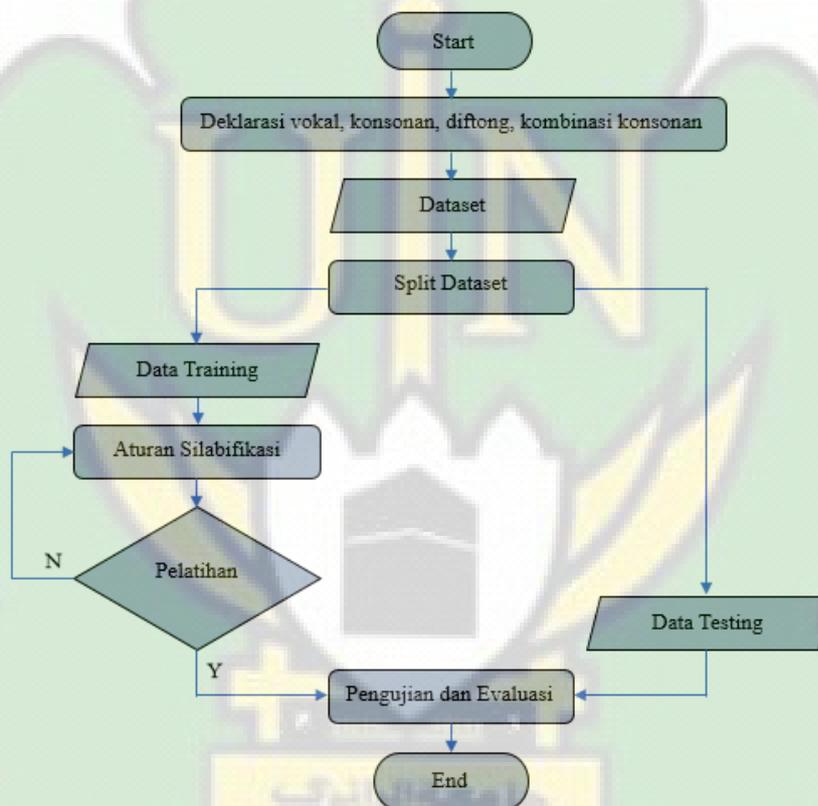
### **3.3.2 Pengumpulan *Dataset***

Tahapan pengumpulan *dataset* dilakukan dengan mengumpulkan beberapa jumlah kata dalam Bahasa aceh. *Dataset* yang digunakan dalam penelitian ini berjumlah 6.225 baris dengan tipe *file txt (text)* yang diperoleh dari penelitian sebelumnya oleh (Ilham, 2023).

*Dataset* yang telah dikumpulkan dibagi menjadi 2 data, yaitu data *training* dan data *testing*. Data *training* digunakan untuk pelatihan atau pembelajaran pola pada model terkait aturan silabifikasi, sedangkan data *testing* digunakan untuk pengujian model terkait aturan silabifikasi yang sudah diterapkan.

### 3.3.3 Pembangunan Model

Pembangunan model dilakukan dengan menggunakan bahasa pemrograman *Python*. Platform yang digunakan adalah *Google Colaboratory* yang hanya bisa diakses secara *online*. Alur kerja model dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur Kerja Model

Penjelasan mengenai alur kerja model pada Gambar 3.2 dapat dilihat sebagai berikut:

- a) Deklarasi vokal, konsonan, diftong, dan gabungan konsonan. Tahapan ini dilakukan dengan cara mendeklarasikan huruf alfabet dalam Bahasa Aceh dengan menjabarkan bentuk dari huruf vokal, konsonan, diftong atau gabungan vokal, dan gabungan konsonan dalam Bahasa Aceh. Deklarasi

huruf vokal, diftong, konsonan, dan gabungan konsonan dapat dilihat pada Tabel 3.1, 3.2, 3.3, dan 3.4 (Rizki & Junaidi, 2020).

Tabel 3.1 Huruf Vokal

Huruf	Contoh Kata	Arti
<i>a</i>	<i>gata</i>	kamu
<i>i</i>	<i>awai</i>	awal
<i>u</i>	<i>gurèe</i>	guru
<i>e</i>	<i>abeh</i>	habis
<i>è</i>	<i>nèk</i>	nenek
<i>é</i>	<i>éh</i>	tidur
<i>o</i>	<i>pujo</i>	puji
<i>ô</i>	<i>lôn</i>	saya
<i>ö</i>	<i>teungöh</i>	tengah
<i>'a</i>	<i>meu'ah</i>	maaf
<i>'i</i>	<i>'idah</i>	idah
<i>'u</i>	<i>ôn 'u</i>	daun kelapa
<i>'è</i>	<i>meu'èn</i>	main
<i>'o</i>	<i>maw'o</i>	mawar
<i>'ö</i>	<i>kl'öt</i>	hitam sekali

Tabel 3.2 Huruf Diftong atau Gabungan Vokal

Huruf	Contoh Kata	Arti
<i>ai</i>	<i>keubai</i>	kebal
<i>'ai</i>	<i>meuh'ai</i>	mahal
<i>ie</i>	<i>ie</i>	air
<i>'ie</i>	<i>cut'iet</i>	cubit
<i>ue</i>	<i>keumuen</i>	sepupu
<i>'ue</i>	<i>'uet</i>	telan
<i>eu</i>	<i>aneuk</i>	anak
<i>'eu</i>	<i>'eungkôt</i>	ikan
<i>èe</i>	<i>batèe</i>	batu
<i>'èe</i>	<i>'èerat</i>	aurat
<i>oe</i>	<i>soe</i>	siapa
<i>öe</i>	<i>lagöe</i>	kali
<i>ei</i>	<i>inggreih</i>	Inggris
<i>oi</i>	<i>boinah</i>	warisan

Huruf	Contoh Kata	Arti
<i>ôi</i>	<i>rhôi</i>	penggaris
<i>öi</i>	<i>lagöina</i>	sangat
<i>ui</i>	<i>tukui</i>	tunduk
<i>eue</i>	<i>saleuem</i>	salam
<i>'eue</i>	<i>bir'euen</i>	bireuen

Tabel 3.3 Huruf Konsonan

Huruf	Contoh Kata	Arti
<i>b</i>	<i>but</i>	kerja
<i>c</i>	<i>cabeueng</i>	cabang
<i>d</i>	<i>darut</i>	belalang
<i>f</i>	<i>seurtifikat</i>	sertifikat
<i>g</i>	<i>sigohlom</i>	sebelum
<i>h</i>	<i>hireuen</i>	heran
<i>j</i>	<i>jaroe</i>	jari
<i>k</i>	<i>kama</i>	kamar
<i>l</i>	<i>leumo</i>	lembu
<i>m</i>	<i>muda</i>	muda
<i>n</i>	<i>niet</i>	niat
<i>p</i>	<i>pageue</i>	pagar
<i>q</i>	-	-
<i>r</i>	<i>rungkhom</i>	menerkam
<i>s</i>	<i>sugot</i>	sisir
<i>t</i>	<i>tuha</i>	tua
<i>v</i>	-	-
<i>w</i>	<i>woe</i>	pulang
<i>x</i>	-	-
<i>y</i>	<i>yahcut</i>	paman
<i>z</i>	-	-

Tabel 3.4 Huruf Gabungan Konsonan

Huruf	Contoh Kata	Arti
<i>ng</i>	<i>jar'eueng</i>	jarang
<i>ny</i>	<i>nyoe</i>	ini
<i>sy</i>	<i>sya'e</i>	syair
<i>ph</i>	<i>phet</i>	pahit
<i>th</i>	<i>tham</i>	larang

Huruf	Contoh Kata	Arti
<i>ch</i>	<i>keuchik</i>	kepala desa
<i>kh</i>	<i>kheun</i>	baca
<i>bh</i>	<i>bhuek</i>	tenggelam
<i>dh</i>	<i>dhoe</i>	dahi
<i>jh</i>	<i>jhô</i>	dorong
<i>gh</i>	<i>gheuem</i>	geraham
<i>lh</i>	<i>peulheueh</i>	melepaskan
<i>rh</i>	<i>rhui</i>	merinding
<i>pl</i>	<i>plueng</i>	lari
<i>cl</i>	<i>clap-clup</i>	suara air
<i>bl</i>	<i>bloe</i>	beli
<i>gl</i>	<i>glie</i>	geli
<i>kl</i>	<i>klèrèng</i>	kelereng
<i>pr</i>	<i>prom</i>	peram
<i>tr</i>	<i>trôh</i>	tiba
<i>cr</i>	<i>croh</i>	goreng
<i>kr</i>	<i>krueng</i>	sungai
<i>br</i>	<i>bri</i>	beri/kasih
<i>dr</i>	<i>drien</i>	durian
<i>jr</i>	<i>jroh</i>	baik sekali
<i>gr</i>	<i>grah</i>	haus

- b) *Dataset*. Pengumpulan *dataset* dilakukan dengan mengumpulkan beberapa jumlah kata dalam Bahasa Aceh. *Dataset* yang digunakan dalam penelitian ini berjumlah 6.225 baris. Namun dalam penelitian ini dibutuhkan 1 *dataset* tambahan, yaitu data *Ground Truth Summarization*. Data tersebut berisi kata yang belum disilabifikasi dan hasil silabifikasi yang sesuai. Tujuan dari *Ground Truth Summarization* adalah untuk membandingkan keakuratan antara hasil yang didapatkan dari model dengan hasil yang didapatkan dari data yang benar (*ground truth*).
- c) *Split dataset*. Tahapan ini dilakukan dengan cara membagikan *dataset* menjadi dua bagian, yaitu data *training* dan data *testing*. Metode yang digunakan untuk pembagian *dataset* menjadi dua bagian adalah dengan menggunakan metode pembagian data acak (*random splitting*) dengan parameter *test\_size=0.2* untuk mengatur perbandingan 80% data *train* dan

20% data *test*, dan parameter *shuffle=True* untuk mengatur data akan diacak sebelum dibagi menjadi data *train* dan data *test*.

