PENDEKATAN BARU *STEMMING* KATA DAN AUGMENTASI DATA TEKS PADA BAHASA LAMPUNG SEBAGAI *LOW-RESOURCE LANGUAGE*

DISERTASI

Oleh

ZAENAL ABIDIN
NPM 2237061006

PROGRAM STUDI DOKTOR MIPA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

PENDEKATAN BARU *STEMMING* KATA DAN AUGMENTASI DATA TEKS PADA BAHASA LAMPUNG SEBAGAI *LOW-RESOURCE LANGUAGE*

Oleh

Zaenal Abidin

Disertasi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
DOKTOR MIPA

Pada

Program Pascasarjana
Doktor MIPA

PROGRAM STUDI DOKTOR MIPA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2026

ABSTRAK

PENDEKATAN BARU *STEMMING* KATA DAN AUGMENTASI DATA TEKS PADA BAHASA LAMPUNG SEBAGAI *LOW-RESOURCE LANGUAGE*

Oleh

ZAENAL ABIDIN

Penelitian ini mengembangkan pendekatan *natural language processing* (NLP) untuk bahasa Lampung berfokus pada *stemming* kata dialek Tulang Bawang dan *Lampung Language Dialect Identification* (LLDI) pada dialek *Api* dan *Nyo*. Untuk *stemming* kata, lima modifikasi metode diuji secara komparatif: modifikasi Nazief-Adriani (MNA), modifikasi *Confix-Stripping* (MCS), MCS disertai *N-Gram Stemming*, *Morphological-based*, dan *N-Gram Stemming*. Eksperimen pada 500 kata data uji dan 200 kata independen menunjukkan MCS disertai *N-Gram Stemming* mencapai performa terbaik dengan nilai *Gold Standard Assessment* (GSA) 98,8%. Pendekatan terbaik dari *stemming* kata diimplementasikan pada aplikasi *Direct Machine Translation* (DMT) Tulang Bawang–Indonesia dan hasil pengujian penerjemahan menunjukan nilai *Bilingual Evaluation Understudy* (BLEU) mencapai 80,07%. Untuk identifikasi dialek, model LLDI dibangun menggunakan *data set* 3000 kalimat dialek *Api* dan 9078 kalimat dialek *Nyo*. *Text Data Augmentation* (TDA) melalui metode permutasi kalimat diterapkan untuk mengatasi ketidakseimbangan data, menghasilkan *data set* sintesis sebesar *n*! dari *n token* sebuah kalimat. Empat pendekatan klasifikasi—*Naive Bayes, Logistic Regression, Support Vector Machine* (SVM), dan *Random Forest*—dievaluasi menggunakan *5-fold cross validation*. Hasil eksperimen menunjukkan SVM *balanced class* mencapai performa tertinggi dengan akurasi 97,4%, diikuti *Random Forest balanced class* dengan akurasi 96,9%. Penyeimbangan kelas terbukti meningkatkan deteksi dialek minoritas (*Api*) tanpa mengorbankan performa dialek mayoritas (*Nyo*). Sebaliknya, kondisi *unbalanced* menghasilkan *precision* tinggi namun *recall* rendah untuk dialek *Api*. Penelitian ini memberikan kontribusi awal yang signifikan dalam pengembangan *Natural Language Understanding* (NLU) pada bahasa Lampung dan berpotensi dilanjutkan pada penelitian fonologi, sintaksis, semantik dan pragmatik bahasa Lampung secara komputasi.

Kata Kunci : *Stemming*, Nazief-Adriani, *Confix-Stripping*, *N-Gram Stemming*, *Text Data Augmentation*, Permutasi, *Lampung Language Dialect Identification*

ABSTRACT

A NEW APPROACH TO WORD STEMMING AND TEXT DATA AUGMENTATION IN LAMPUNG LANGUAGE AS A LOW-RESOURCE LANGUAGE

By

ZAENAL ABIDIN

This study develops a natural language processing (NLP) approach for Lampung language focusing on stemming words in Tulang Bawang dialect and Lampung language dialect identification (LLDI) in *Api* and *Nyo* dialects. For word stemming, five approaches were tested comparatively: modified Nazief-Adriani (MNA), modified Confix-Stripping (MCS), MCS with N-Gram stemming, morphological-based, and N-Gram stemming. Experiments on 500 test words and 200 independent words showed that MCS with N-Gram stemming achieved the best performance with a Gold Standard Assessment (GSA) value of 98,8%. The best approach of word stemming was implemented in the Tulang Bawang-Indonesia Direct Machine Translation (DMT) application and the translation test results showed a bilingual evaluation understudy (BLEU) value of 80,07%. For dialect identification, the LLDI model was built using a dataset of 3000 *Api* dialect sentences and 9076 *Nyo* dialect sentences. Text data augmentation (TDA) using a sentence permutation approach was applied to address data imbalance, generating a synthetic dataset of n! from n sentence tokens. Four classification approaches—Naive Bayes, Logistic Regression, Support Vector Machine (SVM), and Random Forest—were evaluated using 5-fold cross-validation. Experimental results showed that the balanced SVM class achieved the highest performance with 97,4% accuracy, followed by the balanced Random Forest class with 96,9% accuracy. Class balancing was shown to improve the detection of the minority dialect (*Api*) without sacrificing the performance of the majority dialect (*Nyo*). Conversely, the unbalanced condition resulted in high precision but low recall for the Api dialect. This research provides a significant initial contribution to the development of natural language understanding (NLU) in Lampung and has the potential to be continued in computational research on the phonology, syntax, semantics, and pragmatics of Lampung.

Key Words : Stemming, Nazief-Adriani, Confix-Stripping, N-Gram Stemming, Text Data Augmentation, Permutasi, Lampung Language Dialect Identification.

Judul Disertasi	: PENDEKATAN BARU *STEMMING* KATA DAN AUGMENTASI DATA TEKS PADA BAHASA LAMPUNG SEBAGAI *LOW-RESOURCE LANGUAGE*
Nama Mahasiswa	: Zaenal Abidin
NPM	: 2237061006
Program Studi	: Doktor MIPA
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

Bandar Lampung, 30 Januari 2026

MENYETUJUI

1. Komisi Pembimbing

Prof. Wamiliana, Ph.D.
NIP. 196311081989022001

.....

Dr.rer.nat. Akmal Junaidi, M.Sc.
NIP. 197101291997021001

.....

Favorisen R. Lumbanraja, M.Si., Ph.D.
NIP. 198301102008121002

.....

2. Ketua Program Studi Doktor MIPA

Dr. Khoirin Nisa, S.Si., M.Si.
NIP 197407262000032001

PENGESAHAN PENGUJI

PENDEKATAN BARU *STEMMING* KATA DAN AUGMENTASI DATA TEKS PADA BAHASA LAMPUNG SEBAGAI *LOW-RESOURCE LANGUAGE*

DISERTASI

OLEH
ZAENAL ABIDIN
NPM 2237061006

Tim Penguji

Jabatan	Nama	Tanda Tangan
Ketua	: Dr. Eng. Heri Satria, S.Si., M.Si. NIP 197110012005011002	.....
Sekretaris	: Dr. Khoirin Nisa, S.Si., M.Si. NIP 197407262000032001	.....
Anggota	: Prof. Wamiliana, Ph.D. NIP 196311081989022001	.....
	: Dr. rer. nat. Akmal Junaidi, M.Sc. NIP 197101291997021001	.....
	: Favorisen R. Lumbanraja, M.Si., Ph.D. NIP 198301102008121002	.....
	: Dr. Dian Kurniasari, S.Si., M.Sc. NIP 196903051996032001	.....
	: Prof. Dr. Farida Ariyani, M.Pd. NIP 196012141984032002	.....
	: Samsuryadi, S.Si., M.Kom., Ph.D. NIP 197102041997021003	.....

Dekan FMIPA UNILA,

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP 197110012005011002

Direktur Pascasarjana,

Prof. Dr. Ir. Murhadi, M.Si
NIP 196403261989021001

Tanggal Lulus Ujian Disertasi : 30 Januari 2026

PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa disertasi dengan judul "Pendekatan Baru *Stemming* Kata dan Augmentasi Data Teks pada Bahasa Lampung sebagai *Low-Resource Language*" beserta seluruh isinya adalah benar-benar hasil karya sendiri, dan saya tidak melakukan plagiarisme atau pengutipan dengan cara-cara yang tidak sesuai dengan etika yang berlaku pada masyarakat keilmuan. Atas pernyataan ini, saya siap menerima sanksi atau tindakan yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan pelanggaran terhadap etika akademik dalam karya saya ini, atau klaim dari pihak lain terhadap keaslian karya saya ini.

Bandar Lampung, 30 Januari 2026

Penulis,

Zaenal Abidin

RIWAYAT HIDUP

Penulis dilahirkan di kota Serang, 10 Juli 1981. Penulis anak pertama dari 7 bersaudara. Putra pertama dari Bapak H. Abdulloh Bajawi (Alm) dan Ibu Hj. Juhaenah (Almh). Penulis menempuh pendidikan di Madrasah Aliyah Manbaul Ulum Asshiddiqiyah Jakarta Barat, lulus tahun 1999. Pada tahun 2005, penulis lulus S1 dari Jurusan Matematika FMIPA Unila, kemudian tahun 2014 penulis lulus S1 dari Teknik Informatika STMIK Teknokrat serta lulus S2 Informatika dari Sekolah Teknik Elektro dan Informatika (STEI) ITB Bandung tahun 2018. Pada tahun 2022, penulis diterima sebagai mahasiswa pascasarjana di program studi Doktor MIPA Universitas Lampung. Selama studi S3, penulis aktif melakukan penelitian tentang NLP bahasa Lampung.

Bandar Lampung, 30 Januari 2026
Penulis,

Zaenal Abidin

PERSEMBAHAN

Disertasi ini dipersembahkan untuk masyarakat Lampung yang peduli akan eksistensi bahasa Lampung termasuk masyarakat yang peduli akan penelitian bahasa Lampung pada aspek komputasi seluas-luasnya. Disertasi ini untuk Putraku, Putriku dan Istriku. Disertasi ini juga untuk "Alm *Abah*, Almh *Mamah* tercinta sampai akhir hayatku, *Teh Iyah Om Rohman, Teh Enjum Om Lulu, Teh Ebed Om Bambang, Aa Choenk* dan istri, *Aa Ade* dan istri, Teh Iis Om Jaka. Semua adik-adikku: Imat, Almh Nurul Jannah, Iim, Qori, Iko, Mia dan *dulur kule sedantene ning* Banten". Disertasi ini juga untuk kampus Teknokrat, sekaligus ini wujud kesungguhan masa S3 yang luar biasa.

UCAPAN TERIMA KASIH

Puji syukur penulis ucapkan kehadirat Alloh *Azza wa Jalla*, karena atas rahmat-Nya dan hidayah-Nya disertasi ini dapat diselesaikan. Disertasi dengan judul "Pendekatan Baru *Stemming* Kata dan Augmentasi Data Teks pada Bahasa Lampung sebagai *Low-Resource Language*" adalah salah satu syarat untuk memperoleh gelar Doktor pada program studi Doktor MIPA di Universitas Lampung.

Penulis menyampaikan penghargaan dan terima kasih yang setinggi-tingginya kepada:

1. Ibu Prof. Wamiliana, Ph.D. selaku Promotor dan sekaligus pembimbing akademik yang telah membimbing, memberi arahan, motivasi secara intensif selama proses penyusunan disertasi hingga selesai.
2. Bapak Dr. rer.nat. Akmal Junaidi, M.Sc. selaku Ko-Promotor 1 yang telah memberi bimbingan, arahan, motivasi secara intensif selama penulis menjalani S3 di ruang lantai 2 MIPA Terpadu.
3. Bapak Favorisen R. Lumbanraja, M.Si., Ph.D. selaku Ko-Promotor 2 yang telah memberi bimbingan, masukan, kritik, saran yang sangat membangun guna penyempurnaan disertasi ini menjadi lebih baik dan berkualitas.
4. Ibu Dr. Dian Kurniasari, M.Sc. selaku penguji internal 1 yang telah memberi arahan terkait konsistensi penulisan, alur tulisan penelitian ini menjadi lebih baik.
5. Ibu Prof. Dr. Farida Ariyani, M.Pd. selaku penguji internal 2 yang telah memberikan pemikiran dan arahan dari aspek kajian bahasa Lampung pada penelitian ini.
6. Bapak Samsuryadi, M.Kom., Ph.D. selaku penguji eksternal dari Universitas Sriwijaya yang telah meluangkan waktu untuk mengevaluasi dan memberikan saran perbaikan untuk penelitian ini.
7. Ibu Dr. Khoirin Nisa, S.Si., M.Si. selaku kaprodi S3 MIPA FMIPA yang telah memberi dorongan kepada mahasiswa S3 termasuk ke penulis agar selesai tepat waktu.
8. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.

9. Bapak Prof. Dr. Ir. Murhadi, M.Si. selaku Direktur Pascasarjana Universitas Lampung.
10. Yayasan Pendidikan Teknokrat yang telah mendukung secara penuh pelaksanaan studi S3 penulis dari awal sampai wisuda kelak.
11. Rekan-rekan dosen di ruang ICT C atas tambahan motivasi agar penulis segera menyelesaikan S3 ini.
12. Rekan-rekan semua mahasiswa S3 MIPA angkatan 2022 yang telah memberikan warna selama menjalani perkuliahan, khususnya tim Ilmu Komputer Bapak Agus, Ibu Sri, Ibu Apri.

Bandar Lampung, 30 Januari 2026

Zaenal Abidin

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR SINGKATAN / ISTILAH

PBA	Pemrosesan Bahasa Alami
NLP	*Natural Language Processing*
TB	Tulang Bawang
MT	*Machine Translation*
DMT	*Direct Machine Translation*
WS	*Word Stemming*
SMT	*Statistical Machine Translation*
NMT	*Neural Machine Translation*
TDA	*Text Data Augmentation*
NLU	*Natural Language Understanding*
NLG	*Natural Language Generation*
MNA	Modifikasi Nazief-Adriani
MCS	Modifikasi *Confix-Stripping*
GSA	*Gold Standar Asessment*
BLEU	*Bilingual Evaluation Understudy*
DI	*Dialect Identification*
LLDI	*Lampung Language Dialect Identification*
POS	*Part of Speech*
AI	*Artificial Intelligence*
OOV	*Out of Verb*
CS	*Confix-Stripping*
ECS	*Enhanced Confix-Stripping*
IR	*Information Retrieval*
LSTM	*Long Short Term Memory*
SP	Subjek Predikat
SPO	Subjek Predikat Objek
SPOK	Subjek Predikat Objek Keterangan

I. PENDAHULUAN

1.1 Latar Belakang Masalah

Natural language processing (NLP) merupakan bidang yang menggabungkan *artificial intelligence* (AI) dan linguistik, yang bertujuan untuk membuat komputer memahami pernyataan atau kata-kata yang ditulis dalam bahasa manusia baik pada *high-resource language* maupun *low-resource language* (Khurana et al., 2023; Pakray et al., 2025). Linguistik adalah ilmu bahasa yang mencakup fonologi yang berkaitan dengan bunyi, morfologi tentang pembentukan kata, sintaksis terkait struktur kalimat, semantik dan pragmatik yang berkaitan dengan makna atau pemahaman suatu kalimat (Khurana et al., 2023). NLP dikembangkan untuk memudahkan pekerjaan *user* dan memenuhi keinginan *user* untuk berkomunikasi dengan komputer menggunakan bahasa alami. Teknologi NLP telah diterapkan dan menjadi bagian yang tak terpisahkan dalam kehidupan manusia, memfasilitasi komunikasi antara komputer dan manusia dalam berbagai bidang, seperti *machine translation* (MT), deteksi *spam email*, ekstraksi informasi, ringkasan, dan sistem tanya jawab, dan sebagainya (Jabbar et al., 2023; Khurana et al., 2023; Pakray et al., 2025; Zaiton & Alansary, 2025; Zampieri et al., 2020).

Berkat ketersediaan data dan sumber daya linguistik yang melimpah, aplikasi NLP pada bahasa dengan sumber daya linguistik yang melimpah atau *high-resource language* mencakup berbagai tugas dan bidang (Khurana et al., 2023; Pakray et al., 2025). Namun, pengembangan aplikasi NLP untuk bahasa dengan sumber daya linguistik yang terbatas atau *low-resource language* menghadirkan sejumlah tantangan, seperti keterbatasan akses terhadap data dan sumber daya digital, struktur linguistik yang kompleks, serta kurangnya *data set* yang teranotasi dan evaluasi standar (Pakray et al., 2025). Ada kebutuhan untuk pengembangan aplikasi NLP pada kategori *low-resource language* guna melestarikan keragaman linguistik, memberdayakan komunitas, mendukung pendidikan, dan meningkatkan

komunikasi pada berbagai bahasa daerah di Indonesia. Bahasa Lampung termasuk salah satu kategori *low-resource language* dan penelitian NLP ini difokuskan pada bahasa Lampung.

Masalah utama yang ditemukan pada penelitian ini dengan objek bahasa Lampung sebagai *low-resource language* adalah (1) minimnya ketersediaan *data set* digital bahasa Lampung baik dalam kalimat dialek *Api* maupun *Nyo*, (2) minimnya acuan sumber kajian ilmiah bagi ragam dialek yang ada di provinsi Lampung untuk dialek tertentu, (3) belum tersedianya kamus digital yang mengakomodasi ragam dialek yang ada di provinsi Lampung, (4) minimnya bahan kajian secara digital pada aspek linguistik bahasa Lampung yang komprehensif terutama membahas beberapa aspek bahasa Lampung seperti *stemming* kata, sinonim, antonim kata bahasa Lampung, klasifikasi kata atau *part of speech* (POS) *tag* yang valid pada kata bahasa Lampung, deteksi *error* pada *grammar* bahasa Lampung secara sintaksis, kalimat aktif dan kalimat pasif dalam bahasa Lampung. Fokus utama penelitian ini pada *stemming* dan augmentasi data teks berdasarkan *data set* digital yang tersedia pada bahasa Lampung khususnya pada dialek *Api*.

Salah satu *task* pada bidang NLP adalah *machine translation*. Penelitian *machine translation* (MT) Lampung – Indonesia telah dilakukan sejak 2017, dibuktikan melalui hasil tesis penulis pada program studi magister informatika ITB tahun 2018 berjudul "Perbandingan Kinerja *Direct* dan *Data Driven Machine Translation* untuk Bahasa Lampung-Indonesia". Penelitian tersebut memiliki dua kekurangan atau kelemahan atau *research gap,* yaitu (1) pada eksperimen *direct machine translation* (DMT) tidak dapat menerjemahkan kata-kata berimbuhan atau afiksasi dalam bahasa Lampung sehingga akurasi nilai *bilingual evaluation understudy* (BLEU) hasil terjemahan Lampung - Indonesia di bawah 15 %, (2) pada eksperimen *data driven machine translation* yaitu minimnya ketersediaan data digital berupa korpus paralel Lampung – Indonesia.

Tahap awal penelitian dilakukan di tahun 2017. Ketiadaan data korpus digital sangat menyulitkan penelitian NLP pada bahasa Lampung. Untuk mengatasi kebutuhan data korpus digital dilakukan pengetikan secara manual sebanyak 3000

kalimat dialek *Api*, dialek *Nyo* berserta terjemahannya dalam bahasa Indonesia bersumber dari buku pelajaran bahasa Lampung pada tingkat SD dan tambahan sedikit dari buku pelajaran tingkat SMP. Ketersediaan data yang hanya 3000 kalimat ini menjadi penyebab utama rendahnya nilai akurasi *bilingual evaluation understudy* (BLEU) di bawah 78 % pada eksperimen *statistical machine translation* (SMT) sedangkan pada eksperimen *neural machine translation* (NMT) diperoleh nilai akurasi BLEU di bawah 52 %. Pada tahun 2018 - 2022, dilakukan penambahan korpus paralel Lampung – Indonesia khusus pada dialek *Nyo* secara manual bersumber data percakapan sehari-hari penutur asli dialek *Nyo* sehingga saat ini tersedia masing-masing 3000 kalimat dialek *Api* dan 9076 kalimat dialek *Nyo*.

Masalah penerjemahan kata berimbuhan atau afiksasi bahasa Lampung diselesaikan melalui metode *word stemming* (WS), sedangkan masalah *unbalanced data* pada korpus dialek *Api* dan dialek *Nyo* diselesaikan melalui metode *text data augmentation* (TDA) permutasi. WS adalah proses pemetaan bentuk kata yang bervariasi dipetakan ke bentuk dasarnya (Jabbar et el., 2020; Singh & Gupta, 2016, 2017), sedangkan metode TDA bertujuan membuat data sintesis berdasarkan data teks yang ada untuk tujuan memperbanyak korpus digital secara otomatis serta berpotensi digunakan untuk meningkatkan hasil *machine translation*, hasil klasifikasi teks atau hasil *language identification* (Haralabopoulos, et al., 2021). Teknologi *stemming* termasuk dalam bagian *text preprocessing* dasar yang digunakan dalam *language model*, berbagai *task* pada NLP, dan aplikasi *information retrieval* (Jabbar et el., 2020; Singh & Gupta, 2016, 2017). Penelitian *stemming* dan TDA terus berlanjut dalam berbagai arah penerapan (Jabbar et el., 2020; Singh & Gupta, 2016, 2017; Haralabopoulos, et al., 2021).

Perkembangan penelitian WS pada kata berimbuhan pada bahasa Indonesia merujuk pada dua metode utama yaitu metode Nazief-Adriani (Asian et al., 2005) dan metode *Confix-Stripping* (Adriani et al., 2007). Perkembangannya meliputi: (1) pengembangan metode *Confix-Stripping* menjadi metode *Enhanced Confix-Stripping* (Arifin et al., 2009), (2) modifikasi *Confix-Stripping* dilakukan dengan

cara klasifikasi imbuhan pada bahasa Indonesia secara fleksibel dan penanganan reduplikasi (Setiawan et al., 2016), (3) aplikasi *stemming* kata bahasa Indonesia pada berbagai *task*, di antaranya yaitu klusterisasi teks dokumen dengan memanfaatkan *Enhanced Confix-Stripping* (Winarti et al., 2017), (4) komparasi delapan metode *stemmer* kata bahasa Indonesia (Rizki et al., 2019), (5) metode *stemmer* UG18 adalah metode hasil modifikasi pengelompokan, urutan, dan penghapusan afiks berbasis morfofonemik pada bahasa Indonesia (Rizki et al., 2019).

Oleh karena itu, keberhasilan studi WS bahasa Indonesia di Indonesia telah mengilhami penelitian WS pada bahasa daerah (Maesya et al., 2022; Pramana et al., 2022; Paskahningrum, Utami, & Yaqin, 2023). Meskipun bahasa daerah diklasifikasikan sebagai bahasa dengan sumber daya rendah, beberapa bahasa daerah dianggap kurang terwakili meskipun bahasa-bahasa tersebut memiliki potensi untuk penelitian NLP (Jabbar et al., 2023; Maesya et al., 2022; Pramana et al., 2022; Paskahningrum, Utami, & Yaqin, 2023). Penelitian NLP seperti *machine translation*, pencarian informasi, dan analisis sentimen berpotensi untuk dilakukan pada bahasa-bahasa daerah juga.

Penelitian *stemming* pada bahasa daerah telah berlangsung selama satu dekade terakhir. Metode *stemming* yang digunakan pada beberapa bahasa daerah adalah *Ruled-based*, *Brute Force* atau *Tabel look up*, Nazief-Adriani, *Confix-Stripping*, *Enhaced Confix-Stripping*, *N-Gram Stemming*, *Syllable pattern*, dan *Corpus-based*. Bahasa daerah yang digunakan dalam penelitian *stemming* dimulai dari Batak Angkola (Hrp et al., 2023; Muchtar, 2019), Tetun (Guterres et al., 2019), Minangkabau (Sovia et al., 2022; Sovia et al., 2023), Rejang (Wibowo & Wibowo, 2019; Wibowo, et al., 2022), Jawa (Wibawa, et al., 2020), Madura (Lindrawati et al., 2023). Masing-masing bahasa daerah tersebut menggunakan metode berbasis aturan atau morfologi, sedangkan penelitian WS pada bahasa Lampung yang pernah dilakukan yaitu menggunakan metode *brute force* (Abidin et al., 2021).

Penelitian WS bahasa Lampung dialek *Nyo* khususnya pada dialek Tulang Bawang belum pernah dilakukan dan akan dilakukan dengan diawali pada kajian

morfologi. Kajian morfologi bahasa Lampung merujuk pada buku yang dibuat oleh ahli bahasa Lampung dialek Tulang Bawang. Metode pemecahan masalah di atas dilakukan secara komputasi dengan membangun metode model WS berdasarkan Langkah-Langkah: (a) mengkaji dan mencatat semua secara detail pola morfologi kata bahasa Lampung dialek Tulang Bawang, (b) mencari kata-kata atau kalimat-kalimat bahasa Lampung dialek Tulang Bawang yang mengandung imbuhan, (c) membuat model WS secara komputasi, (d) melakukan uji coba metode model WS yang akan dibuat dalam bentuk program dengan *python programming*, (e) menerapkan metode model WS terbaik pada aplikasi DMT Tulang Bawang – Indonesia.

NLP telah mencapai kemajuan besar dalam dekade terakhir melalui penggunaan model *neural* dan *data set* berlabel besar (Haralabopoulos et al., 2021). Ketergantungan pada data linguistik yang melimpah menghambat penerapan model NLP pada lingkungan *low-resource language* atau tugas-tugas baru yang memerlukan waktu, biaya, atau keahlian yang signifikan untuk melabeli sejumlah besar data teks (Haralabopoulos et al., 2021). Belakangan ini, metode augmentasi data telah dieksplorasi sebagai cara untuk meningkatkan efisiensi data dalam NLP (Haralabopoulos et al., 2021). Metode *text data augmentation* (TDA) – sekelompok teknik yang dirancang untuk menghasilkan data pelatihan sintetis – telah menunjukkan hasil yang luar biasa dalam berbagai tugas *Deep Learning* dan *Machine Learning* (Pellicer et al., 2023). Meskipun telah diterapkan secara luas dan sukses dalam komunitas *computer vision*, teknik TDA yang dirancang untuk tugas NLP menunjukkan kemajuan yang jauh lebih lambat dan keberhasilan yang terbatas dalam meningkatkan kinerja (Pellicer et al., 2023).

Penelitian TDA pada bahasa Indonesia yang telah dilakukan dan beberapa penerapannya pada: (1) klasifikasi teks bahasa Indonesia digunakan TDA metode sinonim (Abdurrahman & Purwarianti, 2019), (2) pada *spontaneous Indonesian automatic speech recognition* digunakan TDA metode *statistical machine translation* (SMT) (Vista et al., 2019; Hadiwinoto & Lestari, 2019), (3) *Automatic Essay Scoring* pada bahasa Indonesia digunakan TDA metode *Easy Data Augmentation* (EDA) (Fadilah & Priyatna, 2022), (4) deteksi berita *hoax* digunakan

TDA metode sinonim (Noor et al., 2023), (5) peningkatan model analisis sentimen melalui augmentasi data multiteknik berbasis IndoBERT (Aini et al., 2023), (6) TDA bahasa Indonesia berbasis IndoBERT untuk klasifikasi teks berbahasa Indonesia (Muftie & Haris, 2023), (7) modifikasi EDA dan augmentasi *backtranslation* dalam model *deep learning* untuk analisis sentimen berbasis aspek pada bahasa Indonesia (Natasya & Girsang, 2023), (8) identifikasi parafrasa bahasa Indonesia menggunakan model *Fine-Tuned* IndoBERT dan modifikasi EDA (Kartika et al., 2023),(9) TDA melalui parafrasa untuk mengatasi ketidakseimbangan data pada klasifikasi teks bahasa Indonesia (Sari & Suadaa, 2025).

Penelitian TDA pada bahasa daerah belum banyak ditemukan kecuali temuan hasil terbaru tahun 2025 ini, yaitu TDA dengan metode *Back Translation* khususnya *Neural Machine Translation* pada bahasa Sunda – Inggris (Nicolas et al., 2025) dan Inggris – Indonesia – Madura (Maulana et al., 2025). Adapun penelitian TDA bahasa Lampung difokuskan pada dialek *Api*. Informasi di atas menunjukkan minimnya penelitian TDA pada bahasa daerah dan memperkuat urgensi penelitian TDA pada bahasa Lampung guna mengatasi keterbatasan data korpus paralel dan menunjang kajian NLP bahasa Lampung di masa yang akan datang.

Keberhasilan TDA pada bahasa Lampung berpotensi untuk digunakan untuk melatih klasifikasi otomatis untuk berbagai tujuan, seperti klasifikasi, pemodelan topik, dan analisis sentimen. Masalah minimnya ketersediaan data digital berupa korpus paralel dialek *Api* akan diselesaikan melalui metode TDA permutasi berdasarkan kondisi *data set* yang sudah terkondisi pada dua kelas yaitu dialek *Api* dan *Nyo*. Langkah selanjutnya setelah TDA bahasa Lampung berhasil dilakukan adalah melakukan eksperimen *Lampung Language Dialect Identification* (LLDI) pada dialek *Api* dan dialek *Nyo* sebagai Langkah awal membuat aplikasi *Machine Translation* Lampung – Indonesia yang *smart*.

1.2 Rumusan Masalah

Rumusan masalah penelitian *word stemming* (WS) pada dialek Tulang Bawang (TB) dan *text data augmentation* (TDA) pada kalimat dialek *Api* sebagai berikut:

1. Membuat lima metode model WS pada dialek Tulang Bawang yaitu modifikasi Nazief-Adriani (MNA), modifikasi *Confix-Stripping* (MCS), modifikasi MCS disertai *N-Gram Stemming* sebagai metode *hybrid*, *Morphological-based* serta *N-Gram Stemming* berbasis statistik.
2. Mencari nilai akurasi WS lima metode tersebut berdasarkan nilai *Gold Standar Asessment* (GSA) pada data uji dan data uji independen kata berimbuhan dialek Tulang Bawang.
3. Menerapkan metode WS terbaik dialek Tulang Bawang pada *Direct Machine Translation* (DMT) Tulang Bawang – Indonesia dan mendapatkan nilai skor *Bilingual Evaluation Understudy* (BLEU) dari DMT Tulang Bawang – Indonesia.
4. Membangun model TDA permutasi kalimat dialek *Api* untuk membangkitkan *data set* berupa 6076 kalimat dialek *Api* secara otomatis.
5. Membangun model *Lampung Language Dialect Identification* (LLDI) pada dialek *Api* dan dialek *Nyo* dengan metode *Naive Bayes, Logistic Regression, Support Vector Machine* dan *Random Forest* dengan kondisi *unbalanced data set* yaitu 3000 kalimat dialek *Api* dan 9076 kalimat dialek *Nyo* dengan *5-fold cross validation*.
6. Membangun model *Lampung Language Dialect Identification* (LLDI) pada dialek *Api* dan dialek *Nyo* dengan metode *Naive Bayes, Logistic Regression, Support Vector Machine* dan *Random Forest* dengan kondisi *balanced data set* yaitu 9076 kalimat dialek *Api* dan 9076 kalimat dialek *Nyo* dengan *5-fold cross validation*.