- d) Data *train*. Data *train* adalah data yang digunakan sebagai pembelajaran pola atau aturan pada model. Data *train* berisi 80% data dari keseluruhan *dataset*. Berdasarkan jumlah *dataset*, data *train* yang berisi 80% data berjumlah 4.980 baris.
- e) Aturan silabifikasi. Tahapan ini dilakukan dengan menerapkan aturan silabifikasi berdasarkan pembelajaran data dari data *train*. Penerapan aturan silabifikasi sesuai dengan aturan yang ada dalam Bahasa Aceh. Contoh, pada kata '*Seumangat*' disilabifikasi menjadi '*Seu-ma-ngat*'. Berikut beberapa aturan silabifikasi atau segmentasi suku kata yang sudah dirangkum dan dapat dilihat pada Gambar 3.3 dan 3.4 (Hanafiah & Makam, 1984).

```
FUNGSI syllabify(self, kata):
    inisialisasi syllables sebagai array kosong
    inisialisasi variabel 'i' dengan nilai awal 0.
    set length sebagai panjang(kata)

    SELAMA i < length:
        # Rule: 3 vocal (diftong)
        if i + 3 <= length DAN substring(kata, i, i + 3) in self.diftong:
            if i == 0 ATAU substring(kata, i-1, i) in self.vocal:
                Tambahkan substring(kata, i, i+2) sebagai suku kata baru
            elseif i == 0 ATAU substring(kata, i-1, i) terdapat dalam self.consonant:
                Gabungkan substring(kata, i, i+2) ke suku kata sebelumnya
            elseif i > 0 DAN substring(kata, i-3, i-2) == " ":
                Tambahkan substring(kata, i, i+2) sebagai suku kata baru
            else:
                Tambahkan substring(kata, i, i+2) sebagai suku kata baru
                i += 3
        # Rule: 2 vocal (diftong)
        elseif i + 2 <= length DAN substring(kata, i, i + 2) in self.diftong:
            if i == 0 ATAU substring(kata, i-1, i) in self.vocal:
                Tambahkan substring(kata, i, i+1) sebagai suku kata baru
            elseif i == 0 ATAU substring(kata, i-1, i) in self.consonant:
                Gabungkan substring(kata, i, i+1) ke suku kata sebelumnya
            elseif i > 0 DAN substring(kata, i-2, i-1) == " ":
                Tambahkan substring(kata, i, i+1) sebagai suku kata baru
            else:
                Tambahkan substring(kata, i, i+1) sebagai suku kata baru
                i += 2
        # Rule: 1 vocal
        elseif substring(kata, i, i) in self.vocal:
            if i > 0 DAN substring(kata, i-1, i) == " ":
                Tambahkan substring(kata, i, i) sebagai suku kata baru
            elseif i == 0 ATAU substring(kata, i-1, i) in self.vocal:
                Tambahkan substring(kata, i, i) sebagai suku kata baru
            elseif i < length - 1 DAN substring(kata, i+1) == " ":
                Gabungkan substring(kata, i, i) ke suku kata sebelumnya
            else:
                Gabungkan substring(kata, i, i) ke suku kata sebelumnya
                i += 1
    KEMBALIKAN syllables
```

Gambar 3.3 Pseudocode Algoritma Silabifikasi Vokal

Pada Gambar 3.3 berisi 3 aturan untuk vokal. Penjelasan dari *pseudocode* di atas adalah sebagai berikut:

➤ Aturan untuk 3 vokal (diftong).

Baris pertama menjelaskan jika terdapat 3 karakter, dan 3 karakter tersebut terdapat dalam variabel diftong, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika karakter tersebut berada pada indeks pertama atau indeks sebelumnya adalah 1 vokal, maka tambahkan 3 karakter tersebut sebagai suku kata baru. Kondisi kedua menjelaskan bahwa jika karakter tersebut berada pada indeks pertama atau indeks sebelumnya adalah 1 konsonan, maka gabungkan 3 karakter tersebut dengan suku kata sebelumnya. Kondisi ketiga menjelaskan bahwa jika sebelum 3 karakter tersebut terdapat spasi, maka tambahkan sebagai suku kata baru. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka tambahkan sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 3 karakter untuk melanjutkan pencarian.

➤ Aturan untuk 2 vokal (diftong).

Baris pertama menjelaskan jika terdapat 2 karakter, dan 2 karakter tersebut terdapat dalam variabel diftong, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika karakter tersebut berada pada indeks pertama atau indeks sebelumnya adalah 1 vokal, maka tambahkan 2 karakter tersebut sebagai suku kata baru. Kondisi kedua menjelaskan bahwa jika karakter tersebut berada pada indeks pertama atau indeks sebelumnya adalah 1 konsonan, maka gabungkan dengan suku kata sebelumnya. Kondisi ketiga menjelaskan bahwa jika sebelum 2 karakter tersebut terdapat spasi, maka tambahkan sebagai suku kata baru. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka tambahkan sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 2 karakter untuk melanjutkan pencarian.

- Aturan untuk 1 vokal.

Baris pertama menjelaskan bahwa jika karakter pada indeks  $i$  terdapat dalam variabel vokal, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika sebelum karakter tersebut terdapat spasi, maka tambahkan sebagai suku kata baru. Kondisi kedua menjelaskan bahwa jika karakter tersebut berada pada indeks pertama atau indeks sebelumnya adalah 1 vokal, maka tambahkan sebagai suku kata baru. Kondisi ketiga menjelaskan bahwa jika sesudah karakter tersebut terdapat spasi, maka gabungkan dengan suku kata sebelumnya. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka gabungkan dengan suku kata sebelumnya. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 1 karakter untuk melanjutkan pencarian.

```

FUNGSI syllabify(self, kata):
    inisialisasi syllables sebagai array kosong
    inisialisasi variabel 'i' dengan nilai awal 0.
    set length sebagai panjang(kata)

    SELAMA i < length:
        # Rule: 3 consonants (consonant_combination + consonant)
        if (i + 3 <= length DAN
            substring(kata, i, i+1) in self.consonant_combination DAN
            substring(kata, i, i+2, 2) in self.consonant DAN
            substring(kata, i, i+2, 2) not in self.consonant_combination):
            Gabungkan substring(kata, i, i+1) ke suku kata sebelumnya
            Tambahkan substring(kata, i+2, i+2) sebagai suku kata baru
            i += 3
        # Rule: 2 consonants (consonant_combination)
        elseif i + 2 <= length DAN substring(kata, i, i + 2) in self.consonant_combination:
            if i + 1 == length - 1 :
                Gabungkan substring(kata, i, i+1) ke suku kata sebelumnya
            elseif i < length - 1 DAN substring(kata, i+2)== " ":
                Gabungkan substring(kata, i, i+1) ke suku kata sebelumnya
            else:
                Tambahkan substring(kata, i, i+1) sebagai suku kata baru
            i += 2
        # Rule: 1 consonant
        elseif substring(kata, i, i) in self.consonant:
            if i > 0 DAN substring(kata, i-1, i) == " ":
                Tambahkan substring(kata, i, i) sebagai suku kata baru
            elseif i < length - 1 DAN substring(kata, i+1) == " ":
                Gabungkan substring(kata, i, i) ke suku kata sebelumnya
            elseif i == length - 1:
                Gabungkan substring(kata, i, i) ke suku kata sebelumnya
            else:
                Tambahkan substring(kata, i, i) sebagai suku kata baru
            i += 1
    KEMBALIKAN syllables
  
```

Gambar 3.4 *Pseudocode* Algoritma Silabifikasi Konsonan

Pada Gambar 3.4 berisi 3 aturan untuk konsonan. Penjelasan dari *pseudocode* di atas adalah sebagai berikut:

- Aturan untuk 3 konsonan (kombinasi konsonan + konsonan).

Baris pertama sampai baris ke-4 menjelaskan bahwa jika terdapat 3 karakter konsonan, dimana dua karakter pertama adalah kombinasi konsonan, dan 1 karakter terakhir adalah sebuah konsonan dan tidak ada dalam kombinasi konsonan, maka jalankan kondisi di bawahnya. Langkah pertama adalah dengan menggabungkan 2 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan karakter ke-3 sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 3 karakter untuk melanjutkan pencarian.

➤ Aturan untuk 2 konsonan (kombinasi konsonan).

Baris pertama menjelaskan bahwa jika terdapat 2 karakter, dan 2 karakter tersebut terdapat dalam variabel kombinasi konsonan, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika karakter tersebut berada pada indeks terakhir, maka gabungkan dengan suku kata sebelumnya. Kondisi kedua menjelaskan bahwa jika setelah 2 karakter tersebut terdapat spasi, maka gabungan dengan suku kata sebelumnya. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka tambahkan sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 2 karakter untuk melanjutkan pencarian.

➤ Aturan untuk 1 konsonan.

Baris pertama menjelaskan bahwa jika karakter pada indeks i terdapat dalam variabel konsonan, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika sebelum karakter tersebut terdapat spasi, maka tambahkan sebagai suku kata baru. Kondisi kedua menjelaskan bahwa jika sesudah karakter tersebut terdapat spasi, maka gabungkan dengan suku kata sebelumnya. Kondisi ketiga menjelaskan bahwa jika karakter tersebut berada pada indeks terakhir, maka gabungkan dengan suku kata sebelumnya. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka tambahkan sebagai suku kata baru. Dan baris terakhir menjelaskan

bahwa indeks digeser sebanyak 1 karakter untuk melanjutkan pencarian.