1.3 Tujuan Penelitian

Tujuan dari penelitian *stemming* dan TDA sebagai berikut:

1. Mendapatkan lima metode model WS pada dialek Tulang Bawang yaitu modifikasi MNA, MCS, MCS disertai *N-Gram Stemming*, *Morphological-based* serta *N-Gram Stemming* berbasis statistik.
2. Mendapatkan nilai akurasi WS lima metode tersebut dan model metode WS terbaik berdasarkan nilai *Gold Standar Asessment* (GSA) pada data uji dan data uji independen kata berimbuhan dialek Tulang Bawang.

3. Menyematkan metode model WS terbaik pada bagian *text preprocessing* aplikasi *Direct Machine Translation* (DMT) Tulang Bawang – Indonesia dengan tujuan meningkatkan nilai skor *Bilingual Evaluation Understudy* (BLEU).
4. Mendapatkan model TDA permutasi untuk kalimat dialek *Api* sehingga menghasilkan data sintesis 6076 kalimat dialek *Api*.
5. Melakukan eksperimen *Lampung Language Dialect Identification* (LLDI) pada dialek *Api* dan *Nyo* dengan dua kondisi, yaitu *unbalanced* dan *balanced class*.
6. Mendapatkan model terbaik LLDI pada dialek *Api* dan *Nyo* dengan dua kondisi, yaitu *unbalanced* dan *balanced class*.

1.4 Manfaat Penelitian

Hasil dari penelitian ini memberikan kontribusi nyata pada penelitian NLP bahasa Lampung terutama pada aspek sebagai berikut:

1. Pada aspek *task preprocessing* yaitu penetapan model metode WS terbaik berdasarkan hasil eksperimen untuk kata dialek Tulang Bawang.
2. Aspek *machine translation* yaitu mendapatkan aplikasi DMT Tulang Bawang - Indonesia yang tertanam *stemming* pada *task preprocessing*.
3. Aspek *machine translation* yaitu peningkatan nilai skor BLEU pada aplikasi DMT Tulang Bawang - Indonesia yang tertanam *stemming* pada *text preprocessing*.
4. Aspek peningkatan jumlah korpus berupa data teks kalimat dialek *Api* yang digunakan pada pembuatan model LLDI.
5. Aspek model LLDI atau model identifikasi dialek pada bahasa Lampung.

1.5 Batasan Penelitian

Batasan penelitian yang digunakan adalah sebagai berikut:

1. Teks *roman latin* yang digunakan pada penelitian WS berupa teks *roman latin* bahasa Lampung dialek Tulang Bawang.
2. *Domain* penelitian diambil dari teks buku-buku pelajaran bahasa Lampung SD dan SMP.

3. Teks *roman latin* yang digunakan pada penelitian TDA pada bahasa Lampung dialek *Api* dan sumber kalimat diperoleh dari buku pelajaran bahasa Lampung dialek *Api* dan *Nyo* baik tingkat SD dan SMP.
4. Pada *stemming* kata bahasa Lampung dialek Tulang Bawang tidak menyertakan unsur semantik melainkan hanya aspek morfologi kata untuk mendapatkan kata dasar sesuai kamus.
5. Tidak memproses aksara Lampung dalam bentuk gambar sebagai *input*.
6. Kata dasar diambil dari Kamus Bahasa Lampung edisi kedua yang diterbitkan oleh Kantor Bahasa Provinsi Lampung.
7. TDA berbentuk kalimat pernyataan.

1.6 Kebaruan Penelitian (*Novelty*)

Penelitian ini menghasilkan kebaruan model WS dialek Tulang Bawang terbaik, aplikasi model DMT Tulang Bawang – Indonesia terbaik, model TDA permutasi pada kalimat bahasa Lampung dialek *Api* dan model LLDI untuk dialek *Api* dan *Nyo*. Adapun penjelasan detailnya sebagai berikut:

1. Model modifikasi *Confix-Stripping* disertai *N-Gram Stemming*.
   Model ini berhasil mendapatkan nilai *Gold Standar Asessment* (GSA) terbaik, pada data uji dan data uji independen kata berimbuhan dialek TB, dibandingkan pada model *stemming* modifikasi MNA, MCS, *Morphological-based* dan *N-Gram stemming* berbasis statistik untuk kata bahasa Lampung dialek Tulang Bawang dengan nilai *threshold* 0,5 dan 0,55.
2. Model DMT Tulang Bawang – Indonesia terbaik.
   Aplikasi model DMT Tulang Bawang – Indonesia yang tertanam model modifikasi *Confix-Stripping* disertai *N-Gram Stemming* pada bagian *text preprocessing*.
3. Model TDA permutasi kalimat bahasa Lampung dialek *Api*.
   Penelitian ini menghasilkan model TDA permutasi untuk kalimat dialek *Api*.
4. Model *Lampung Language Dialect Identification* (LLDI) pada dialek *Api* dan *Nyo*.

Penelitian ini menghasilkan model terbaik *Lampung Language Dialect Identification* (LLDI) pada dialek *Api* dan *Nyo* dengan dua kondisi yaitu *unbalanced* dan *balanced class*.

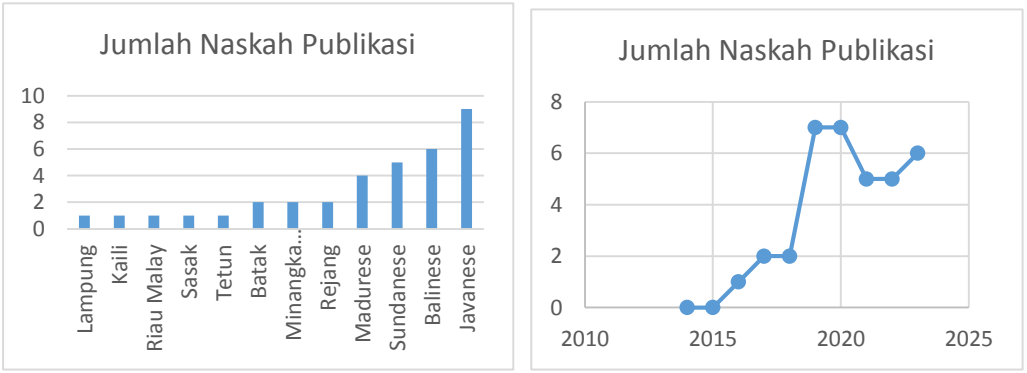
II. TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka (*State of the Art*)

Hasil pencarian topik *stemming* dan *lemmatization* pada bahasa daerah di Indonesia membuahkan hasil berupa fakta terdapat 35 publikasi yang relevan (Abidin et al., 2024). Gambar 2.1 menunjukkan jumlah publikasi topik *stemming* dan *lemmatization* berdasarkan nama bahasa daerah dan tahun penerbitan, sedangkan Tabel 2.1 menunjukkan sebaran 35 publikasi berdasarkan jenis publikasi.

Tabel 2.1 Distribusi 35 publikasi *word stemming* bahasa daerah di Indonesia 2014 – 2023 (Abidin et al., 2024)

Jenis Publikasi	Jumlah Naskah Terpublikasi
Jurnal	27
Prosiding	7
Tesis	1



Gambar 2.1 Visualisasi 35 publikasi *word stemming* bahasa daerah di Indonesia (Abidin et al., 2024)

Penggalian informasi publikasi terkait *stemming* dan *lemmatization* pada bahasa daerah di Indonesia disajikan dalam bentuk naskah publikasi *Systematic Literature Review* (SLR) terhadap 35 artikel ilmiah terkait (Abidin et al., 2024). Perkembangan penelitian topik *stemming* dan *lemmatization* pada bahasa daerah di

Indonesia dari tahun 2014 hingga tahun 2023 dirangkum Tabel 2.2. Metode yang dominan digunakan dalam kajian topik *stemming* dan *lemmatization* pada bahasa daerah di Indonesia adalah dengan memanfaatkan teknik penghilangan imbuhan dan dipadukan dengan kamus digital.

Tabel 2.2 Metode yang digunakan pada 35 naskah publikasi *word stemming* atau *lemmatization* bahasa daerah di Indonesia (Abidin et al., 2024)

Penulis	Bahasa Daerah	Metode
(Melia, et al., 2023)	Jawa Ngoko	Modifikasi ECS
(Wibawa, et al., 2020)	Jawa	Modifikasi Nazief-Adriani (NA)
(Wibawa & Hakim, 2021)	Jawa	*Damerau Levenshtein Distance*
(Amin & Razaq, 2018)	Jawa	Aturan berdasarkan Morfologi
(Hidayatullah, Wibawa, & Rosyid, 2019)	Jawa	ECS untuk Modifikasi NA
(Amin, et al., 2017)	Jawa	Aturan berdasarkan Morfologi dan *String Matching*
(Cahyani, Utami, & Setiadi, 2019)	Jawa Krama Alus	Modifikasi Nazief-Adriani (NA)
(Nq, Manik, & Widiyatmoko, 2020)	Jawa	Modifikasi Nazief-Adriani (NA)
(Wijono, et al., 2021)	Jawa	*Transformer*
(Wardani & Nugraha, 2020)	Bali	Aturan berdasarkan Morfologi
(Nata & Yudiastra, 2017)	Bali	Modifikasi Metode *Porter Stemmer*
(Arimbawa & ER, 2020)	Bali	*Levenshtein Distance*
(Wirayasa, Wirawan, & Pradnyana, 2019)	Bali	Modifikasi Nazief-Adriani (NA)
(Subali & Fatichah, 2019)	Bali	Aturan berdasarkan Morfologi dan *N-Gram*
(Wardani & Nugraha, 2020)	Bali	ECS *Stemmer*
(Putra, et al., 2020)	Sunda	*Not stated*
(Sutedi, Elsen, & Nasrulloh, 2021)	Sunda	*Syllable Pattern* / *Canonical-based*

(Suryani, et al., 2018)	Sunda	Aturan berdasarkan Morfologi
(Sutedi, Nasrulloh, & Elsen, 2022)	Sunda	*Multi Rule-based* dan *Corpus-based*
(Maesya, et al., 2023)	Sunda	*Morphophenemics*
(Maulidi, 2016)	Madura	Modifikasi ECS
(Rachman, et al., 2022)	Madura	Modifikasi ECS dan Aturan berdasarkan Morfologi
(Lindrawati, Utami, & Yaqin, 2023)	Madura	Modifikasi ECS dan NA
(Lindrawati, Utami, & Yaqin, 2023)	Madura	Modifikasi NA
(Wibowo, et al., 2022)	Rejang	Modifikasi ECS dan *New* ECS
(Wibowo & Wibowo, 2019)	Rejang	Modifikasi ECS
(Muchtar, et al., 2019)	Batak Angkola	Aturan berdasarkan Morfologi
(Hrp, Fikry & Yusra, 2023)	Batak Angkola	Aturan berdasarkan Morfologi
(Sovia, Defit, & Yuhandri, 2022)	Minangkabau	Aturan berdasarkan Morfologi
(Sovia, et al., 2023)	Minangkabau	Aturan berdasarkan Morfologi
(Fikry & Yusra, 2021)	Melayu Riau	Aturan berdasarkan Morfologi
(Tamrizal, 2023)	Kaili	Modifikasi NA
(Abidin, Wijaya, and & Pasha, 2021)	Lampung *Api*	*Brute Force*/*Tabel Look Up*
(Guterres, Gunawan, and & Santoso 2019)	Tetun	Aturan berdasarkan Morfologi
(Andriani, Utami, & Suwanto, 2020)	Sasak	Modifikasi Metode *Porter Stemmer*

Hasil pencarian topik *text data augmentation* (TDA) pada bahasa daerah di Indonesia membuahkan 2 hasil yaitu TDA bahasa Sunda – Inggris (Nicolas et al., 2025), Inggris – Indonesia – Madura (Maulana et al., 2025) dan TDA pada bahasa Indonesia diperoleh 11 publikasi yang relevan. Tabel 2.3 menunjukkan sebaran 11 publikasi berdasarkan jenis publikasi dan Tabel 2.4 menunjukkan informasi berdasarkan *task* dan metode yang digunakan.

Tabel 2.3 Distribusi 11 Publikasi TDA bahasa Indonesia 2014 – 2023 (Abidin et al., 2024)

Jenis Publikasi	Jumlah Naskah Terpublikasi
Jurnal	6
Prosiding	5

Tabel 2.4 Distribusi 11 publikasi TDA bahasa Indonesia berdasarkan *task* dan metode

Penulis	*Task*	Metode
(Abdurahman & Purwarianti, 2019)	*Indonesian Text Classification*	*Synonyms Replacement* dan *Language Model*
(Hadiwinoto & Lestari, 2020)	*Indonesian Spontaneus Speech Recognition System*	*Random Augmentation, Augmentation on the Beginning of a Sentence, Augmentation After Connecting Word, Augmentation by Estimate Filled Pause using Hidden Event Language Model*
(Vista, Lestari, & Widyantoro, 2019)	*Spontaneus Indonesian Speech Recognition System*	*Statistical Machine Translation*
(Fadilah & Priyanta, 2022)	*Automatic Essay Scoring*	*EDA (Easy Data Augmentation Techniques)* dan IndoBERT : *Synonym Replacement (SR), Random Insertion (RI), Random Swab (RS), and Random Deletion (RD)*
(Aini, et al., 2023)	*Sentiment Analysis*	*Sequence Generative Adversarial Network (SeqGAN), Easy Data Augmentation (EDA), and An Easier Data Augmentation (AEDA.*
(Noor, Gernowo, & Nurhayati, 2023)	*Hoax Detection in Indonesian News*	*EDA (Easy Data Augmentation Techniques)* : *Synonym Replacement (SR), Random Insertion (RI), Random Swab (RS), and Random Deletion (RD)*
(Kartika, Alfredo, & Kusuma, 2023)	*Indonesia Language Parapharse Identification*	*Modified EDA*
(Indrahimawan, Santosa, & Adji, 2023)	*Classifying Public Complaints*	*Text Augmentation-based oversampling with the Synonym Replacement*
(Muftie & Haris, 2023)	*Indonesian Text Classification*	*IndoBERT Based Data Augmentation*
(Natasya & Girsang, 2023)	*Indonesian Aspect-Based Sentiment Analysis*	*Modified EDA and Backtranslation Augmentation*
(Sujana & Kao, 2023)	*Text Classifcation*	*Language-independent Data Augmentation (LiDA)*

2.2 Konsep *Stemming* dan *Lemmatization*

Konsep *stemming* dan *lemmatization* diambil dari naskah publikasi karya Jasmeet Singh dan Vishal Gupta yang berjudul *A systematic review of text stemming techniques* (Singh & Gupta, 2016) dan *Text Stemming: Approaches, Applications, and Challenges* (Singh & Gupta, 2017). Mengutip pendapat Jasmeet Singh dan Vishal Gupta bahwa pada bidang *Information Retrieval* (IR) dan NLP memerlukan alat *text preprocessing* tertentu untuk analisis tingkat leksikal, morfologi, sintaksis, dan semantik (Singh & Gupta, 2016, 2017). *Stemming* adalah salah satu dari banyak alat *text preprocessing* dan berguna di bidang IR dan NLP seperti klasifikasi teks, pengelompokan, pencarian, peringkasan, *Part of Speech* (POS) *Tag*, dll (Singh & Gupta, 2016, 2017).