Pada proses penerapan aturan silabifikasi, terdapat beberapa aturan yang perlu ditambahkan guna memperbaiki dan memperkaya model yang dilakukan dalam penelitian ini. Dengan demikian, dapat dipastikan bahwa model yang dikembangkan mampu menangani beragam situasi dengan akurat dan efektif. Hal ini merupakan langkah penting dalam meningkatkan kemampuan model untuk memecahkan kata-kata dalam Bahasa Aceh dengan presisi yang tinggi. Beberapa aturan tambahan tersebut dapat dilihat pada Gambar 3.5 sampai 3.7.

```
FUNGSI syllabify(self, kata):
    inialisasi syllables sebagai array kosong
    inialisasi variabel 'i' dengan nilai awal 0.
    set length sebagai panjang(kata)
    SELAMA i < length:
        # Rule: 4 vocal (diftong)
        if i + 4 <= length DAN substring(kata, i+2, i+3) in self.diftong:
            if i == 0 ATAU substring(kata, i-1, i) in self.vocal:
                Tambahkan substring(kata, i+2, i+3) sebagai suku kata baru
            elseif i > 0 DAN substring(kata, i-1, i) == " ":
                Tambahkan substring(kata, i+2, i+3) sebagai suku kata baru
            else:
                Gabungkan substring(kata, i+2, i+3) ke suku kata sebelumnya
                i += 4
        # Rule: 4 vocal (diftong + diftong)
        elseif (i + 4 <= length DAN
                substring(kata, i, i+1) in self.diftong and
                substring(i+2, i+3, 2, 4) in self.diftong):
                Gabungkan substring(kata, i, i+1) ke suku kata sebelumnya
                Tambahkan substring(i+2, i+3, 2, 4) sebagai suku kata baru
                i += 4
        # Rule: 3 vocal (diftong + vocal)
        elseif (i + 3 <= length DAN
                substring(kata, i, i+1) in self.diftong DAN
                substring(kata, i, i+2, 2) in self.vocal DAN
                substring(kata, i, i+2, 2) not in self.diftong):
                Gabungkan substring(kata, i, i+1) ke suku kata sebelumnya
                Tambahkan substring(kata, i+2, i+2) sebagai suku kata baru
                i += 3
        # Rule: 3 vocal (vocal + diftong)
        elseif (i + 3 <= length DAN
                substring(kata, i, i) in self.vocal DAN
                substring(kata, i+1, i+2) in self.diftong):
                Gabungkan substring(kata, i, i) ke suku kata sebelumnya
                Tambahkan substring(kata, i+1, i+2) sebagai suku kata baru
                i += 3
        # Rule: 2 vocal (bukan diftong)
        elseif (i + 2 <= length DAN
                substring(kata, i, i+1) not in self.diftong DAN
                substring(kata, i, i, 1) in self.vocal DAN
                substring(kata, i+1, i+1, 1) in self.vocal):
                if i == 0:
                    Tambahkan substring(kata, i, i, 1) sebagai suku kata baru
                    Tambahkan substring(kata, i+1, i+1, 1) sebagai suku kata baru
                else:
                    Gabungkan substring(kata, i, i, 1) ke suku kata sebelumnya
                    Tambahkan substring(kata, i+1, i+1, 1) sebagai suku kata baru
                i += 2
    KEMBALIKAN syllables
```

Gambar 3.5 Pseudocode Algoritma Silabifikasi Tambahan Vokal

Pada Gambar 3.5 berisi 5 aturan tambahan untuk vokal. Penjelasan dari *pseudocode* di atas adalah sebagai berikut:

➤ Aturan untuk 4 vokal (diftong).

Baris pertama menjelaskan bahwa jika terdapat 4 karakter, dan 4 karakter tersebut terdapat dalam variabel diftong, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika karakter tersebut berada pada indeks pertama atau indeks sebelumnya adalah 1 vokal, maka tambahkan sebagai suku kata baru. Kondisi kedua menjelaskan bahwa jika sebelum 4 karakter tersebut terdapat spasi, maka tambahkan sebagai suku kata baru. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka gabungkan dengan suku kata sebelumnya. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 4 karakter untuk melanjutkan pencarian.

➤ Aturan untuk 4 vokal (diftong + diftong).

Baris pertama sampai baris ke-3 menjelaskan bahwa jika terdapat 4 karakter vokal, dimana dua karakter pertama adalah diftong, dan 2 karakter terakhir adalah diftong, maka jalankan kondisi di bawahnya. Langkah pertama adalah dengan menggabungkan 2 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan 2 karakter terakhir sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 4 karakter untuk melanjutkan pencarian.

➤ Aturan untuk 3 vokal (diftong + vokal).

Baris pertama sampai baris ke-4 menjelaskan bahwa jika terdapat 3 karakter vokal, dimana dua karakter pertama adalah diftong, dan 1 karakter terakhir adalah sebuah vokal, maka jalankan kondisi di bawahnya. Langkah pertama adalah dengan menggabungkan 2 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan 1 karakter terakhir sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 3 karakter untuk melanjutkan pencarian.

➤ Aturan untuk 3 vokal (vokal + diftong).

Baris pertama sampai baris ke-3 menjelaskan bahwa jika terdapat 3 karakter vokal, dimana 1 karakter pertama adalah sebuah vokal, dan 2 karakter terakhir adalah diftong, maka jalankan kondisi di bawahnya. Langkah pertama adalah dengan menggabungkan 1 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan 2 karakter terakhir sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 3 karakter untuk melanjutkan pencarian.

➤ Aturan untuk 2 vokal (bukan diftong).

Baris pertama sampai baris ke-4 menjelaskan bahwa jika terdapat 2 karakter vokal, dimana 1 karakter pertama adalah sebuah vokal, dan 1 karakter terakhir juga sebuah vokal. Akan tetapi, dua karakter tersebut tidak termasuk ke dalam variabel diftong, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika karakter tersebut berada pada indeks pertama, maka langkah pertama yang dilakukan adalah dengan menambahkan 1 karakter pertama sebagai suku kata baru, dan langkah berikutnya adalah dengan menambahkan 1 karakter terakhir sebagai suku kata baru. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka langkah pertama yang dilakukan adalah dengan menggabungkan 1 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan 1 karakter terakhir sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 2 karakter untuk melanjutkan pencarian.

```

FUNGSI syllabify(self, kata):
    inisialisasi syllables sebagai array kosong
    inisialisasi variabel 'i' dengan nilai awal 0.
    set length sebagai panjang(kata)

    SELAMA i < length:
    # Rule: 4 consonants (consonant_combination + consonant_combination)
    if (i + 4 <= length DAN
        substring(kata, i, i+1) in self.consonant_combination DAN
        substring(i+2, i+3, 2, 4) in self.consonant_combination):
        Gabungkan substring(kata, i, i+1) ke suku kata sebelumnya
        Tambahkan substring(i+2, i+3, 2, 4) sebagai suku kata baru
        i += 4
    # Rule: 3 consonants (consonant_combination)
    elseif i + 3 <= length DAN substring(kata, i, i + 3) in self.consonant_combination:
        if i + 1 == length - 1 :
            Gabungkan substring(kata, i, i + 3) ke suku kata sebelumnya
        else:
            Tambahkan substring(kata, i, i + 3) sebagai suku kata baru
            i += 3
    # Rule: 3 consonants (consonant + consonant_combination)
    elseif (i + 3 <= length DAN
        substring(kata, i, i) in self.consonant DAN
        substring(kata, i+1, i+2) in self.consonant_combination):
        Gabungkan substring(kata, i, i) ke suku kata sebelumnya
        Tambahkan substring(kata, i+1, i+2) sebagai suku kata baru
        i += 3
    # Rule: 2 consonants (bukan consonant_combination)
    elseif (i + 2 <= length DAN
        substring(kata, i, i+1) not in self.consonant_combination DAN
        substring(kata, i, i, 1) in self.consonant DAN
        substring(kata, i+1, i+1, 1) in self.consonant):
        if i == 0:
            Tambahkan substring(kata, i, i+1) sebagai suku kata baru
        else:
            Gabungkan substring(kata, i, i, 1) ke suku kata sebelumnya
            Tambahkan substring(kata, i+1, i+1, 1) sebagai suku kata baru
        i += 2
    KEMBALIKAN syllables

```

Gambar 3.6 *Pseudocode* Algoritma Silabifikasi Tambahkan Konsonan

Pada Gambar 3.6 berisi 4 aturan tambahan untuk konsonan. Penjelasan dari *pseudocode* di atas adalah sebagai berikut:

- Aturan untuk 4 konsonan (kombinasi konsonan + kombinasi konsonan).

Baris pertama sampai baris ke-3 menjelaskan bahwa jika terdapat 4 karakter konsonan, dimana 2 karakter pertama adalah kombinasi konsonan, dan 2 karakter terakhir adalah kombinasi konsonan, maka jalankan kondisi di bawahnya. Langkah pertama adalah dengan menggabungkan 2 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan 2 karakter terakhir sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 4 karakter untuk melanjutkan pencarian.

- Aturan untuk 3 konsonan (kombinasi konsonan ).

Baris pertama menjelaskan bahwa jika terdapat 3 karakter konsonan, dimana 3 karakter tersebut terdapat dalam variabel kombinasi konsonan , maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika karakter tersebut berada pada indeks terakhir, maka gabungkan dengan suku kata sebelumnya. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka tambahkan sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 3 karakter untuk melanjutkan pencarian.

- Aturan untuk 3 konsonan (konsonan + kombinasi konsonan ).

Baris pertama sampai baris ke-3 menjelaskan bahwa jika terdapat 3 karakter konsonan, dimana 1 karakter pertama adalah sebuah konsonan, dan 2 karakter terakhir adalah kombinasi konsonan , maka jalankan kondisi di bawahnya. Langkah pertama adalah dengan menggabungkan 1 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan 2 karakter terakhir sebagai suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 3 karakter untuk melanjutkan pencarian.

- Aturan untuk 2 konsonan (bukan kombinasi konsonan ).

Baris pertama sampai baris ke-4 menjelaskan bahwa jika terdapat 2 karakter konsonan, dimana 1 karakter pertama adalah sebuah konsonan, dan 1 karakter terakhir juga sebuah konsonan. Akan tetapi, 2 karakter tersebut tidak terdapat dalam variabel kombinasi konsonan, maka jalankan kondisi di bawahnya. Pada kondisi pertama menjelaskan bahwa jika karakter tersebut berada pada indeks pertama, maka tambahkan sebagai suku kata baru. Kondisi terakhir menjelaskan bahwa jika tidak ada kondisi yang terpenuhi, maka langkah pertama yang dilakukan adalah dengan menggabungkan 1 karakter pertama dengan suku kata sebelumnya, dan langkah berikutnya adalah dengan menambahkan 1 karakter terakhir sebagai

suku kata baru. Dan baris terakhir menjelaskan bahwa indeks digeser sebanyak 2 karakter untuk melanjutkan pencarian.

```

FUNGSI syllabify(self, kata):
    inisialisasi syllables sebagai array kosong
    inisialisasi variabel 'i' dengan nilai awal 0.
    set length sebagai panjang(kata)

    SELAMA i < length:
    # Rule: special words
    if kata in self.kata_khusus:
        Tambahkan kata sebagai suku kata baru
        Tambahkan panjang kata ke dalam i
    # Rule: Handle jika ada tanda kutip tunggal (')
    elseif substring(kata, i, i) == "'":
        if i > 0 DAN substring(kata, i-1, i) in self.consonant:
            Gabungkan substring(kata, i, i) ke suku kata sebelumnya
        else:
            Tambahkan substring(kata, i, i) sebagai suku kata baru
        i += 1
    # Rule : Handle jika ada spasi
    elseif substring(kata, i, i) == " ":
        Tambahkan substring(kata, i, i) sebagai suku kata baru
        i += 1

    Selama ada spasi dalam syllables:
        Hapus spasi dari syllables
    KEMBALIKAN syllables
    
```

Gambar 3.7 Pseudocode Algoritma Silabifikasi Tambahan Khusus

Pada Gambar 3.7 berisi 3 aturan tambahan khusus untuk mengatur jika suatu kondisi terdapat spasi dan tanda kutip tunggal dalam kata. Contoh kata dari beberapa aturan silabifikasi dapat dilihat pada Tabel 3.5.

Tabel 3.5 Contoh Kata Aturan Silabifikasi

Aturan	Contoh Kata	Hasil
1 vokal	<i>a</i>	<i>a</i>
2 vokal	<i>ie</i>	<i>ie</i>
3 vokal	<i>'eue</i>	<i>'eue</i>
vokal+konsonan	<i>éh</i>	<i>eh</i>
konsonan+vokal	<i>ka</i>	<i>ka</i>
konsonan+vokal+konsonan	<i>poh</i>	<i>poh</i>
konsonan+konsonan+vokal	<i>klô</i>	<i>klo</i>
konsonan+konsonan+vokal+konsonan	<i>plah</i>	<i>plah</i>
1 vokal di awal kata	<i>alue</i>	<i>a-lue</i>
1 vokal di pertengahan kata	<i>keurija</i>	<i>keu-ri-ja</i>
1 vokal di akhir kata	<i>haba</i>	<i>ha-ba</i>

Aturan	Contoh Kata	Hasil
1 konsonan di awal kata	<i>kah</i>	<i>kah</i>
1 konsonan di pertengahan kata	<i>pula</i>	<i>pu-la</i>
1 konsonan di akhir kata	<i>pjuet</i>	<i>pi-juet</i>
2 vokal (diftong) yang setelah huruf tersebut adalah huruf terakhir	<i>droeneuh</i>	<i>droe-neuh</i>
2 vokal (diftong) di akhir kata	<i>dhoe</i>	<i>dhoe</i>
2 vokal (diftong) di pertengahan kata	<i>geutanyoe</i>	<i>geu-ta-nyoe</i>
2 vokal bukan diftong	<i>dua</i>	<i>du-a</i>
3 vokal (diftong) yang setelah huruf tersebut adalah huruf terakhir	<i>peuet</i>	<i>peuet</i>
3 vokal (diftong) di akhir kata	<i>'eue</i>	<i>'eue</i>
3 vokal (diftong) di pertengahan kata	<i>kureueng</i>	<i>ku-reueng</i>
3 vokal (1 vokal bukan diftong dan 2 vokal adalah diftong)	<i>seui</i>	<i>seu-i</i>
2 konsonan (gabungan konsonan) yang setelah huruf tersebut adalah huruf terakhir	<i>deungo</i>	<i>deu-ngo</i>
2 konsonan (gabungan konsonan) di akhir kata	<i>teurimong</i>	<i>teu-ri-mong</i>
2 konsonan (gabungan konsonan) di pertengahan kata	<i>geulinyueng</i>	<i>geu-li-nyueng</i>
2 konsonan dan bukan gabungan konsonan	<i>bintang</i>	<i>bin-tang</i>
3 konsonan (1 konsonan bukan gabungan konsonan dan 2 konsonan adalah gabungan konsonan)	<i>bangke</i>	<i>bang-ke</i>

- f) Pelatihan, pada tahapan pelatihan menggunakan data *train* sebagai data latih. Pelatihan dilakukan dengan menerapkan aturan silabifikasi yang sudah dibuat.
- g) Data *test*, data *test* adalah data yang digunakan untuk evaluasi model. Data *test* berisi 20% data dari keseluruhan *dataset*. Berdasarkan jumlah *dataset*, data *test* yang berisi 20% data berjumlah 1.245 baris.
- h) Pengujian dan evaluasi. Tahapan ini merupakan tahapan terakhir dari alur kerja model. Pada tahap pengujian menggunakan data *test* sebagai data uji. Hasil dari tahap pengujian ini menghasilkan sejumlah kata yang tidak berhasil disilabifikasi dengan benar. Sedangkan dari tahap evaluasi

menghasil nilai WER (Word Error Rate) yang menggambarkan tingkat kesalahan antara hasil prediksi model dengan teks referensi.

### 3.4 Metode Analisis Hasil

Tahapan terakhir dalam penelitian ini adalah metode analisis hasil. Metode analisis hasil dilakukan dengan melakukan evaluasi pada model yang dibangun. Tujuannya adalah agar dapat mengetahui tingkat akurasi atau tingkat keberhasilan model yang dibangun.

#### 3.4.1 Evaluasi Model

Dalam melakukan suatu penelitian dibutuhkan evaluasi model untuk mendapatkan hasil akurasi atau hasil kesesuaian antara rancangan model dengan hasil model yang dibuat. Penelitian ini menggunakan model evaluasi *Word Error Rate* (WER) terhadap hasil model yang menerapkan metode *Rule-Based* dalam proses silabifikasi pada kata Bahasa Aceh. Penentuan akurasi menggunakan WER dilakukan dengan menghitung jumlah kata yang salah dibagi dengan keseluruhan jumlah kata. Rumus untuk menghitung tingkat akurasi menggunakan WER dapat dilihat pada Persamaan 1. Hasil yang didapat akan menentukan tingkat WER yang diperoleh dari model. Jika hasil nilai yang didapatkan semakin mendekati nol, maka tingkat keberhasilan model yang dibangun semakin bagus. Contoh, pada suatu model yang mendapatkan nilai WER sebesar 0,01 (1%) maka tingkat akurasi dari hasil model yang diprediksi dengan benar adalah 99%.

### 3.5 Alat Bantu Penelitian

Spesifikasi *hardware* dan *software* yang digunakan dalam proses pembangunan model dalam penelitian ini adalah:

- i) Laptop Asus A455L.
- j) Sistem Operasi *Windows 8.1 Pro*.
- k) *Processor Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz*.
- l) RAM 8,00 GB (7,89 GB *usable*).
- m) Platform *Google Colaboratory*.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Pelatihan Model

Bahan pelatihan utama untuk mengajarkan model tentang aturan dan pola dalam silabifikasi pada kata Bahasa Aceh adalah menggunakan data *train*. Data *train* adalah sumber daya yang penting dalam proses pembelajaran mesin. Hasil dari pelatihan model, yang mencakup kemampuan model dalam memprediksi silabifikasi kata dapat dilihat pada Tabel 4.1.

Tabel 4.1 Sampel Hasil Data *Train*

Kata	Prediksi	Hasil Silabifikasi
<i>alamat</i>	'a', 'la', 'mat'	'a', 'la', 'mat'
<i>agenda</i>	'a', 'gen', 'da'	'a', 'gen', 'da'
<i>brat</i>	'brat'	'brat'
<i>komidi</i>	'ko', 'mi', 'di'	'ko', 'mi', 'di'
<i>wareuna</i>	'wa', 'reu', 'na'	'wa', 'reu', 'na'
<i>sikada</i>	'si', 'ka', 'da'	'si', 'ka', 'da'
<i>ngui</i>	'ngui'	'ngui'
<i>awak elit</i>	'a', 'wak', 'e', 'lit'	'a', 'wak', 'e', 'lit'
<i>reugu</i>	'reu', 'gu'	'reu', 'gu'
<i>mupaloe</i>	'mu', 'pa', 'loe'	'mu', 'pa', 'loe'

Pada Tabel 4.1 berisi sampel hasil data *train*, yang mencakup hasil silabifikasi dari *ground truth* dan hasil prediksi dari proses pelatihan model. Tabel ini memberikan gambaran tentang hasil pelatihan model dan merupakan sumber data yang digunakan untuk menghasilkan prediksi yang lebih akurat. Dari tabel tersebut dapat dilihat bahwa hasil yang didapatkan dari model menggunakan data *train* sesuai antara hasil silabifikasi dari *ground truth* dan hasil prediksi yang didapatkan dari pelatihan model. Proses dalam pelatihan model penting dalam pengembangan model silabifikasi, dan membantu dalam memperbaiki dan

mengoptimalkan model agar semakin akurat dan efisien dalam mengatasi tantangan dalam bahasa Aceh

#### 4.2 Hasil Pengujian Model

Pengujian model dilakukan dengan menggunakan data *test* sebanyak 1.245 kata. Sampel hasil data *test* yang digunakan dalam tahap pengujian model dapat dilihat pada Tabel 4.2.