Singh & Gupta (2016, 2017) menyatakan bahwa bentuk kata dasar dalam sebagian besar bahasa dimodifikasi untuk membentuk varian bentuk kata sesuai dengan fungsi kata tersebut dalam suatu kalimat. Bentuk-bentuk kata ini terbentuk melalui proses linguistik yang berbeda seperti pemajemukan (kombinasi dua kata atau lebih), afiksasi (penambahan prefiks dan/atau sufiks), konversi (pembentukan kata baru dari kata yang sudah ada), dll. Bentuk-bentuk kata ini sering kali memiliki arti yang sama. *Stemming* adalah prosedur dimana berbagai varian morfologi kata dicocokkan dengan kata dasarnya. Program yang melakukan *stemming* disebut *stemmer*.

Stemming dan *lemmatizing* sering dianggap sebagai proses mirip satu sama lain. Kedua proses ini saling berkaitan dan melakukan fungsi yang sama yaitu mengurangi varian kata dalam teks masukan atau *input*. Perbedaan mendasar antara kedua proses tersebut terletak pada *output* atau luarannya (Singh & Gupta, 2016, 2017). Produk luaran dari *stemming* adalah *"stem"* dan produk keluaran dari *lemmatization* adalah "*lemma*". *Stem* biasanya memiliki arti yang berbeda dan sering kali berorientasi pada *task* tertentu. *Stem* adalah bagian dari kata (dengan atau tanpa makna) yang digunakan untuk membentuk kata-kata baru melalui berbagai metode linguistik seperti penggabungan (misalnya *six-pack*, *day-dream*) atau afiksasi (misalnya *perish-able*, *dur-able*). Kata dasar dapat berupa kata yang valid dan dapat dimengerti sepenuhnya (kata dasar bebas) atau kata yang tidak valid

yang membutuhkan imbuhan untuk membentuk kata dasar (kata dasar terikat). Contohnya, "*perish*" adalah kata dasar bebas dan "*dur*" adalah kata dasar terikat. *Lemma*, di sisi lain, adalah komponen linguistik yang valid dan merupakan bentuk kamus dari *lexeme*. *Lexeme* berhubungan dengan kumpulan semua bentuk varian kata yang memiliki arti yang sama dan *lemma* adalah satu varian tertentu yang digunakan untuk mewakili *lexeme*. Misalnya *run, ran, runs, running* adalah bentuk-bentuk *lexeme* yang berbeda yang diwakili oleh *lemma* yaitu "*run*".

Singh & Gupta (2016, 2017) menjelaskan penekanan antara *stemming* dan *lemmatization*, yaitu *stemming* adalah proses yang lebih sederhana, mudah dan cepat yang menggunakan aturan untuk menentukan *stem* tanpa mempertimbangkan kosakata, konteks kata atau bagian dari kata, sedangkan *lemmatization* adalah prosedur yang relatif rumit yang pertama-tama menentukan bagian dari kata dan konteks kata untuk menghasilkan *lemma*. *Lemmatization* melakukan analisis morfologi lengkap dari kata-kata untuk menentukan *lemma*, sedangkan *stemming* menghilangkan variasi yang mungkin atau mungkin bukan bentuk kata yang benar secara morfologis. Dalam beberapa kasus, *stemmer* dan *lemmatiser* dapat saling menggantikan satu sama lain karena *stemmer* tidak dapat digunakan ketika *output* yang diinginkan adalah kata yang valid dari suatu bahasa. Di sisi lain, *stemmers* dapat dirancang untuk menghapus sufiks derivasional, sedangkan *lemmatizers* hanya menghapus variasi infleksi (Singh & Gupta, 2016, 2017).

Klasifikasi teknik *stemming* terbagi menjadi dua metode utama (Singh & Gupta, 2016, 2017) yaitu pertama *rule-based* dan kedua yaitu *statistical-based*. Pada bagian tinjauan pustaka dan landasan teori ini hanya akan berfokus pada pemaparan tentang metode pertama yaitu *rule-based*. Pada bagian *rule-based* ini terdapat tiga metode yang akan diuraikan sebagai berikut :

a. *Tabel lookup* (*brute force stemming*) : teknik ini menggunakan tabel pencarian yang berisi kata dasar yang sesuai dengan kata berimbuhan atau kata turunan. Untuk menemukan akar kata, tabel tersebut diperiksa. Jika kecocokan ditemukan, akar kata tersebut dikembalikan. Metode ini juga disebut sebagai metode berbasis kamus. Ini adalah teknik yang sederhana

dan mudah digunakan yang dapat menangani kasus-kasus luar biasa juga. Namun, teknik-teknik ini membutuhkan berbagai sumber daya bahasa, dan tidak dapat menangani kata-kata di luar kamus.

b. *Affix stripping algorithms* : Teknik ini menghapus sufiks dan/atau prefiks kata sesuai dengan aturan tertentu atau daftar sufiks. Banyak riset yang telah dilakukan dalam pengupasan sufiks dibandingkan dengan prefiks. Pengembangan aturan untuk *stemmer* membutuhkan keahlian bahasa yang lengkap dan berbagai sumber daya bahasa. Teknik-teknik ini tidak dapat menangani variasi yang disebabkan oleh penggabungan, variasi ejaan dan menghasilkan sejumlah kesalahan karena kata-kata yang dihasilkan setelah pengupasan imbuhan terkadang bukan kata yang sebenarnya.

c. *Morphological stemmers* : Teknik ini mempertimbangkan morfologi bahasa saat melakukan *stemming*. *Stemmer* infleksional mempertimbangkan morfologi infleksional yaitu *stemmer* dapat mendeteksi perubahan kata yang disebabkan oleh sintaksis seperti bentuk kata benda, kata kerja, mengubah bentuk tunggal menjadi jamak tetapi bagian dari ucapan *Part of Speech* (POS) tetap sama. *Derivational stemmer* memperhitungkan morfologi derivasional, yaitu perubahan dalam kategori seperti *nominalization* yaitu kata benda yang dihasilkan dari beberapa kelas lain seperti "*informer*" dari "*inform*", *deadjectival* yaitu kata yang berasal dari kata sifat seperti "*happiness*" dari "*happy*", *deverbal* yaitu kata yang berasal dari kata kerja yang biasanya kata benda atau kata sifat seperti "*readable*" dari ("*read*", "*able*"), dan *denomial* yaitu kata yang berasal dari kata benda seperti "*useful*" dari "*use*". *Stemmer* ini memperhitungkan informasi kamus seperti konteks, arti kata, kosakata, dll. Efisiensi dari *stemmer* ini cukup tinggi karena metode ini mempertimbangkan sintaks dan juga semantik bahasa. Pengembangan metode *Morphological stemmers* membutuhkan pengetahuan yang lengkap tentang bahasa dan morfologinya.

2.3 Evaluasi *Stemming* dan *Lemmatization*

Jabbar et al. (2020) menjelaskan tentang evaluasi hasil dari *stemming*, aplikasi dari metode *stemming*, analisis dari berbagai metode evaluasi *stemming*, tantangan

stemming serta *future research* pada *stemming*. *Text stemming* adalah salah satu Langkah *text preprocessing* dasar untuk NLP yang digunakan untuk mengubah bentuk kata yang berbeda menjadi bentuk dasar yang standar. Jabbar et al. (2020) menyebutkan bahwa evaluasi kinerja adalah metode utama untuk menemukan keefektifan metode dan metode yang dikembangkan untuk memecahkan berbagai masalah ilmiah. Metode evaluasi yang efisien dapat meningkatkan penerapan solusi. Metode evaluasi kinerja menggambarkan dan menentukan sejauh mana solusi dapat mencapai tujuan yang diinginkan.

Jabbar et al. (2020) menyajikan definisi dan deskripsi yang diperlukan untuk memahami evaluasi *stemming* sebagai berikut:

(a) Morfem adalah unit gramatikal terkecil dari suatu bahasa yang tidak dapat dibagi lagi menjadi bagian-bagian yang lebih kecil yang bermakna dan digabungkan bersama untuk membentuk kata-kata yang bermakna.

(b) Afiks adalah morfem yang dapat didefinisikan sebagai kata atau huruf yang dilekatkan pada kata dasar atau kata dasar pada posisi apa pun, misalnya di akhir atau awal atau di kedua sisi kata atau di mana saja di tengah kata. Imbuhan digunakan untuk menghasilkan infleksi dan bentuk turunan dari suatu kata.

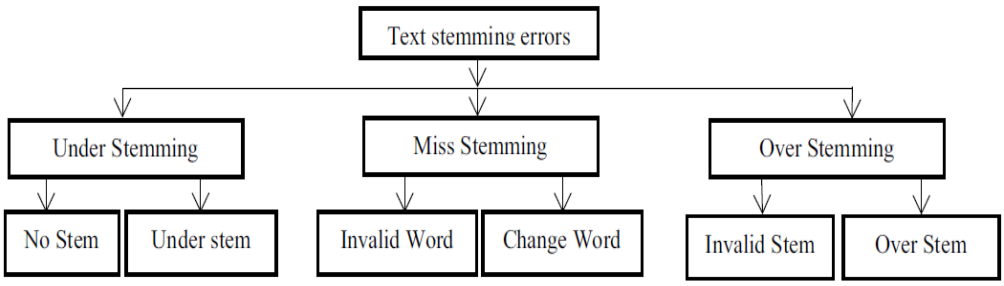
(c) *Inflectional affixes* mengacu pada modifikasi yang mengubah kategori tata bahasa seperti bentuk kata, bentuk tunggal, bentuk jamak, maskulin, feminin, dan netral. *Derivational affixes* adalah kebalikan dari *Inflectional affixes*; *Derivational affixes* membangun kata-kata baru dengan menambahkan imbuhan ke kata dasar.

(d) *Stem* adalah morfem dasar yang dilekatkan dengan morfem lain seperti imbuhan. *Root*, *root* seperti *stem*, tetapi hanya terdiri dari dua unit morfologis yang sederhana. Dengan definisi ini, kita akan menggunakan "*stem*" dan "*root*" secara bergantian. Jika diperlukan, konteks akan menjelaskan arti tertentu.

(e) *Lexeme* menunjukkan morfem yang sama dalam bentuk varian suatu kata. Di sisi lain, *lemma* adalah bentuk pasti yang dipilih dari koleksi *Lexeme* untuk mencirikan *Lexeme* tersebut. *Lemma* adalah bentuk kata dalam kamus yang valid (Singh & Gupta, 2016, 2017). Sebagai contoh kata-kata "*write"*,

"writing", "wrote", "writes" dan *"written"* adalah *lexeme* dan *"write"* adalah *lemma.*

Jabbar et al. (2020) menyajikan informasi terkait *error* pada hasil *stemming* yang diperlukan untuk memahami evaluasi *stemming*. Mengenali jenis-jenis *error* yang dapat dihasilkan oleh suatu *stemmer* adalah Langkah pertama untuk mengukur keefektifan *stemmer* yang diberikan. Jenis-jenis kesalahan ini dapat membantu untuk menemukan jawaban atas pertanyaan seperti kapan dan mengapa kesalahan tersebut terjadi dan apa pengaruhnya terhadap kinerja *stemmer*. Berikut penjelasan berbagai *error* dari hasil *stemming* disajikan pada Gambar 2.2 dan penjelasannya sebagai berikut:



Gambar 2.2 Rangkuman *Error*s pada *Stemming* (Jabbar et al., 2020).

Under stemming errors (USE) yaitu *error* yang mengacu pada fakta ketika *stemmer* memotong huruf-huruf di bawah tingkat yang dapat diterima. Dalam jenis kesalahan ini, *stemmer* menghasilkan kata sebagai kata itu sendiri (tanpa *stem*) atau proses penghilangan imbuhan menghasilkan kata dengan makna yang berubah seperti yang ditunjukkan pada contoh Tabel 2.5.

Tabel 2.5 Contoh *Under Stemming Errors* (Jabbar et al., 2020).

Input Word	***Actual Stem***	***Produced Stem***	***Types of Error***
Acceptance	*Accept*	*Acceptance*	*No Stem*
Acceptances	*Accept*	*Acceptance*	*Under Stem*

Over stemming errors (OSE) adalah *error* di mana *stemmer* memotong lebih banyak karakter dari yang dibutuhkan. Kesalahan OSE mengarah ke *stem* yang tidak valid atau kata di luar kosakata atau *out of verb* (OOV) seperti yang ditunjukkan pada contoh Tabel 2.6.

Tabel 2.6 Contoh *Over Stemming Errors* (Jabbar et al., 2020).

Input Word	***Actual Stem***	***Produced Stem***	***Types of Error***
receiving	*receive*	*receiv*	*Invalid Stem*
consistently	*consist*	*consistent*	*Over Stem*

Mis-stemming errors (MSE) Istilah "*Mis-stemming errors*" mengacu pada kesalahan-kesalahan di mana karakter yang dilucuti tidak membentuk imbuhan yang tepat seperti yang ditunjukkan pada contoh Tabel 2.7.

Tabel 2.7 Contoh *Miss-Stemming Errors* (Jabbar et al., 2020).

Input Word	***Actual Stem***	***Produced Stem***	***Types of Error***
Red	*Red*	*r*	*Invalid Word*
kneel	*kneel*	*knee*	*Change Word*

Jabbar et al. (2020) menyatakan bahwa *stemming* sebagai salah satu tahapan pada *text preprocessing* bermanfaat pada berbagai aplikasi seperti *Information Extraction* (IE), *Information Reterival* (IR), *Text Classification* (TC), *Text Clustering* (TClu), *Question Answering* (QA), *Text Summarizations* (TS), *Machine Translation* (MT), *Text Segmentation* (TS), *Indexing* (Ind), *Automatic Speech Recognition* (ASR), dan *Language Generation*. Dengan kata lain, *stemming* meningkatkan performa dengan mereduksi waktu dan kompleksitas ruang untuk beberapa aplikasi NLP.

Pada penelitian *word stemming* bahasa Lampung berfokus pada penggunaan *Gold Standar Assessments* (GSA). GSA yaitu hasil *stemming* yang diperiksa secara manual untuk mengidentifikasi kata dasar yang benar atau yang mengalami yang mengalami kesalahan (Jabbar et al., 2020). Hasilnya dievaluasi dengan menggunakan kamus Bahasa Lampung lalu dilakukan evaluasi dengan menghitung nilai akurasi. Perhitungan nilai akurasi ditunjukkan pada rumus berikut dengan menggunakan rumus *Gold Standard Assessment* sebagai berikut:

$$\text{Akurasi} = \frac{\text{Jumlah kata Stemming yang benar}}{\text{Jumlah kata Stemming yang diuji}} \text{ x } 100\%$$

Perhitungan akurasi ini biasanya digunakan untuk pengembangan dan evaluasi metode NLP untuk membandingkan hasil keluaran metode dengan hasil yang dianggap benar. Dalam konteks *stemming*, ini dapat merujuk pada satu *set* dokumen

atau kata-kata yang dianggap sebagai standar untuk mengevaluasi kinerja suatu metode *stemming*. Metode GSA bagus untuk *data set* berukuran kecil, tetapi tidak cocok untuk evaluasi skala besar (Jabbar et al., 2020). Metode ini mencerminkan rasio *stem* yang benar yang dihasilkan oleh *stemmer*, tetapi tidak menyebutkan kata-kata yang sudah menjadi *stem* yang diberikan kepada *stemmer*.