Tabel 4.2 Sampel Hasil Data *Test*

Kata	Prediksi	Hasil Silabifikasi
<i>teubit</i>	<i>['teu', 'bit']</i>	<i>['teu', 'bit']</i>
<i>galak</i>	<i>['ga', 'lak']</i>	<i>['ga', 'lak']</i>
<i>gadoh</i>	<i>['ga', 'doh']</i>	<i>['ga', 'doh']</i>
<i>bijeh</i>	<i>['bi', 'jeh']</i>	<i>['bi', 'jeh']</i>
<i>panteue</i>	<i>['pan', 'teue']</i>	<i>['pan', 'teue']</i>
<i>sikawan</i>	<i>['si', 'ka', 'wan']</i>	<i>['si', 'ka', 'wan']</i>
<i>ureueng</i>	<i>['u', 'reueng']</i>	<i>['u', 'reueng']</i>
<i>kapai</i>	<i>['ka', 'pai']</i>	<i>['ka', 'pai']</i>
<i>silamanya</i>	<i>['si', 'la', 'ma', 'nya']</i>	<i>['si', 'la', 'ma', 'nya']</i>
<i>teupok</i>	<i>['teu', 'pok']</i>	<i>['teu', 'pok']</i>
<i>rakan</i>	<i>['ra', 'kan']</i>	<i>['ra', 'kan']</i>

Pada tabel 4.2 berisi sampel hasil data *test*, yang mencakup hasil silabifikasi dari *ground truth* dan hasil prediksi dari tahap pengujian. Tabel ini memberikan gambaran tentang hasil yang didapatkan berdasarkan model yang sudah dikembangkan. Namun, dalam tabel tersebut hanya berisi sampel kata yang berhasil disilabifikasi dengan benar sesuai dengan hasil silabifikasi dari *ground truth*. Informasi terkait hasil data *test* yang tidak berhasil disilabifikasi dengan benar dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Data *Test* yang Salah disilabifikasi

Kata	Prediksi	Hasil Silabifikasi
<i>adapt</i>	<i>['a', 'dap', 't']</i>	<i>['a', 'dapt']</i>
<i>basa denmark</i>	<i>['ba', 'sa', 'den', 'mar', 'k']</i>	<i>['ba', 'sa', 'den', 'mark']</i>
<i>puisi</i>	<i>['pui', 'si']</i>	<i>['pu', 'i', 'si']</i>
<i>meu 'èn ski</i>	<i>['meu', ' 'èn', ' 's', 'ki']</i>	<i>['meu', ' 'èn', 'ski']</i>

Kata	Prediksi	Hasil Silabifikasi
<i>meusr'ap</i>	<i>['meus', 'r'ap']</i>	<i>['meu', 'sr'ap']</i>
<i>èkspor</i>	<i>['èk', 's', 'por']</i>	<i>['èks', 'por']</i>
<i>gapu gips</i>	<i>['ga', 'pu', 'gip', 's']</i>	<i>['ga', 'pu', 'gips']</i>
<i>basa dairah</i>	<i>['ba', 'sa', 'dai', 'rah']</i>	<i>['ba', 'sa', 'da', 'i', 'rah']</i>
<i>homosèks</i>	<i>['ho', 'mo', 'sèk', 's']</i>	<i>['ho', 'mo', 'sèks']</i>
<i>peurangeui</i>	<i>['peu', 'ra', 'ngeu', 'i']</i>	<i>['peu', 'ra', 'ngeui']</i>
<i>ekspidisi</i>	<i>['ek', 's', 'pi', 'di', 'si']</i>	<i>['eks', 'pi', 'di', 'si']</i>

Berdasarkan pada Tabel 4.3, dapat dilihat bahwa terdapat sejumlah kata yang tidak berhasil disilabifikasi dengan benar. Hasil ini tidak hanya memberi gambaran tentang seberapa baik model yang dibangun akan memprediksi dengan benar, tetapi juga untuk mengidentifikasi beberapa kata yang merupakan titik lemah dalam model ini. Sehingga dapat menambah pemahaman yang lebih baik tentang sejauh mana model silabifikasi memprediksi dengan benar dan di mana yang perlu dilakukan perbaikan.

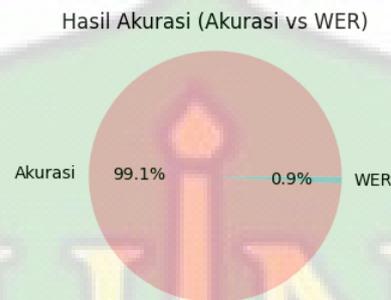
### 4.3 Evaluasi Hasil Pengujian

Evaluasi Hasil Pengujian merupakan tahap akhir dalam mengukur kinerja model yang sudah dikembangkan. Sehingga hasil evaluasi ini juga menghasilkan angka-angka yang mewakili tingkat akurasi model dalam memprediksi silabifikasi kata dalam Bahasa Aceh. Nilai akurasi tidak hanya membantu dalam mengukur performa model, tetapi juga memberikan gambaran yang jelas tentang sejauh mana model berhasil dalam proses silabifikasi.

Berdasarkan implementasi model pada data *test* sebanyak 1.245, terdapat 11 kata yang salah diprediksi. WER yang dihasilkan dari nilai tersebut dapat dilihat pada perhitungan berikut ini.

$$\begin{aligned}
 \text{Persamaan WER} &= \frac{\text{Jumlah kata yang salah}}{\text{Total jumlah kata}} \\
 &= \frac{11}{1.245} \\
 &= 0,009
 \end{aligned}$$

Hasil perhitungan menunjukkan bahwa nilai WER sebesar 0,009, setara dengan 0,9% jika dijadikan persentase. Dengan demikian, keakuratan model yang diperoleh dari akurasi mencapai 99,1%. Hasil dari akurasi diperoleh dari perhitungan dengan mengurangkan nilai WER dari 100%. Visualisasi hasil akurasi dan WER dapat ditemukan pada Gambar 4.1.



Gambar 4.1 Grafik Hasil Akurasi

Berdasarkan hasil akurasi dan WER seperti pada grafik yang terlihat pada Gambar 4.1, dapat diamati bahwa performa model tergambar dengan jelas. Grafik tersebut menunjukkan dua nilai utama, yaitu akurasi dan Word Error Rate (WER). Akurasi mencerminkan sejauh mana model dapat memprediksi dengan benar, sementara WER memberikan informasi mengenai kinerja model terkait kesalahan kata.

Penelitian ini merujuk pada pendekatan metodologi *Rule-Based* yang telah digunakan dalam penelitian sebelumnya, termasuk penelitian yang dilakukan oleh (Rahman et al., 2021) dengan judul “Silabifikasi Menggunakan Metode *Rule-Based* dalam Bahasa Indonesia”. Penelitian ini memperluas implementasi metode *Rule-Based* pada konteks Bahasa Aceh, berbeda dengan penelitian sebelumnya yang lebih memfokuskan pada Bahasa Indonesia atau bahasa lainnya. Berdasarkan hasil evaluasi, dapat disimpulkan bahwa model silabifikasi dengan menerapkan metode *Rule-Based* dalam Bahasa Aceh berhasil dilakukan, dengan memperoleh nilai WER sebesar 0,9%. Hal ini menunjukkan bahwa model memiliki kinerja yang sangat baik dalam memprediksi silabifikasi dengan tingkat kesalahan yang sangat rendah dan menegaskan bahwa penerapan metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh dapat menghasilkan kinerja yang lebih baik dibandingkan dengan

penelitian sebelumnya, karena disebabkan oleh adanya beberapa aturan silabifikasi yang sudah ditambahkan. Hal ini memberikan kontribusi yang signifikan terhadap pelestarian Bahasa Aceh dan pemahaman yang lebih mendalam tentang cara membaca kata dalam Bahasa Aceh.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil pembahasan dan penelitian yang telah diuraikan pada bab sebelumnya, dapat disimpulkan bahwa:

1. Implementasi metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh melibatkan penerapan beberapa aturan yang telah ditentukan untuk membagi suatu kata menjadi suku kata. Proses implementasi ini memungkinkan pemahaman tentang cara membaca suatu kata dalam Bahasa Aceh semakin meningkat. Dalam hasil penelitian ini, dapat disimpulkan bahwa implementasi metode *Rule-Based* pada proses silabifikasi dalam Bahasa Aceh telah berhasil dilakukan.
2. Tingkat akurasi pada proses silabifikasi dalam Bahasa Aceh dengan menerapkan metode *Rule-Based* telah memperoleh tingkat akurasi yang sangat baik. Berdasarkan hasil evaluasi, model silabifikasi yang menggunakan metode *Rule-Based* telah menghasilkan nilai WER sebesar 0,9%. Hasil ini menunjukkan bahwa model mampu memprediksi dengan sangat baik dan menghasilkan tingkat kesalahan yang sangat rendah.

#### 5.2 Saran

Berdasarkan kesimpulan dari hasil dan pembahasan pada penelitian ini, maka penulis mengemukakan beberapa saran yang dapat digunakan untuk pengembangan lanjutan dari penelitian ini, sebagai berikut:

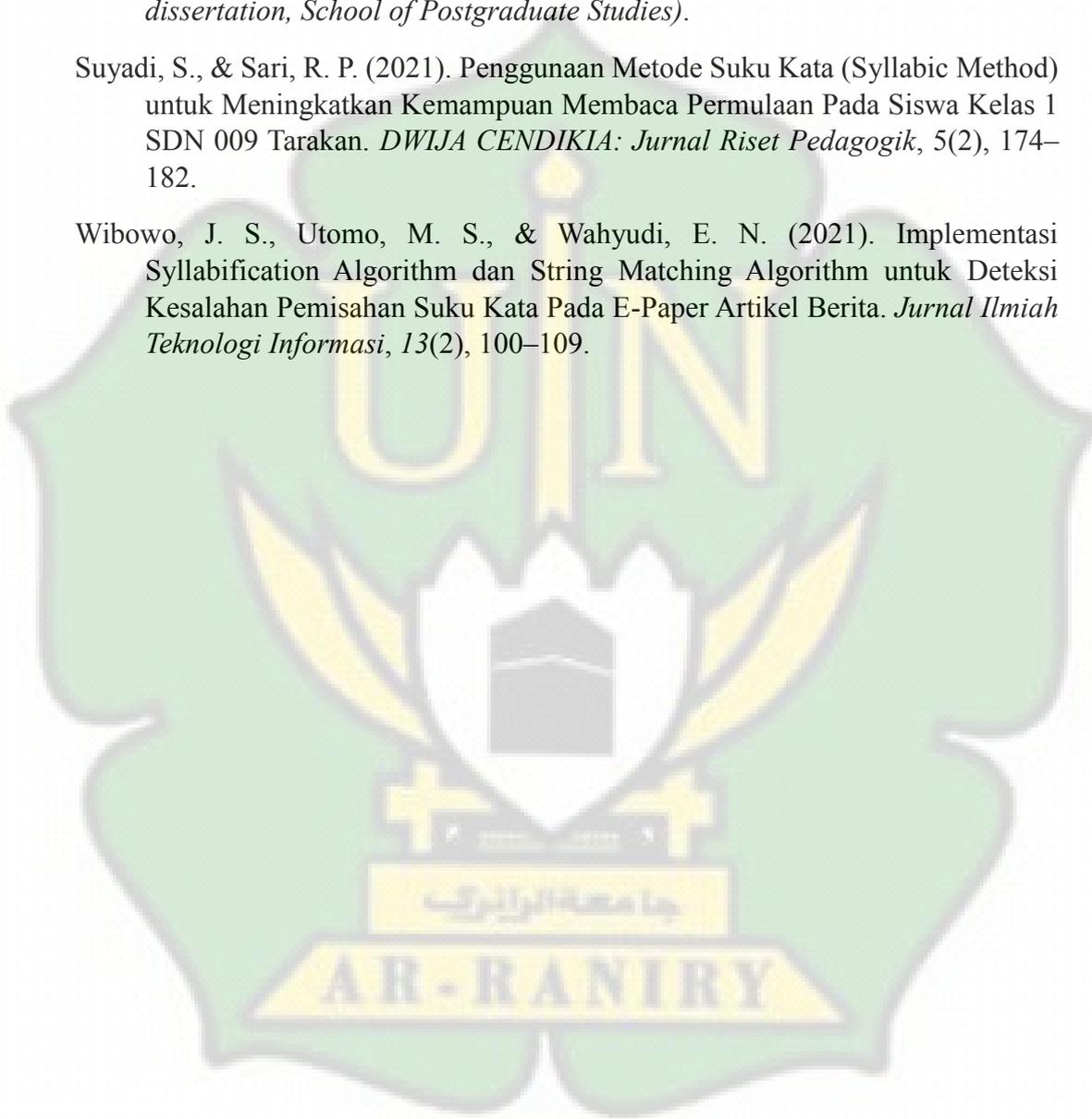
1. Dari hasil penelitian ini, didapatkan beberapa kata yang tidak berhasil disilabifikasi dengan benar dan beberapa kata tersebut menjadi fokus utama dalam upaya perbaikan model. Dengan memahami jenis kesalahan yang terjadi, penelitian selanjutnya dapat mengembangkan aturan tambahan atau metode perbaikan yang akan meningkatkan akurasi model silabifikasi.
2. Penelitian ini menggunakan 1.245 data *test*, diharapkan pada penelitian berikutnya untuk meningkatkan jumlah data *test*.

## DAFTAR PUSTAKA

- Alamsyah, T., Taib, R., Azwardi, N., & Idham, M. (2016). Pemilihan Bahasa Indonesia sebagai Bahasa Pertama Anak dalam Keluarga Masyarakat Aceh Penutur Bahasa Aceh di Nanggroe Aceh Darussalam. *Jurnal Pendidikan Bahasa Melayu*, 1(2), 31–44.
- Arifin, A., Haryanto, H., & Pratama, B. (2019). Segmentasi Suku Kata Bahasa Indonesia Berbasis Aturan untuk Pengembangan Sistem Text-To-Audiovisual. *SNATIF*, 5(2), 79–86.
- Aritonang, T. K. (2022). Pengenalan Algoritma Pada Pembelajaran Pemrograman Komputer.
- Asyi, Y. A. Q. A. (2020). The History of Aceh: Mengenal Asal Usul Nama, Bahasa, dan Orang Aceh. *Yayasan PeNA Banda Aceh*.
- Bartlett, S., Kondrak, G., & Cherry, C. (2008). Automatic Syllabification with Structured SVMs for Letter-To-Phoneme Conversion. *Proceedings of ACL-08: HLT*, 568–576.
- Daqiqil, I. (2021). Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python (Vol. 1). *Unri Press*.
- Enterprise, J. (2019). Python untuk Programmer Pemula. *Elex Media Komputindo*.
- Hanafiah, M. A., & Makam, I. (1984). Struktur Bahasa Aceh.
- Humami, A. G. (2022). Aplikasi Penentuan Kedudukan Kata Bahasa Arab Menggunakan Metode Rule-Based. *Doctoral Dissertation, Universitas Islam Negeri Maulana Malik Ibrahim*.
- Ilham, R. (2023). *Implementasi Algoritma Levenshtein Distance Untuk Fitur Pencarian Kata Pada Pengembangan Kamus Bahasa Aceh Berbasis Web* (Doctoral dissertation, UIN Ar-Raniry).
- Kurniawan, D. (2022). Pengenalan Machine Learning dengan Python. *Elex Media Komputindo*.
- Kusuma, P. D. (2020). Machine Learning Teori, Program, dan Studi Kasus. *Deepublish*.
- Laksono, Z. D. (2020). Ekstraksi Identitas Buku Berbasis Aturan. *Doctoral Dissertation, Universitas Komputer Indonesia*.
- Mahastama, A. W. (2022). Aplikasi Bantu Belajar Menulis Aksara Jawa untuk Siswa Sekolah Dasar. *Seminar Nasional Teknik Elektro, Informatika Dan Sistem Informasi*, 1(1).

- Mahastama, A. W. (2022). Model Berbasis Aturan untuk Transliterasi Bahasa Jawa dengan Aksara Latin ke Aksara Jawa. *Jurnal Buana Informatika*, 13(2), 146–154.
- Martono, K. T. (2011). Augmented Reality sebagai Metafora Baru dalam Teknologi Interaksi Manusia dan Komputer. *Jurnal Sistem Komputer*, 1(2), 60–64.
- Mulyani, E., Ismantohadi, E., & Koriah. (2020). Sistem Prediksi Potensi Drop Out Mahasiswa Menggunakan Rule Based System Pada Jurusan Teknik Informatika Politeknik Negeri Indramayu. *Informatika*, 8(1), 19–25.
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, 5(2), 697-711.
- Putra, G. B. P., & ERa, N. A. S. (2019). Syllabification of Balinese Words Using the Syllabification Algorithm. *Jurnal Elektronik Ilmu Komputer Udayana*, 8(2), 125–128.
- Rahma, S. (2020). Penuturan Bahasa Aceh di Kalangan Masyarakat Sibreh Kecamatan Sukamakmur. *Doctoral Dissertation, UIN Ar-Raniry*.
- Rahman, A., Murdiansyah, D. T., & Lhaksana, K. M. (2021). Silabifikasi Menggunakan Metode Rule-Based dalam Bahasa Indonesia. *EProceedings of Engineering*, 8(5).
- Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D. (2021). Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once). *Jurnal Nasional Ilmu Komputer*, 2(3), 213–232.
- Ratama, N., Kom, M., & Munawaroh, M. Kom. (2019). Konsep Kecerdasan Buatan dengan Pemahaman Logika Fuzzy dan Penerapan Aplikasi. *Uwais Inspirasi Indonesia*.
- Rizki, A., Junaidi, T. (2020). Pengantar Pembelajaran Bahasa Daerah Aceh.
- Retnoningsih, E., & Pramudita, R. (2020). Mengenal Machine Learning dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python. *Bina Insani Ict Journal*, 7(2), 156–165.
- Santoso, B. (2006). Bahasa dan Identitas Budaya. *Sabda: Jurnal Kajian Kebudayaan*, 1(1), 44–49.
- Shabihah, S. F. (2019). Safedio : Deteksi Konten Youtube Berunsur Pornografi dengan Judul yang Mengandung Bahasa Gaul (Alay) Menggunakan Pendekatan Query Expansion dan Rule Based.
- Siahaan, M., Jasa, C. H., Anderson, K., Valentino, M., Lim, S., & Yudianto, W. (2020). Penerapan Artificial Intelligence (AI) Terhadap Seorang Penyandang Disabilitas Tunanetra. *Journal Of Information System And Technology*, 1(2), 186-193.

- Subhayni, Armia, & Nurrahmah. (2020). Restrukturisasi Sapaan Kekerabatan Bahasa Aceh Sebagai Pendidikan Strategi Tutur Sapa Bagi Kaum Muda Aceh. *Jurnal Serambi Ilmu*, 21(1), 118–130.
- Sulistiyo, B., Surarso, B., & Syafei, W. A. (2020). Sistem Pakar Identifikasi dan Alternatif Solusi terhadap Permasalahan yang Dihadapi oleh Peserta Didik Sekolah Menengah Menggunakan Rule-Based Machine Learning. (*Doctoral dissertation, School of Postgraduate Studies*).
- Suyadi, S., & Sari, R. P. (2021). Penggunaan Metode Suku Kata (Syllabic Method) untuk Meningkatkan Kemampuan Membaca Permulaan Pada Siswa Kelas 1 SDN 009 Tarakan. *DWIJA CENDIKIA: Jurnal Riset Pedagogik*, 5(2), 174–182.
- Wibowo, J. S., Utomo, M. S., & Wahyudi, E. N. (2021). Implementasi Syllabification Algorithm dan String Matching Algorithm untuk Deteksi Kesalahan Pemisahan Suku Kata Pada E-Paper Artikel Berita. *Jurnal Ilmiah Teknologi Informasi*, 13(2), 100–109.