2.4 Metode Nazief Adriani, *Confix-Stripping* dan *Enhanced Confix Stripping*

Arifin et al. (2009) berhasil melakukan modifikasi dari metode Nazief-Adriani dan metode *confix-stripping* (CS). Modifikasi dilakukan untuk mengatasi kekurangan dari metode *confix stripping* yang sudah ada dengan mengusulkan versi terbaru yang disebut *Enhanced confix stripping stemmer* (Arifin et al., 2009). Metode CS adalah teknik *stemming* bahasa Indonesia yang pertama kali diperkenalkan oleh Nazief dan Adriani pada tahun 1996 lalu kemudian dikembangkan oleh Asian et al. (2005). Metode CS mengelompokkan imbuhan ke dalam beberapa kategori:

1. *Inflection Suffixes*, kumpulan sufiks yang tidak mengubah kata dasarnya. *Inflection Suffixes* dibagi menjadi:
   • Partikel (P), termasuk partikel "-lah", "-kah", "-tah", dan "-pun".
   • *Possessive Pronoun* (PP), termasuk "-ku", "-mu", dan "-nya".
2. *Derivation Suffixes* (DS), yaitu rangkaian sufiks yang langsung diterapkan pada kata dasar, yang termasuk dalam jenis sufiks ini adalah "-i", "-kan", dan "-an".
3. *Derivation Prefixes* (DP), yaitu kumpulan prefiks yang diterapkan secara langsung pada kata dasar, atau pada kata yang memiliki hingga dua prefiks turunan lainnya. Jenis prefiks ini dibagi menjadi prefiks kompleks ("me-", "be-", "pe-", dan "te-") dan prefiks biasa ("di-", "ke-", dan "se-").

Klasifikasi imbuhan ini menghasilkan urutan model kata bahasa Indonesia sebagai berikut:

[DP + [DP + [DP+]]] kata dasar [[+DS] [+PP] [+P]]

Namun, ada pengecualian dan batasan yang dimasukkan ke dalam aturan tersebut:

- Tidak semua kombinasi imbuhan dapat digunakan. Kombinasi prefiks dan sufiks yang tidak diperbolehkan ditunjukkan pada Tabel 2.8.

- Imbuhan yang sama tidak dapat diterapkan berulang kali.

- Jika suatu kata hanya memiliki satu atau dua karakter, *stemming* tidak akan dilakukan.

- Penambahan prefiks dapat mengubah kata dasar atau prefiks yang telah diterapkan sebelumnya. Sebagai contoh, prefiks "me-" yang diterapkan pada kata "tambah", akan menghasilkan kata "menambah" dimana "t" diubah menjadi "n".

Tabel 2.8 Kombinasi Prefiks - Sufiks yang dilarang (Arifin et al., 2009)

Prefiks	Sufiks yang dilarang
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

Langkah-Langkah *Confix stripping stemmer* adalah sebagai berikut:

Langkah 1.

Pada awal pemrosesan dan setiap Langkah, periksa kata saat ini terhadap kamus kata dasar. Jika pencarian berhasil, kata tersebut dianggap sebagai kata dasar, dan pemrosesan berhenti.

Langkah 2.

Periksa prioritas aturan. Jika kata tersebut memiliki kombinasi prefiks-sufiks "be-lah", "be-an", "me-i", "di-i", "pe-i", atau "te-i", Langkah selanjutnya yang akan dijalankan adalah Langkah (5, 6, 3, 4, 7). Jika tidak, langkah-langkah *stemming* dijalankan secara normal adalah Langkah (3, 4, 5, 6, 7).

Langkah 3.

Menghilangkan partikel infleksi P ("-lah", "-kah", "-tah", "-pun") jika ada di dalam kata, dan dilanjutkan dengan menghilangkan kata ganti kepunyaan PP ("-ku", "-mu", "-nya").

Langkah 4.

Menghilangkan sufiks derivasional DS ("-i", "-kan", atau "-an").

Langkah 5.

Menghapus prefiks derivasional DP ("di-", "ke-", "se-", "me-", "be-", "pe", "te-") dengan maksimal tiga kali iterasi:

a) Hentikan pemrosesan jika:

- Prefiks yang teridentifikasi membentuk kombinasi imbuhan yang tidak diperbolehkan dengan sufiks yang telah dihapus pada Langkah sebelumnya;

- Prefiks yang teridentifikasi identik dengan prefiks yang telah dihapus sebelumnya; atau

- Tiga prefiks telah dilakukan atau dihapus.

b) Mengidentifikasi jenis prefiks dan menghapusnya. Prefiks dapat terdiri dari dua jenis:

- Prefiks biasa: prefiks "di-", "ke-", dan "se-" yang dapat dihapus secara langsung.

- Prefiks Kompleks: prefiks "me-", "be-", "pe", dan "te-" yang dapat mengubah kata dasar. Gunakan aturan yang dijelaskan pada Tabel 2.9, 2.10, 2.11, 2.12 untuk mendapatkan penghilangan prefiks yang benar.

c) Langkah ini (Langkah 5) dicoba kembali ketika pencarian kamus gagal pada kata yang sedang dicari.

Langkah 6.

Jika, setelah penghapusan prefiks pada Langkah 5, kata dasar masih belum ditemukan, periksa apakah pengodean ulang dapat dilakukan dengan memeriksa kolom terakhir Tabel 2.9, 2.10, 2.11, atau 2.12. Pengodean ulang dilakukan dengan mengganti dan menambahkan karakter *recording* atau pengodean ulang pada huruf pertama dari kata tersebut. Karakter *recording* atau pengodean ulang pada Tabel 2.9, 2.10, 2.11, dan 2.12 adalah karakter huruf kecil setelah tanda hubung (“-”) dan terkadang di luar tanda kurung. Sebagai contoh, ketika menghilangkan prefiks "me-" pada kata "menangkap", dengan melihat aturan 6 pada Tabel 2.9, ada dua kemungkinan karakter yang dapat digunakan, yaitu "n" dan "t". Menambahkan "n" akan menghasilkan kata "nangkap", dan sayangnya ini bukan kata yang valid.

Menambahkan "t" akan menghasilkan kata "tangkap" yang merupakan kata dasar yang valid.

Langkah 7.

Jika semua langkah gagal, metode akan mengembalikan kata asli yang belum di-*stemming*.

Tabel 2.9 Aturan Penghapusan pada Prefiks “me-” (Arifin et al., 2009).

Aturan	Pola Konstruksi Prefiks	Pola Penghapusan Prefiks
1	me{l|r|w|y}V...	me-{l|r|w|y}V...
2	mem{b|f|v}...	mem-{b|f|v}...
3	mempe...	mem-pe...
4	mem{rV|V}...	me-m{rV|V}...|me-p{rV|V}...
5	men{c|d|j|z}...	men-{c|d|j|z}...
6	menV...	me-nV...|me-tV...
7	meng{g|h|q|k}...	meng{g|h|q|k}...
8	mengV...	meng-V...|meng-Kv...
9	menyV...	meny-sV...
10	mempV...	mem-pV... Dimana V!=“e”

Tabel 2.10 Aturan Penghapusan pada Prefiks “pe-” (Arifin et al., 2009).

Aturan	Pola Konstruksi Prefiks	Pola Penghapusan Prefiks
1	pe{w|y}V...	pe-{w|y}V...
2	perV...	per-V...|pe-rV...
3	perCAP	per-CAP... Dimana C!=“r” dan P!=“er”
4	perCAerV...	per-CAerV... Dimana C!=“r”
5	pem{b|f|V}...	pem-{b|f|V}
6	pem{rV|V}...	pe-m{rV|V}...|pe-p{rVV|}
7	pen{c|d|j|z}...	pen-{c|d|j|z}...
8	penV...	pe-nV...|pe-tV...
9	peng{g|h|q}...	peng-{g|h|q}...
10	pengV...	peng-V...|peng-kV...
11	penyV...	peny-sV...
12	pelV...	pe-Lv... Kecuali “pelajar” menjadi “ajar”
13	peCerV...	per-erV... Dimana C!={r|w|y|l|m|n}
14	peCP...	pe-CP... Dimana C!={r|w|y|l|m|n} dan P!=“er”
15	teC$_1$erC$_2$	te-C$_1$erC$_2$ ... Dimana C$_1$!=“r”
16	peC$_1$erC$_2$	pe-C$_1$erC$_2$ ... Dimana C$_1$!= {r|w|y|l|m|n}

Ada kondisi khusus ketika tanda hubung (“-”) ditemukan pada kata yang akan dibendakan. Hal ini mengindikasikan bahwa kata tersebut mungkin merupakan kata jamak atau kata yang diulang. Untuk jenis kata seperti ini, *stemming* dilakukan secara terpisah pada sub-kata yang mendahului dan mengikuti tanda hubung. *Stemming* berhasil jika kedua sub-kata tersebut memiliki kata dasar yang sama. Huruf “C” pada Tabel 2.9, 2.10, 2.11 dan 2.12 menunjukkan konsonan, huruf “V” menunjukkan vokal, huruf “A” menunjukkan huruf apa saja, dan huruf “P” menunjukkan penggalan kata yang pendek, seperti "er".

Tabel 2.11 Aturan Penghapusan pada Prefiks “be-” (Arifin et al., 2009).

Aturan	Pola Konstruksi Prefiks	Pola Penghapusan Prefiks
1	berV...	ber-V...|be-Rv...
2	berCAP...	ber-CAP... dimana C!=“r” dan P!=“er”
3	berCAerV...	ber-CAerV... Dimana C!=“r”
4	belajar	bel-ajar
5	beC_1erC_2	beC_1erC_2 ... Dimana C_1 !={r|l}

Tabel 2.12 Aturan Penghapusan pada Prefiks “te-” (Arifin et al., 2009).

Aturan	Pola Konstruksi Prefiks	Pola Penghapusan Prefiks
1	terV...	ter-V...|te-rV...
2	terCerV...	ter-CerV... dimana C!=“r”
3	terCP...	ter-CP... dimana C!=“r” dan P!=“er”
4	teC_1erC_2...	teC_1erC_2... dimana C_1 != “r”

Arifin et al. (2009) menyatakan bahwa setelah melakukan percobaan terbatas pada metode *confix-stripping* (CS), diperoleh beberapa kegagalan yang dibuat oleh metode CS dan diklasifikasikan sebagai berikut:

1. Tidak ada aturan penghapusan prefiks untuk kata-kata dengan konstruksi "mem+p...", misalnya, "mempromosikan", "memproteksi", dan "memprediksi".

2. Tidak ada aturan penghilangan prefiks untuk kata dengan konstruksi "men+s...", misalnya, "mensyaratkan", dan "mensyukuri".
3. Tidak ada aturan penghilangan prefiks untuk kata dengan konstruksi "menge+...", misalnya, "mengerem".
4. Tidak ada aturan penghilangan prefiks untuk kata dengan konstruksi "penge+...", misalnya, "pengeboman".
5. Tidak ada aturan penghilangan prefiks untuk kata dengan konstruksi "peng+k...", misalnya, "pengkajian".
6. Kegagalan penghapusan sufiks - terkadang penggalan terakhir dari suatu kata menyerupai sufiks tertentu. Sebagai contoh, kata-kata seperti "pelanggan" dan "pelaku" gagal dibendung, karena "-an" dan "-ku" pada bagian akhir kata tidak boleh dihilangkan.

Berdasarkan kegagalan tersebut, Arifin et al. (2009) mencoba memperluas metode CS dan menyajikan metode CS yang telah dimodifikasi yang disebut *enhanced confix stripping* (ECS) *stemmer*. Arifin et al. (2009) melakukan perbaikan sebagai berikut:

1. Memodifikasi beberapa aturan pada Tabel 2.9 dan 2.10, sehingga proses *stemming* pada kata dengan konstruksi "mem+p...", "men+s...", "menge+...", "penge+...", dan "peng+k..." dapat dilakukan. Modifikasi ini tercantum dalam Tabel 2.13.
2. Menambahkan langkah *stemming* tambahan untuk menyelesaikan masalah penghilangan sufiks. Arifin et al. (2009) menyebut langkah tambahan ini sebagai *loopPengembalianSufiks*. Langkah ini dilakukan ketika pengodean ulang yaitu pada pada Langkah 6, di tahapan *CS stemmer* gagal melakukan *stemming*.

Arifin et al. (2009) menyatakan bahwa setiap proses l*oopPengembalianSufiks*, pencarian kamus dilakukan untuk mengecek hasil dari kata yang sedang dicari. Langkah-langkah dalam *loopPengembalianSufiks* didefinisikan sebagai berikut:

Langkah 1.

Kembalikan kata ke bentuk *pre-recording* dan kembalikan semua prefiks yang telah dihapus pada proses terakhir, sehingga akan menghasilkan model kata seperti berikut:

[DP + [DP + [DP]]] + Kata dasar

Selanjutnya, penghapusan prefiks akan dicoba. Jika pencarian kamus berhasil, maka proses akan berhenti. Jika tidak, maka Langkah selanjutnya akan dijalankan.

Langkah 2.

Kembalikan sufiks yang telah dihapus sebelumnya. Artinya, pengembalian dimulai dari DS ("-i", "-kan", "-an") jika ada, kemudian diikuti dengan PP ("-ku", "-mu", "-nya"), dan yang terakhir adalah P ("-lah", "-kah", "-tah", "-pun"). Pada setiap pengembalian, Langkah 3 hingga Langkah 5 di bawah ini dicoba. Kasus khusus untuk DS "-kan", karakter "k" dikembalikan terlebih dahulu dan Langkah 3 hingga Langkah 5 dijalankan. Jika masih gagal, maka "an" dikembalikan.

Langkah 3.

Penghapusan prefiks dilakukan sesuai dengan aturan yang didefinisikan pada Tabel 2.9, 2.10, 2.11 dan 2.12 (dengan modifikasi pada Tabel 2.13 dan 2.14).

Langkah 4.

Lakukan *Recording*. Jika pencarian kamus tidak berhasil, kembalikan kata ke bentuk *pre-recording* dan kembalikan semua prefiks yang telah dihapus. Sufiks berikutnya sesuai urutan pada Langkah 1 dikembalikan dan Langkah 3 hingga Langkah 5 dilakukan terhadap kata saat ini.

Tabel 2.13 Modifikasi Aturan Penghapusan pada Tabel 2.10 (Arifin et al., 2009)

Aturan	Pola Konstruksi Prefiks	Pola Penghapusan Prefiks
5	men{c|d|j|s|z}...	men-{c|d|j|s|z}...
6	mengV...	meng-V...|meng-kV....| (mengV-... Jika V=“e”)
10	mempA...	mem-pA... Dimana A!=“e”

Tabel 2.14 Modifikasi Aturan Penghapusan pada Tabel 2.11 (Arifin et al., 2009)

Aturan	Pola Konstruksi Prefiks	Pola Penghapusan Prefiks
9	pengC...	peng-C...
10	pengV...	peng-V...|peng-kV...| (pengV-... Jika V=“e”)

2.5 Morfologi Dialek Tulang Bawang

Morfologi linguistik mempelajari struktur, bentuk, dan penciptaan kata. Unit bahasa yang paling sederhana, morfem seperti akar kata, prefiks, sufiks, infiks, konfiks dan reduplikasi diperiksa (Hermawan et al., 2001). Studi morfologi juga meneliti bagaimana morfem bergabung untuk menghasilkan kata-kata baru atau mengubah makna. Morfologi memungkinkan kita mempelajari pola produksi kata dalam berbagai bahasa dan bagaimana struktur kata mempengaruhi sintaksis dan semantik. Struktur dan konstruksi kata membedakan morfologi Tulang Bawang (TB) dengan dialek lain. Dialek ini menggunakan prefiks, sufiks, infiks, konfiks, dan reduplikasi untuk membuat kata atau mengubah artinya. Semua penjelasan morfologi TB berasal dari buku "Sistem Morfologi Verba Bahasa Lampung Dialek Tulang Bawang" (Hermawan et al., 2001) dan buku "Afiksasi Verba Bahasa Lampung" karya Prof. Dr. Farida Ariyani, M.Pd. (Ariyani, 2014).

A. Kata Dasar

Kata kerja berasal dari kata dasar tanpa afiksasi, reduplikasi, atau komposisi. Kata kerja dasar bahasa Tulang Bawang menjelaskan tindakan, sikap, dan aktivitas sederhana dengan makna langsung. Tindakan sederhana seperti "mengan" (berarti "makan") dan "cekak" (berarti "naik") adalah contohnya. Makna dan konteks tata bahasa dari kata kerja turunan adalah diubah dengan menambahkan prefiks atau sufiks pada kata kerja dasar ini.

B. Kata Kerja Turunan

Kata kerja turunan dibuat dengan menambahkan bentuk morfologi seperti afiksasi, reduplikasi, atau penggabungan ke kata kerja dasar. Proses ini mengubah arti atau fungsi tata bahasa dari kata kerja dasar, sehingga menghasilkan variasi linguistik yang lebih baik dan ketepatan kontekstual. Afiksasi dalam dialek Tulang Bawang terdiri dari sufiks, prefiks, infiks, dan konfiks. Kata majemuk adalah pengecualian dalam penelitian ini. Prefiks adalah imbuhan yang berada di awal kata. Jenis-jenis prefiks adalah "di-", "be-", "te-", "pe-", "pegh-" sedangkan prefiks nasal N- = {"nge-", "ng-", "ny-", "n-", "m-" }. Tabel 2.15 Aturan yang berisi prefiks dan dampaknya dalam mengubah kata kerja dasar bahasa Tulang Bawang. Tabel 2.16

berisi prefiks dan terjemahannya dalam bahasa Indonesia. Catatan tambahan untuk Tabel 2.15, di mana A adalah huruf sembarang, C adalah huruf konsonan dan V adalah huruf vokal.

Tabel 2.15 Aturan Prefiks (Hermawan et al., 2001)

Aturan	Pola Kontruksi Prefiks	Pola Penghapusan Prefiks
1	nge{b|d|g|h|l|n|r|w|y}AA...	nge- +{b|d|g|h|l|n|r|w|y}AA...
2	ng{a|i|u|e|o}CV...	ng- +{a|i|u|e|o}CV... | ng- + k{a|i|u|e|o}CV...
3	nyVC...	ny- + cVC... | ny- + sVC...
5	nVC...	n- + tVC...
6	mVC...	m- + pVC...
8	beAA...	be- +AA...
9	peCV...	pe- +CV...
11	perCV...	per- +CV...
12	teAA...	te- +AA..
13	diAA...	di- +AA...

Tabel 2.16 Prefiks pada Dialek Tulang Bawang beserta Contoh (Hermawan et al., 2001).

Prefiks	Akar Kata	Contoh pada Dialek Tulang Bawang	Terjemahan dalam bahasa Indonesia
di-	*kayun*	*dikayun*	disuruh
be-	*tanei*	*betanei*	bertani
te-	*alau*	*tealau*	terkejar
pe-	*wawai*	*pewawai*	perindah
per-	*tego*	*pertego*	pertiga
nge-	*nah*	*ngenah*	melihat
ny-	*cobou*	*nyobou*	mencoba
ng-	*kawil*	*ngawil*	mengail
n-	*taban*	*naban*	menggendong
m-	*peppul*	*meppul*	membakar

Kata-kata yang diakhiri dengan infleksi diakhiri dengan sufiks. Sufiks termasuk “-ei”, “-ken”, “-nou”, “-meu”, “-keu”, dan “-lah”. Tabel 2.17 menunjukkan sufiks,

dan hasil modifikasi kata dasar Tulang Bawang yang tidak berubah bentuk. Infiks masuk ke dalam dan berada pada posisi karakter kedua dan ketiga atau posisi karakter ketiga dan keempat dari akar kata. Efek infiks ini mengubah kata. Pada dialek Tulang Bawang terdapat infiks yaitu "-em-". Tabel 2.18 menunjukkan infiks dan modifikasi kata dasar Tulang Bawang merubah bentuk kata.

Tabel 2.17 Sufiks pada Dialek Tulang Bawang beserta Contoh (Hermawan et al., 2001).

Sufiks	**Akar Kata**	**Contoh pada Dialek Tulang Bawang**	**Terjemahan dalam bahasa Indonesia**
-ei	*baccuh*	*baccuhei*	tambahi
-ken	*oloh*	*olohken*	kembalikan
-nou	*nuwou*	*nuwounou*	rumahnya
-meu	*adik*	*adikmeu*	adikmu
-keu	*katan*	*katankeu*	lukaku
-lah	*mejeng*	*mejenglah*	duduklah

Tabel 2.18 Infiks pada Dialek Tulang Bawang beserta Contoh (Hermawan et al., 2001).

Infiks	**Akar Kata**	**Contoh pada Dialek Tulang Bawang**	**Terjemahan dalam bahasa Indonesia**
-em-	*cengguk*	*cemengguk*	menunduk

Tabel 2.19 menunjukkan tiga bentuk reduplikasi dalam bahasa Tulang Bawang, yaitu reduplikasi sempurna, reduplikasi suku kata pertama, dan reduplikasi dengan imbuhan. Reduplikasi suku kata pertama terjadi jika suku kata pertama memiliki pola karakter pertama adalah konsonan dan karakter kedua adalah "a", "u", atau "o", maka suku kata baru dibuat dengan pola di mana karakter pertama adalah konsonan dan karakter kedua adalah "e" pada posisi awal kata. Reduplikasi dengan imbuhan terjadi pada kata pertama atau kedua yang dipisahkan oleh simbol "-". Reduplikasi dalam bahasa Lampung memiliki pola yang unik. Konfiks memiliki dua bagian: satu pada prefiks kata dasar dan satu lagi pada sufiks. Kedua elemen tersebut mengubah kata kerja dasar atau membuat kata kerja baru. Konfiks tersebar luas dalam bahasa Indonesia dan bahasa daerah. Konfiks dalam dialek Tulang

Bawang dapat dilihat pada Tabel 2.20. Kata-kata yang mengandung prefiks pada konfiks akan dibedakan sesuai dengan aturan yang diuraikan dalam Tabel 2.20.

Tabel 2.19 Reduplikasi pada Dialek Tulang Bawang beserta Contoh (Hermawan et al., 2001)

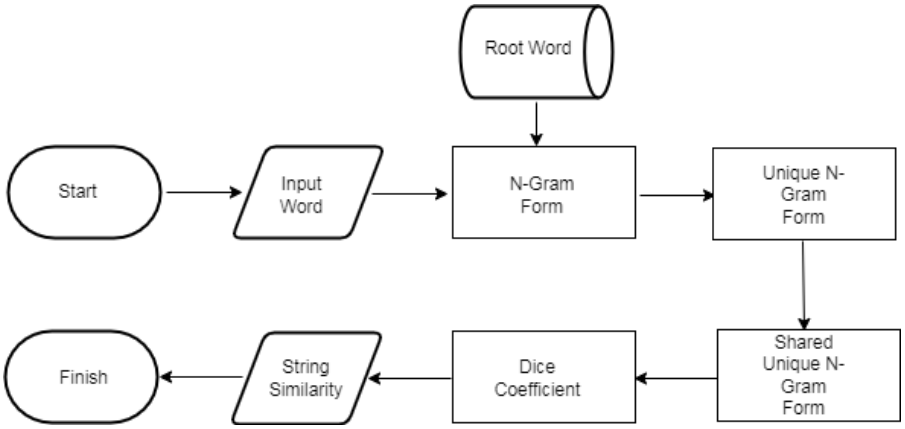
Reduplikasi	**Akar Kata**	**Contoh pada Dialek Tulang Bawang**	**Terjemahan dalam bahasa Indonesia**
Reduplikasi Sempurna	*cobou*	*cobou-cobou*	coba-coba
	nah	*nah-nah*	lihat-lihat
Reduplikasi pada Silabel Awal + Sufiks *-an*	*dakep*	*dedakepan*	saling berpelukan
	lapah	*lelapahan*	berjalan-jalan
Reduplikasi disertai Afiksasi	*balah*	*bebalah-balah*	berkata-kata
	leccak	*beleccak-leccakan*	berlompat-lompatan

Tabel 2.20 Konfiks pada Dialek Tulang Bawang beserta Contoh (Hermawan et al., 2001)

Konfiks	**Akar Kata**	**Contoh pada Dialek Tulang Bawang**	**Terjemahan dalam bahasa Indonesia**
nge-ei	*biyak*	*ngebiyakei*	memberati
ng-ei	*arit*	*ngaritei*	mengariti
ny-ei	*cabut*	*nyabutei*	mencabuti
m-ei	*pacul*	*maculei*	mencangkuli
nge-ken	*golai*	*ngegolaiken*	menggulaikan
ng-ken	*akuk*	*ngakukken*	mengambilkan
ny-ken	*sugeu*	*nyugeuken*	menyuguhkan
n-ken	*tanem*	*nanemken*	menanamkan
m-ken	*pajak*	*majakken*	merebuskan
be-ken	*pakkul*	*bepakkulken*	beratapkan
be-an	*dakep*	*bedakepan*	berpelukan
di-ken	*akuk*	*diakukken*	diambilkan
di-ei	*kan*	*dikanei*	dimakani
per-ken	*nah*	*pernahken*	perlihatkan
ke-an	*kughuk*	*kekughukan*	kemasukan

2.6 *N-Gram Stemming*

Jika aturan yang tersedia pada MNA, MCS, *morphological-based* tidak dapat mengenali kata berimbuhan tersebut, proses kemiripan *string* dapat digunakan metode *n-gram stemming* (Bali et al., 2019), misalkan salah satunya karena kesalahan penulisan kata berimbuhan, dapat diatasi dengan *n-gram stemming* (Bali et al., 2019). Alur proses dari metode *n-gram stemming* yang diusulkan Bali et al. (2019) ditunjukkan pada Gambar 2.3.



Gambar 2.3 Aliran Proses *Stemming* dengan *N-Gram Stemming* (Bali et al., 2019)

Awalnya, kata imbuhan dan kata dasar dalam basis data ditransformasikan ke dalam format *n-gram*, diikuti dengan perbandingan yang melibatkan penghitungan jumlah *n-gram* unik atau jumlah karakter *n-gram* yang dihasilkan dan *n-gram* unik yang sama, atau jumlah karakter *n-gram* yang sama antara kata imbuhan dan kata dasar. Kemiripan antara *n-gram* kata imbuhan dan *n-gram* kata dasar dikuantifikasi dengan menggunakan metode *Dice Coefficient* sesuai dengan persamaan (1) (Bali et al., 2019).

$$dc = (2 . c) / (a + b) \quad (1)$$

Pada persamaan (1), *dc* mewakili *dice coefficient*. *c* mewakili *n-gram* unik bersama antara dua kata, *a* menunjukkan *n-gram* unik dari kata masukan, dan *b* menunjukkan *n-gram* unik dari kata dasar (Bali et al., 2019). Proses menilai kesamaan antara dua kata, *a* dari kata masukan dan *b* dari kata dasar, menggunakan

stemming n-gram, menggunakan jumlah karakter *n*, yaitu dua sebagai *bi-gram* atau tiga sebagai *tri-gram* (Bali et al., 2019).

Menetapkan Nilai Ambang Batas / *Threshold*

Ketika hasil dari komputasi *stemming n-gram* antara kata masukan dan kata dasar memenuhi nilai ambang batas, maka kata dasar akan ditampilkan. Nilai ambang batas yang digunakan adalah antara 0,5 sampai 0,8 dengan kenaikan sebesar 0,05 (Bali et al., 2019). Simulasi dilakukan untuk mendapatkan nilai *threshold* yang optimal.

Menilai Akurasi *Stemming*

Hasil akurasi *stemming* dihitung menggunakan persamaan (2) (Jabbar et al., 2020).

$$\textit{gold standar assessment} = (t / n) \times 100 \qquad (2)$$

Pada persamaan (2), *gold standar assessment* merepresentasikan akurasi *stemming*. *t* merepresentasikan jumlah kata yang telah di-*stemming* secara akurat, dan *n* merepresentasikan jumlah kata yang di-*stemming*.

2.7 *Direct Machine Translation*

DMT dilakukan dengan cara memproses pemetaan satu per satu kata yang terdalam dalam kalimat dari bahasa sumber menuju bahasa tujuan dengan menggunakan bantuan kamus dwi bahasa. Dalam proses penerjemahan secara langsung, *machine translation* tidak mengamati struktur kalimat bahasa sumber melainkan hanya melakukan *text preprocessing* dan analisis morfologi yang dangkal guna menjadikan kalimat tersebut menjadi suatu daftar kata-kata. Daftar kata-kata yang dihasilkan, dari bahasa sumber, akan dilakukan pencocokan satu per satu dengan menggunakan kamus dwi bahasa. Daftar padanan kata-kata, dari bahasa sumber menuju bahasa target, yang menemui kecocokan dengan kamus dwi bahasa akan dikumpulkan kembali guna dilakukan penyusunan ulang sesuai tata susunan bahasa target. Langkah yang terakhir adalah pembangkitan hasil terjemahan secara langsung secara morfologi untuk mendapatkan susunan kalimat yang sesuai dengan bahas tujuan (Jurafsky & Martin, 2026).

Rata-rata geometris diambil dari skor presisi yang dimodifikasi dari korpus uji, lalu dikalikan hasilnya dengan suatu faktor *brevity penalty* eksponensial. Saat ini, perubahan huruf besar menjadi huruf kecil adalah satu-satunya normalisasi teks yang dilakukan sebelum menghitung presisi (Papineni et al,, 2002). Pertama, rata-rata geometris dihitung dari presisi *n-gram* yang dimodifikasi, p_n, menggunakan *n-gram* hingga panjang *N* dimana *N* = 4 dan bobot positif w_n yang berjumlah satu (Papineni et al,, 2002). Diketahui bahwa *c* menyatakan panjang kandidat terjemahan, dan *r* menyatakan panjang efektif korpus referensi (Papineni et al,, 2002). *Brevity penalty* (BP) dihitung sebagai:

$$BP = 1 \; if \; c > r \; or \; BP = \; e^{\left(1-\frac{r}{c}\right)} \; if \; c \leq r \tag{3}$$

$$BLEU = BP \; . exp(\sum_{n=1}^{N} w_n log p_n) \tag{4}$$

Perilaku pemeringkatan lebih mudah diamati dalam domain logaritmik.

$$\log BLEU = min\left(1 - \frac{r}{c}, 0\right) + \; exp(\sum_{n=1}^{N} w_n log p_n) \tag{5}$$

Dalam *baseline*, dipilih *N* = 4 dan bobot seragam $w_n = 1/N$ (Papineni et al,, 2002). *Bilingual Evaluation Understudy* (BLEU) adalah metrik yang dirancang untuk menilai kualitas terjemahan yang dihasilkan oleh mesin dengan membandingkannya dengan teks referensi yang ditulis oleh manusia. BLEU menghitung kemiripan menggunakan presisi *n-gram*, yang memeriksa berapa banyak urutan kata-seperti kata tunggal (*unigram*), pasangan (*bigram*), atau hingga potongan empat kata (*4-gram*)-dalam keluaran mesin yang sesuai dengan referensi. Skor ketepatan ini digabungkan ke dalam rata-rata geometris, dan penalti keringkasan menyesuaikan skor untuk menghukum terjemahan yang terlalu pendek.