## LAMPIRAN

Adapun lampiran lengkap hasil dari penelitian ini sebagai berikut:

### Lampiran 1. *Source Code* untuk *Load Dataset*

*Code* berikut merupakan program *python* yang bertujuan untuk membaca *dataset* Bahasa Aceh yang terdapat dalam *file* teks dengan nama 'dataset.txt'. Dalam *code* ini, *file* dibuka dalam mode baca ('r') dan mengambil seluruh baris teks dari *file* tersebut. Hasilnya disimpan dalam variabel 'data', yang selanjutnya dapat digunakan untuk memproses atau menganalisis *dataset* tersebut dalam Bahasa Aceh.

```
data = 'dataset.txt'  
with open(data, 'r') as file:  
    data = file.readlines()
```

### Lampiran 2. *Source Code* untuk *Splitting Dataset*

Berikut *code* untuk mengimpor *library* dan modul yang dibutuhkan untuk *splitting dataset*.

```
# Mengimpor modul 'train_test_split' dari pustaka scikit-learn  
(sklearn) untuk membagi dataset menjadi data train dan test  
# Mengimpor modul 'numpy' untuk mendukung operasi yang terkait  
dalam pengelolaan data  
from sklearn.model_selection import train_test_split  
import numpy as np
```

Dalam *code* di bawah ini, *dataset* dibagi menjadi dua bagian, yaitu data *train* dan data *test* dengan menggunakan metode pembagian acak (*random splitting*) dengan rasio 80:20. Selain itu, *code* di bawah ini juga menampilkan jumlah keseluruhan *dataset*, data *training*, dan data *testing*.

```

data_train, data_test = train_test_split(data, test_size=0.2,
shuffle=True)
print(f"Jumlah keseluruhan dataset: {len(data)}")
print(f"Jumlah data training: {len(data_train)}, Jumlah data
testing: {len(data_test)}")

```

Data *training* dan data *testing* yang sudah dibagi akan disimpan ke dalam *file* baru dengan format teks.

```

# Menampilkan data train
print(data_train)
# Menyimpan data train ke dalam format .txt
train = 'train.txt'
with open(train, 'w') as file:
    for line in data_train:
        file.write(line)

# Menampilkan data test
print(data_test)
# Menyimpan data test ke dalam format .txt
test = 'test.txt'
with open(test, 'w') as file:
    for line in data_test:
        file.write(line)

```

### Lampiran 3. Source Code untuk Model

Code di bawah ini adalah definisi kelas ‘AcehneseSyllabificationModel’. Kelas ini memiliki metode konstruktor ‘\_\_init\_\_’, yang menginisialisasi beberapa daftar karakter vokal, konsonan, diftong, kombinasi konsonan, dan kata-kata khusus dalam Bahasa Aceh. Kelas ini didesain untuk pemodelan silabifikasi dalam Bahasa Aceh.

```

class AcehneseSyllabificationModel:
    def __init__(self):
        self.vocal = ['a', 'i', 'u', 'e', 'è', 'é', 'o', 'ô',
'ö', 'ʼa', 'ʼi', 'ʼu', 'ʼe', 'ʼè', 'ʼé', 'ʼo', 'ʼö']

```

```

        self.consonant = ['b', 'c', 'd', 'f', 'g', 'h', 'j',
                          'k', 'l', 'm', 'n', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x',
                          'y', 'z']
        self.diftong = ['ai', 'ai', 'au', 'ie', 'ie', 'ue',
                        'ue', 'eu', 'eu', 'eu', 'ee', 'èe', 'èe', 'ée', 'èè', 'èè',
                        'ée', 'oe', 'ôe', 'öe', 'ei', 'oi', 'ou', 'ôi', 'ôi', 'ui',
                        'eue', 'eue', 'éue']
        self.consonant_combination = ['ng', 'ny', 'sy', 'ph',
                                       'th', 'ch', 'kh', 'bh', 'dh', 'jh', 'gh', 'lh', 'rh', 'pl',
                                       'cl', 'bl', 'gl', 'kl', 'pr', 'tr', 'cr', 'kr', 'br', 'dr',
                                       'jr', 'gr', 'khl']
        self.kata_khusus = ['ski', 'mark', 'gips', 'strôp',
                             'ngeui', 'dapt']

```

Dalam metode ‘syllabify’, sebuah daftar suku kata diinisialisasi untuk menyimpan suku kata dari data yang akan diproses. Selain itu, variabel ‘i’ digunakan untuk melacak indeks karakter saat memproses huruf-huruf dalam kata, dan ‘length’ adalah panjang kata yang akan diurai menjadi suku kata. Langkah-langkah selanjutnya akan mengisi daftar suku kata dengan suku kata yang ditemukan dalam kata pada *dataset*.

```

def syllabify(self, word):
    syllables = []
    i = 0
    length = len(word)

```

Code berikut adalah perulangan *while* yang bertujuan untuk memeriksa dan mengenali diftong dalam kata. Proses ini memungkinkan pengenalan suku kata yang mengandung diftong dalam kata-kata pada *dataset* Bahasa Aceh.

```

while i < length:
    # Rule: 4 vocal (diftong)
    if i + 4 <= length and word[i:i+4] in
self.diftong:
        if i == 0 or word[i-1] in self.vocal:
            syllables.append(word[i:i+4])
        elif i > 0 and word[i-1] == " ":
            syllables.append(word[i:i+4])
        else:
            syllables[-1] += word[i:i+4]
        i += 4

    # Rule: 4 vocal (diftong + diftong)
    elif (i + 4 <= length and

```

```

        word[i:i+2] in self.diftong and
        word[i:i+4][2:4] in self.diftong):
    syllables[-1] += word[i:i+2]
    syllables.append(word[i:i+4][2:4])
    i += 4

    # Rule: 3 vocal (diftong)
    elif i + 3 <= length and word[i:i+3] in
self.diftong:
        if i == 0 or word[i-1] in self.vocal:
            syllables.append(word[i:i+3])
        elif i == 0 or word[i-1] in self.consonant:
            syllables[-1] += word[i:i+3]
        elif i > 0 and word[i-3] == " ":
            syllables.append(word[i:i+3])
        else:
            syllables.append(word[i:i+3])
        i += 3

    # Rule: 3 vocal (diftong + vocal)
    elif (i + 3 <= length and
        word[i:i+2] in self.diftong and
        word[i:i+3][2] in self.vocal and
        word[i:i+3][2] not in self.diftong):
        syllables[-1] += word[i:i+2]
        syllables.append(word[i+2:i+3])
        i += 3

    # Rule: 3 vocal (vocal + diftong)
    elif (i + 3 <= length and
        word[i] in self.vocal and
        word[i+1:i+3] in self.diftong):
        syllables[-1] += word[i]
        syllables.append(word[i+1:i+3])
        i += 3

    # Rule: 2 vocal (diftong)
    elif i + 2 <= length and word[i:i+2] in
self.diftong:
        if i == 0 or word[i-1] in self.vocal:
            syllables.append(word[i:i+2])
        elif i == 0 or word[i-1] in self.consonant:
            syllables[-1] += word[i:i+2]
        elif i > 0 and word[i-2] == " ":
            syllables.append(word[i:i+2])
        else:
            syllables.append(word[i:i+2])

```

```

        i += 2

# Rule: 2 vocal (bukan diftong)
elif (i + 2 <= length and
      word[i:i+2] not in self.diftong and
      word[i:i+2][0] in self.vocal and
      word[i:i+2][1] in self.vocal):
    if i == 0:
        syllables.append(word[i:i+2][0])
        syllables.append(word[i:i+2][1])
    else:
        syllables[-1] += word[i:i+2][0]
        syllables.append(word[i:i+2][1])
    i += 2

# Rule: 1 vocal
elif word[i] in self.vocal:
    if i > 0 and word[i-1] == " ":
        syllables.append(word[i])
    elif i == 0 or word[i-1] in self.vocal:
        syllables.append(word[i])
    elif i < length - 1 and word[i+1] == " ":
        syllables[-1] += word[i]
    else:
        syllables[-1] += word[i]
    i += 1

```

Berikut adalah sambungan dari perulangan *while* di atas yang bertujuan untuk memeriksa dan mengenali jika suatu kondisi terdapat kata khusus, spasi, dan tanda kutip tunggal dalam kata. Proses ini memungkinkan untuk memisahkan dan mengidentifikasi suku kata dengan benar dalam kata-kata pada *dataset* Bahasa Aceh.

```

# Rule: special words
elif word in self.kata_khusus:
    syllables.append(word)
    i += len(word)

# Rule: Handle jika ada tanda kutip tunggal (')
elif word[i] == "'":
    if i > 0 and word[i-1] in self.consonant:
        syllables[-1] += word[i]
    else:
        syllables.append(word[i])
    i += 1

```

```

# Rule : Handle jika ada spasi
elif word[i] == " ":
    syllables.append(word[i])
    i += 1

```

Code di bawah ini juga merupakan sambungan dari perulangan *while* di atas yang bertujuan untuk memeriksa dan mengenali konsonan dalam kata. Proses ini memungkinkan pemenggalan suku kata menjadi lebih akurat dalam Bahasa Aceh dengan menerapkan aturan untuk konsonan.

```

# Rule: 4 consonants (consonant_combination +
consonant_combination)
elif (i + 4 <= length and
word[i:i+2] in self.consonant_combination
and
word[i:i+4][2:4] in
self.consonant_combination):
    syllables[-1] += word[i:i+2]
    syllables.append(word[i:i+4][2:4])
    i += 4

# Rule: 3 consonants (consonant_combination)
elif i + 3 <= length and word[i:i+3] in
self.consonant_combination:
    if i + 1 == length - 1 :
        syllables[-1] += word[i:i+3]
    else:
        syllables.append(word[i:i+3])
    i += 3

# Rule: 3 consonants (consonant_combination +
consonant)
elif (i + 3 <= length and
word[i:i+2] in self.consonant_combination
and
word[i:i+3][2] in self.consonant and
word[i:i+3][2] not in
self.consonant_combination):
    syllables[-1] += word[i:i+2]
    syllables.append(word[i+2:i+3])
    i += 3

# Rule: 3 consonants (consonant +
consonant_combination)
elif (i + 3 <= length and

```

```

        word[i] in self.consonant and
        word[i+1:i+3] in
self.consonant_combination):
        syllables[-1] += word[i]
        syllables.append(word[i+1:i+3])
        i += 3

        # Rule: 2 consonants (consonant_combination)
        elif i + 2 <= length and word[i:i+2] in
self.consonant_combination:
        if i + 1 == length - 1 :
            syllables[-1] += word[i:i+2]
        elif i < length - 1 and word[i+2] == " ":
            syllables[-1] += word[i:i+2]
        else:
            syllables.append(word[i:i+2])
            i += 2

        # Rule: 2 consonants (bukan consonant_combination)
        elif (i + 2 <= length and
            word[i:i+2] not in
self.consonant_combination and
            word[i:i+2][0] in self.consonant and
            word[i:i+2][1] in self.consonant):
        if i == 0:
            syllables.append(word[i:i+2])
        else:
            syllables[-1] += word[i:i+2][0]
            syllables.append(word[i:i+2][1])
            i += 2

        # Rule: 1 consonant
        elif word[i] in self.consonant:
            if i > 0 and word[i-1] == " ":
                syllables.append(word[i])
            elif i < length - 1 and word[i+1] == " ":
                syllables[-1] += word[i]
            elif i == length - 1:
                syllables[-1] += word[i]
            else:
                syllables.append(word[i])
            i += 1

        else:
            if not syllables:
                syllables.append(word[i])
            else:

```

```

        syllables[-1] += word[i]
        i += 1

    while ' ' in syllables:
        syllables.remove(' ')

    return syllables

```

Beberapa aturan di atas berperan penting dalam proses pemenggalan suku kata pada teks Bahasa Aceh. Aturan-aturan ini membantu mengidentifikasi dan memisahkan kata menjadi suku kata pada *dataset*, memastikan hasil pemenggalan suku kata yang akurat, dan memahami struktur suku kata Bahasa Aceh.