Skor BLEU berkisar dari 0 hingga 1, dengan nilai yang lebih tinggi menunjukkan kemiripan yang lebih dekat dengan referensi. Kekuatannya meliputi kesederhanaan dan kecepatan, sehingga banyak digunakan, tetapi memiliki keterbatasan: BLEU melewatkan sinonim atau kata ulang yang memiliki makna yang sama dan memprioritaskan kecocokan yang sama persis daripada keakuratan semantik.

Singkatnya, BLEU menawarkan cara yang praktis namun tidak sempurna untuk mengevaluasi kualitas terjemahan.

2.9 *Text Data Augmentation* dengan Metode Permutasi

Teks secara tradisional telah digunakan untuk melatih pengklasifikasi otomatis untuk berbagai tujuan, seperti: klasifikasi, pemodelan topik, dan analisis sentimen. Pengklasifikasi model *long short term memory* (LSTM) membutuhkan sejumlah *data training* yang besar untuk menghindari bias dan melakukan generalisasi model (Haralabopoulos, et al., 2021). Data berlabel berpotensi meningkatkan hasil klasifikasi, tetapi tidak semua *data set* modern menyertakan sejumlah sampel besar berlabel (Haralabopoulos, et al., 2021). Pelabelan adalah tugas yang kompleks, mahal, memakan waktu, dan berpotensi menimbulkan bias (Haralabopoulos, et al., 2021). Metode augmentasi data menciptakan data sintetis berdasarkan contoh berlabel yang sudah ada, dengan tujuan meningkatkan hasil klasifikasi (Haralabopoulos, et al., 2021).

Haralabopoulos et al. (2021) mengusulkan metode permutasi kalimat untuk memperkaya *data set* awal, dengan tetap mempertahankan properti statistik utama dari *data set* tersebut seperti frekuensi istilah dan distribusi kelas serta meningkatkan hasil klasifikasi. Dalam *supervised learning*, suatu model dilatih menggunakan data teks yang telah diberi label sebelumnya. Label untuk teks paling sering diberikan oleh manusia, yang direkrut baik secara internal maupun menggunakan aplikasi *crowd-sourcing*. Pemberian label oleh manusia pada informasi teks yang bersifat subjektif merupakan hal yang menantang, mahal, dan memakan waktu, terutama jika dilakukan dalam skala besar. Pada saat yang sama, semakin besar himpunan data yang diberi label, makin baik hasil klasifikasinya (Haralabopoulos, et al., 2021).

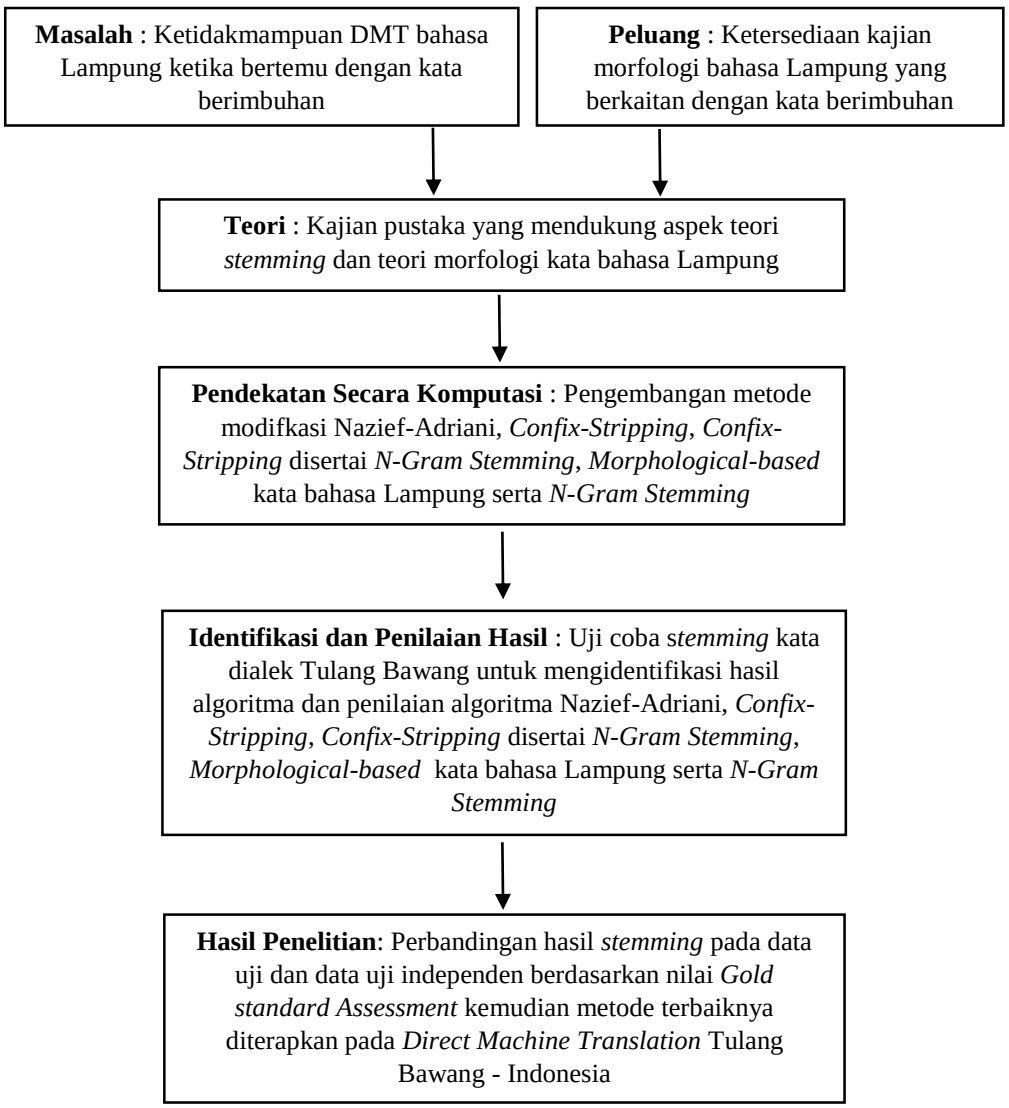
TDA dengan metode permutasi dapat diterapkan beberapa kali per *data set* tanpa menduplikasi entri, sehingga menghindari *overfitting*, berbeda dengan metode yang diusulkan sebelumnya, seperti penataan ulang token secara acak (Wei & Zou, 2019). Pertimbangkan suatu kalimat dengan n jumlah istilah $t_1\ t_2 \ldots t_n$. Metode augmentasi permutasi yang diusulkan bertujuan untuk mempertahankan semua

properti statistik dari *data set* dan menjaga informasi yang terkandung dalam suatu kalimat (Haralabopoulos, et al., 2021). Permutasi dari setiap kalimat disusun ulang sebanyak *n*! kali, di mana *n* adalah jumlah minimum istilah dalam suatu kalimat dari korpus. Ini memastikan bahwa setiap kalimat dilakukan permutasi secara merata dan sifat-sifat statistik utama tetap terjaga (Haralabopoulos, et al., 2021).

III. DESAIN RISET DAN METODE PENELITIAN

3.1 Desain Riset

Desain riset NLP bahasa Lampung berfokus pada dua topik yaitu *word stemming* (WS) dialek Tulang Bawang (TB) yang divisualisasikan pada Gambar 3.1 dan *text data augmentation* (TDA) kalimat dialek *Api* yang divisualisasikan pada Gambar 3.2.



Gambar 3.1 Desain penelitian *stemming* kata dialek Tulang Bawang.

Penjelasan desain riset pada Gambar 3.1 :

1. Masalah dan Peluang

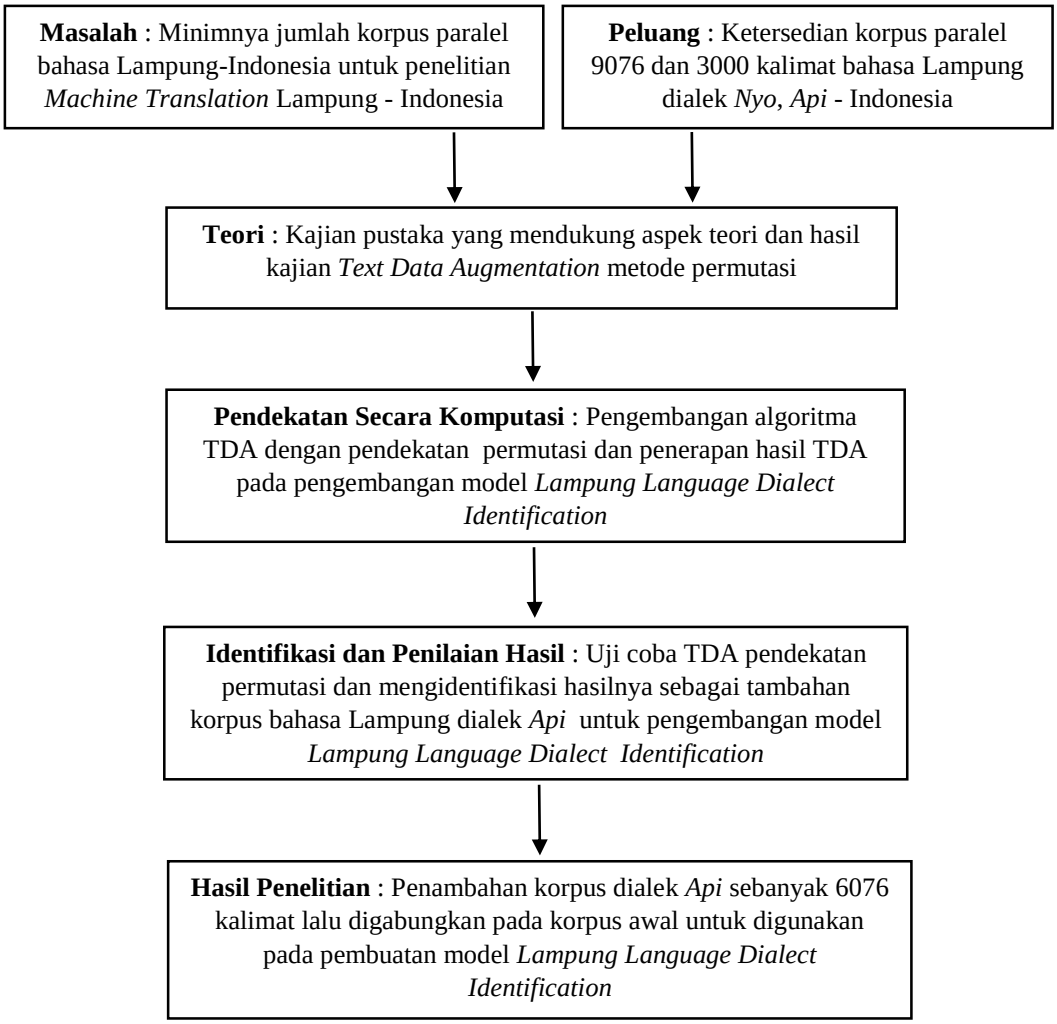
   Masalah yang akan diselesaikan pada penelitian ini adalah *stemming* kata dialek Tulang Bawang. Bahasa Lampung memiliki kajian morfologi yang telah terstandarkan pada buku Afiksasi Verba Bahasa Lampung, buku Morfologi Bahasa Lampung Dialek Tulang Bawang dan Kamus Bahasa Lampung edisi kedua yang diterbitkan oleh Kantor Bahasa Provinsi Lampung.
2. Teori

   Referensi terkait teori *stemming* dan kajian *stemming* pada berbagai bahasa daerah di Indonesia ditemukan sebanyak 34 publikasi dan satu tesis *stemming* bahasa daerah serta ketersediaan teori morfologi kata bahasa Lampung.
3. Metode secara Komputasi

   Metode *stemming* yang dilakukan yaitu diawali dengan modifikasi metode Nazief-Adriani, *Confix-Stripping*, *Confix-Stripping* disertai *N-Gram Stemming*, *Morphological-based* pada kata bahasa Lampung serta *N-Gram Stemming*.
4. Identifikasi dan Penilaian Hasil

   Uji coba *stemming* dialek Tulang Bawang dilakukan untuk mengidentifikasi hasil metode dan penilaian modifikasi metode Nazief-Adriani, *Confix-Stripping*, *Confix-Stripping* disertai *N-Gram Stemming*, *Morphological-based* pada kata bahasa Lampung serta *N-Gram Stemming*. Model yang dibuat berbentuk kode program Python untuk *stemming* kata dialek Tulang Bawang dan dinilai berdasarkan nilai *gold standar assesment* (GSA).
5. Hasil Penelitian

   Hasil *stemming* dilihat melalui perbandingan hasil *stemming* kata dan kata rujukan sebagai penerapan *Gold standard Assessment* pada *stemming* hasil modifikasi metode Nazief-Adriani, *Confix-Stripping*, *Confix-Stripping* disertai *N-Gram Stemming*, *Morphological-based* kata bahasa Lampung dan *N-Gram Stemming* pada kata dialek Tulang Bawang. Model, berbentuk kode program Python, *stemming* terbaik disematkan pada bagian *text preprocessing* DMT Tulang Bawang – Indonesia dan dinilai berdasarkan *BLEU score*.



Gambar 3.2 Desain penelitian TDA metode permutasi pada dialek *Api*.

Penjelasan desain riset pada Gambar 3.2 :

1. Masalah dan Peluang

   Masalah yang diselesaikan pada penelitian ini adalah membangkitkan kalimat dialek *Api* secara otomatis dengan melakukan TDA dengan metode permutasi. Ketersediaan 9076 dan 3000 kalimat bahasa Lampung dialek *Nyo* dan *Api* serta kamus Bahasa Lampung, dari hasil penelitian sebelumnya, memungkinkan untuk melakukan eksperimen TDA.

2. Teori

   Referensi terkait teori TDA serta didukung oleh publikasi TDA pada bahasa Indonesia sebanyak 10 publikasi dan dua publikasi TDA pada bahasa Sunda dan Madura.

3. Metode secara Komputasi

   Metode TDA yang dilakukan yaitu pengembangan TDA metode permutasi kalimat pada dialek *Api*.
4. Identifikasi dan Penilaian Hasil

   Uji coba TDA metode permutasi dan mengidentifikasi hasilnya sebagai tambahan korpus bahasa Lampung dialek *Api* untuk pengembangan model *Lampung Language Dialect Identification*.
5. Hasil Penelitian

   Penambahan korpus dialek *Api* sebanyak 6076 kalimat lalu digabungkan pada korpus awal untuk digunakan pada pembuatan model *Lampung Language Dialect Identification*.

3.2 Modifikasi Metode *Stemming* Dialek Tulang Bawang

Hasil penelusuran secara sistematis dari beragam publikasi terkait *stemming* atau *lemmmatization* pada berbagai kata bahasa daerah di Indonesia menunjukkan bahwa *stemming* dengan metode Nazief-Adriani pada kata bahasa Indonesia banyak digunakan sebagai rujukan awal kemudian diikuti oleh metode *Confix-Stripping* sedangkan beberapa bahasa daerah lain memilih menggunakan metode *Morphological-based* yang dibangun sesuai kaidah bahasa daerahnya sendiri. *N-Gram stemming* untuk mengatasi kata yang gagal dalam *stemming* atau digabungkan pada metode sebelumnya.