#### Lampiran 4. *Source Code* untuk Pelatihan Model

*Code* di bawah ini bertujuan untuk menginisialisasi kamus ‘syllabifications’ yang digunakan untuk menyimpan hasil pemenggalan suku kata pada *dataset* yang dibaca dari *file* teks ‘data\_silabifikasi.txt’. Setiap baris dalam *file* tersebut dipisahkan menjadi dua bagian, yaitu sebelum koma sebagai kata (*word*) dan setelah koma sebagai suku kata (*syllables*). Informasi pemenggalan suku kata ini disimpan dalam kamus ‘syllabifications’ dengan kata sebagai kunci dan suku kata sebagai nilainya. Proses ini memungkinkan untuk mengakses pemenggalan suku kata yang sesuai dengan kata tertentu dalam *dataset*.

```

# Inisialisasi kamus syllabifications
syllabifications = {}

# Baca dataset dari file .txt
with open('data_silabifikasi.txt', 'r') as file:
    for line in file:
        line = line.strip()
        if line:
            word, *syllables = line.split(", ")
            syllabifications[word] = syllables

```

*Code* di bawah ini bertujuan untuk mendapatkan pemenggalan suku kata yang benar (*ground truth*) dari kamus ‘syllabifications’ berdasarkan kata pada *dataset*

'dataset.txt'. Fungsi ini menggunakan kamus yang telah dibaca dari *file dataset* 'data\_silabifikasi.txt' dan mengembalikan pemenggalan suku kata yang sesuai dengan kata yang telah diberikan. Jika kata tidak ditemukan dalam kamus, maka fungsi akan mengembalikan nilai 'None'. Hal ini memungkinkan perbandingan hasil pemenggalan suku kata berdasarkan model dengan suku kata yang sebenarnya.

```
# Fungsi untuk mendapatkan silabifikasi dari kamus
def get_ground_truth_syllabification(word):
    return syllabifications.get(word, None)
```

Pada *code* di bawah bertujuan untuk menghitung WER (*Word Error Rate*) antara pemenggalan suku kata yang benar (*ground truth*) dan suku kata yang diprediksi oleh model. Fungsi ini mengukur kesalahan berdasarkan jumlah karakter yang tidak cocok. Hasil dari perhitungan ini adalah tingkat kesalahan (*error rate*) yang mencerminkan sejauh mana hasil pemenggalan suku kata yang diprediksi oleh model sesuai dengan pemenggalan suku kata yang sebenarnya dari *ground truth*.

```
# Menghitung nilai akurasi (WER)
def calculate_wer(ground_truth, predicted):
    len_gt = sum(len(gt) for gt in ground_truth)
    errors = 0 # Inisialisasi errors ke 0

    for gt, pred in zip(ground_truth, predicted):
        wer = 0 if gt == pred else 1
        errors = wer

    return wer
```

Berikut *code* untuk membaca *file* data 'train.txt' yang digunakan dalam proses pelatihan model.

```
# Load data
def load_data(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        data = file.readlines()
    return [line.strip() for line in data]
data = load_data("train.txt")
```

Berikut *code* yang digunakan untuk menginisialisasi model pemenggalan suku kata Bahasa Aceh.

```

# Inisialisasi model
model = AcehneseSyllabificationModel()

total_wer = 0 # Menghitung nilai total WER dari seluruh kata
yang diuji

print("-----")
print(f>Data Train\n-----\n")

```

Code di bawah ini digunakan untuk mendapatkan nilai WER dari proses pelatihan model pada data *train*. Hasil yang didapatkan adalah nilai WER dari setiap baris yang terdiri dari nilai 0 dan 1. Nilai 0 yang berarti sesuai, dan nilai 1 yang berarti tidak sesuai.

```

for line_num, word in enumerate(data, start=1):
    predicted = model.syllabify(word)
    ground_truth = get_ground_truth_syllabification(word)

    if ground_truth:
        wer = calculate_wer(ground_truth, predicted)
        total_wer += wer
        ...

        # Print semua hasil kata
        print(f"Line: {line_num}. Word: {word} - Ground Truth:
{ground_truth} - Predicted: {predicted} - WER: {wer}")

        ...

        # Kondisi di bawah hanya menampilkan kata yang nilai
akurasi tidak mencapai 0.0%
        if wer != 0.0:
            print(f"Line: {line_num}. Word: {word} - length:
{len(ground_truth)} - Ground Truth: {ground_truth} -
Predicted: {predicted} - WER: {wer}")

```

Pada bagian ini dilakukan perhitungan total nilai WER yang didapatkan berdasarkan hasil pengujian pada data latih. Hasil dari perhitungan ini memberikan gambaran tentang sejauh mana model telah berhasil dalam pemenggalan suku kata Bahasa Aceh pada data *train*.

```

# Calculate WER overall

print("\nWER = Jumlah kata yang salah / Total jumlah kata")

print("WER = ",total_wer,"/",line_num)

```

```

overall_average_wer_train = total_wer / line_num if line_num >
0 else float('inf')

print("\n-----
")

print(f"Nilai akurasi pada data train menggunakan WER:
{overall_average_wer_train}\n-----
\n")

```

### Lampiran 5. Source Code untuk Hasil Pengujian dan Evaluasi

Berikut *code* untuk membaca *file* data 'test.txt' yang digunakan dalam proses pengujian dan evaluasi.

```

# Load data
def load_data(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        data = file.readlines()
    return [line.strip() for line in data]
data = load_data("test.txt")

```

*Code* berikut berfungsi untuk menampilkan nilai WER dari setiap baris dan total nilai akursi WER yang didapatkan dari tahap pengujian dengan menggunakan data *test*. Hasil pengujian ini berguna untuk mengukur kinerja model dalam pemenggalan suku kata pada Bahasa Aceh.

```

# Inisialisasi model
model = AcehneseSyllabificationModel()

total_wer = 0 # Menghitung nilai total WER dari seluruh kata
yang diuji

print("-----")
print(f>Data test\n-----\n")

for line_num, word in enumerate(data, start=1):
    predicted = model.syllabify(word)
    ground_truth = get_ground_truth_syllabification(word)

    if ground_truth:

```

```

wer = calculate_wer(ground_truth, predicted)
total_wer += wer
'''
# Print semua hasil kata
print(f"Line: {line_num}. Word: {word} - Ground Truth:
{ground_truth} - Predicted: {predicted} - WER: {wer}")

'''
# Kondisi di bawah hanya menampilkan kata yang nilai
akurasi tidak mencapai 0.0%
if wer != 0.0:
    print(f"Line: {line_num}. Word: {word} - length:
{len(ground_truth)} - Ground Truth: {ground_truth} -
Predicted: {predicted} - WER: {wer}")

```

Code di bawah ini digunakan untuk mengevaluasi nilai WER dari tahap pengujian. Total WER dan nomor kata digunakan untuk menghitung keseluruhan nilai WER yang didapatkan dari data *test*. Hasil keseluruhan akurasi ini ditampilkan dalam bentuk persentase, seperti 0,01 menjadi 1%.

```

# Calculate WER overall
print("\nWER = Jumlah kata yang salah / Total jumlah kata")
print("WER = ",total_wer,"/",line_num)
overall_average_wer_test = total_wer / line_num if line_num >
0 else float('inf')
print("\n-----
")
print(f"Nilai akurasi pada data test menggunakan WER:
{overall_average_wer_test} ({overall_average_wer_test:.1%})\n-
-----\n")

```

Code di bawah ini bertujuan untuk mengimpor modul 'matplotlib' yang digunakan untuk menampilkan grafik akurasi yang diperoleh dari tahap pengujian. Data yang akan digunakan untuk plot disimpan dalam 2 *list*, yaitu 'labels', dan 'accuracy\_values'.

```

# Grafik Lingkaran (Pie Chart) untuk hasil menggunakan data
testing
import matplotlib.pyplot as plt

# Buat dua daftar untuk menyimpan label dan nilai akurasi
labels = ['Akurasi', 'WER']
accuracy_values = [1 - overall_average_wer_test,
overall_average_wer_test]

```

Code di bawah ini mengatur isi grafik untuk menampilkan hasil akurasi (akurasi dan WER). Grafik yang digunakan adalah grafik lingkaran (*pie chart*).

```
# Grafik
plt.figure(figsize=(6, 3))
plt.pie(accuracy_values, labels=labels, autopct='%1.1f%%',
        colors=['pink', 'lightblue'])
plt.title('Hasil Akurasi (Akurasi vs WER)')
plt.axis('equal')
```

Berikut code untuk menampilkan grafik.

```
# Menampilkan grafik
plt.show()
```