Dua fakta tersebut memperkuat landasan bahwa dalam melakukan modifikasi berbagai metode untuk *stemming* kata dialek Tulang Bawang (TB) bisa dimulai dari modifikasi metode Nazief-Adriani, *Confix-Stripping*, *Confix-Stripping* disertai *N-Gram Stemming*, *Morphological-based* kata bahasa Lampung dan *stemming* dengan metode *N-Gram Stemming*. Kajian morfologi bahasa Lampung menjadi hal utama sebelum masuk dalam pembahasan *stemming* karena dengan kajian morfologi para peneliti dapat memahami afiksasi yang ada pada bahasa Lampung.

3.2.1 Modifikasi Nazief-Adriani untuk Dialek Tulang Bawang

Modifikasi Nazief-Adiani digambarkan melalui *flowchart* pada Gambar 3.3. Urutan langkah modifikasi metode Nazief-Adriani untuk kata dialek Tulang Bawang (TB) adalah sebagai berikut:

Langkah 1.

Masukkan kata yang akan dilakukan *stemming*. Lalu kata tersebut dicari atau dicek di dalam kamus. Jika kata tersebut ditemukan di dalam kamus, diasumsikan kata tersebut adalah kata dasar dan metode berhenti, jika tidak proses lanjut ke Langkah 2.

Langkah 2.

Lakukan proses *stemming* pada sufiks infleksi {*"kah", "lah", "nou", "meu", "keu", "no", "ko"*}. atau sufiks derivasional {"*an*", "*ken*", "*ei*"} untuk dihapus. Jika ini berhasil, hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 3.

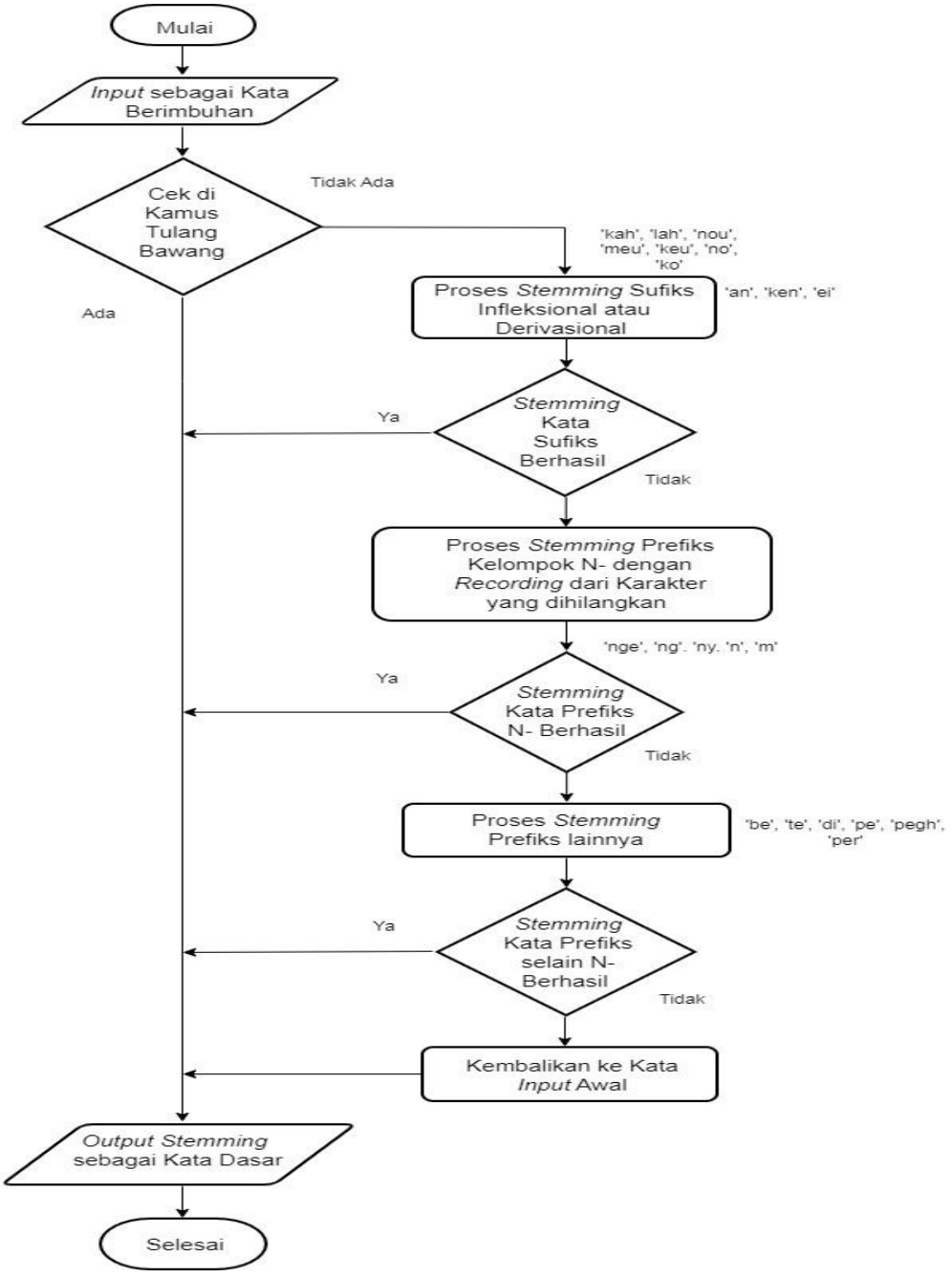
Langkah 3.

Lakukan proses pada prefiks pada kata yang akan dilakukan *stemming* dengan memperhatikan beberapa Langkah berikut:

a. Prioritas satu adalah memproses kata yang mengandung prefiks kelompok N- = {*"nge", "ng", "ny", "n", "m"*} dan lakukan proses *stemming* sesuai aturan Tabel 2.15, jika berhasil maka diperoleh kata dasarnya jika tidak lakukan Langkah b. Pada bagian ini tersedia mekanisme *recording* yaitu memberikan karakter tertentu yang hilang akibat proses *stemming* pada prefiks peN atau N- sesuai acuan pada Tabel 2.15.
b. Prioritas selanjutnya adalah memproses kata yang mengandung prefiks {*"be", "te", "di", "pe", "pegh", "per"*}, jika berhasil maka diperoleh kata dasarnya jika tidak maka proses lanjut ke Langkah 4.

Langkah 4.

Jika Langkah 1, 2, dan 3 gagal maka metode atau program akan mengembalikan kata awal sebagai *input* menjadi *output*.



Gambar 3.3 *Flowchart* Modifikasi Nazief-Adriani untuk Dialek Tulang Bawang

Penjelasan rinci mengenai metode Nazief-Adriani dalam bahasa Indonesia dapat ditemukan pada naskah publikasi yang ditulis oleh Jelita Asian, dkk (Asian, et al., 2005). Modifikasi Nazief-Adriani untuk kata dialek Tulang Bawang bergantung

pada aturan morfologi yang luas yang mengkategorikan dan mencakup imbuhan yang diizinkan. Teknik ini juga memfasilitasi perekaman, mengembalikan huruf awal yang dihilangkan dari kata dasar sebelum menambahkan prefiks. Selain itu, teknik ini menggunakan kamus kata dasar tambahan pada berbagai tahap untuk memverifikasi apakah proses *stemming* telah berhasil mengidentifikasi kata dasar.

3.2.2 Modifikasi *Confix-Stripping* untuk Dialek Tulang Bawang

Modifikasi *Confix-Stripping* (CS) digambarkan melalui *flowchart* pada Gambar 3.4. Urutan langkah modifikasi metode *Confix-Stripping* untuk kata dialek TB adalah sebagai berikut:

Langkah 1.

Masukkan kata yang akan dilakukan *stemming*. Lalu kata tersebut dicari atau dicek di dalam kamus. Jika kata tersebut ditemukan di dalam kamus, diasumsikan kata tersebut adalah kata dasar dan metode CS berhenti, jika tidak proses lanjut ke Langkah 2.

Langkah 2.

Lakukan proses *stemming* pada kata yang mengandung perulangan atau reduplikasi. Reduplikasi utuh atau reduplikasi sebagian atau reduplikasi yang mendapatkan penambahan afiks atau reduplikasi dwipurwa. Jika ini berhasil, hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 3.

Langkah 3.

Lakukan proses *stemming* pada sufiks infleksi {*“kah”, “lah”, “nou”, “meu”, “keu”, “no”, “ko”*}. atau sufiks derivasional {“an”, “ken”, “ei”, “ko”} untuk dihapus. Jika ini berhasil, hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 4.

Langkah 4.

Lakukan proses pada prefiks pada kata yang akan dilakukan *stemming* dengan memperhatikan beberapa langkah berikut:

a. Prioritas satu adalah memproses kata yang mengandung prefiks kelompok N- = {*"nge", "ng", "ny", "n", "m"*} dan lakukan proses *stemming* sesuai aturan Tabel 2.15. Jika berhasil maka diperoleh kata dasarnya jika tidak lakukan Langkah b. Pada bagian ini tersedia mekanisme *recording* yaitu memberikan karakter tertentu yang hilang akibat proses *stemming* pada prefiks peN atau N- sesuai acuan pada Tabel 2.15.
b. Prioritas selanjutnya adalah memproses kata yang mengandung prefiks {*"be", "te", "di", "pe", "pegh", "per"*}, jika berhasil maka diperoleh kata dasarnya jika tidak maka lanjutkan ke Langkah 5.

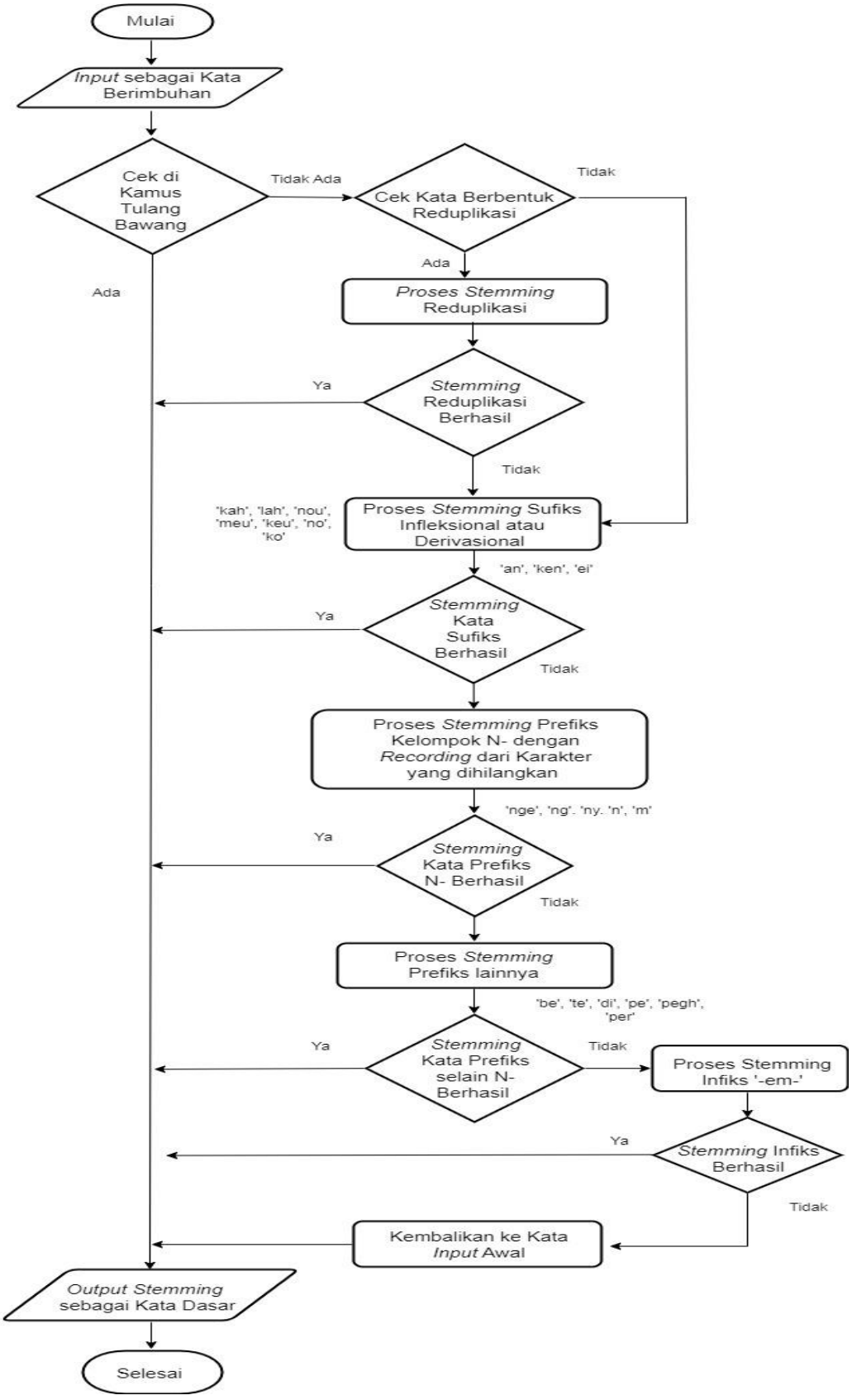
Langkah 5.

Lakukan proses infiks yaitu melihat suatu kata yaitu dengan menghapus urutan karakter kedua dan ketiga mengandung karakter infiks "-em-". Gabungkan karakter awal dan sisanya, cek hasil penggabungan apakah membentuk kata dasar yang ada di kamus jika iya sisa karakter sebagai kata dasarnya jika tidak maka proses lanjut ke Langkah 6.

Langkah 6.

Jika Langkah 1, 2, 3, 4, dan 5 gagal maka metode mengembalikan kata awal sebagai *input* menjadi *output*.

Penjelasan rinci mengenai metode *Confix-stripping* dalam bahasa Indonesia dapat dilihat pada makalah yang ditulis oleh Adriani dkk (Adriani et al., 2007). Modifikasi *Confix-stripping* untuk dialek Tulang Bawang dilakukan dengan menambahkan proses pengecekan reduplikasi dan infiks sebagai bentuk diferensiasi dari metode Nazief-Adriani. Metode *Confix-stripping* juga memfasilitasi perekaman, mengembalikan huruf awal yang dihilangkan dari kata dasar sebelum menambahkan prefiks. Selain itu, teknik ini menggunakan kamus kata dasar tambahan pada berbagai tahap untuk memverifikasi apakah proses stemming telah berhasil mengidentifikasi kata dasar.



Gambar 3.4 *Flowchart* Modifikasi *Confix-Stripping* untuk Dialek Tulang Bawang

3.2.3 Modifikasi *Confix-Stripping* disertai *N-Gram Stemming* untuk Dialek Tulang Bawang

Modifikasi *Confix-Stripping* disertai *N-Gram Stemming* digambarkan melalui *flowchart* pada Gambar 3.5. Urutan Langkah modifikasi metode *Confix-Stripping* disertai *N-Gram Stemming* untuk kata dialek TB sebagai berikut :

Langkah 1.

Masukkan kata yang akan dilakukan *stemming*. Lalu kata tersebut dicari atau dicek di dalam kamus. Jika kata tersebut ditemukan di dalam kamus, diasumsikan kata tersebut adalah kata dasar dan metode berhenti, jika tidak proses lanjut ke Langkah 2.

Langkah 2.

Lakukan proses *stemming* pada kata yang mengandung perulangan atau reduplikasi. Reduplikasi utuh atau reduplikasi sebagian atau reduplikasi yang mendapatkan penambahan afiks atau reduplikasi dwipurwa. Jika ini berhasil, hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 3.

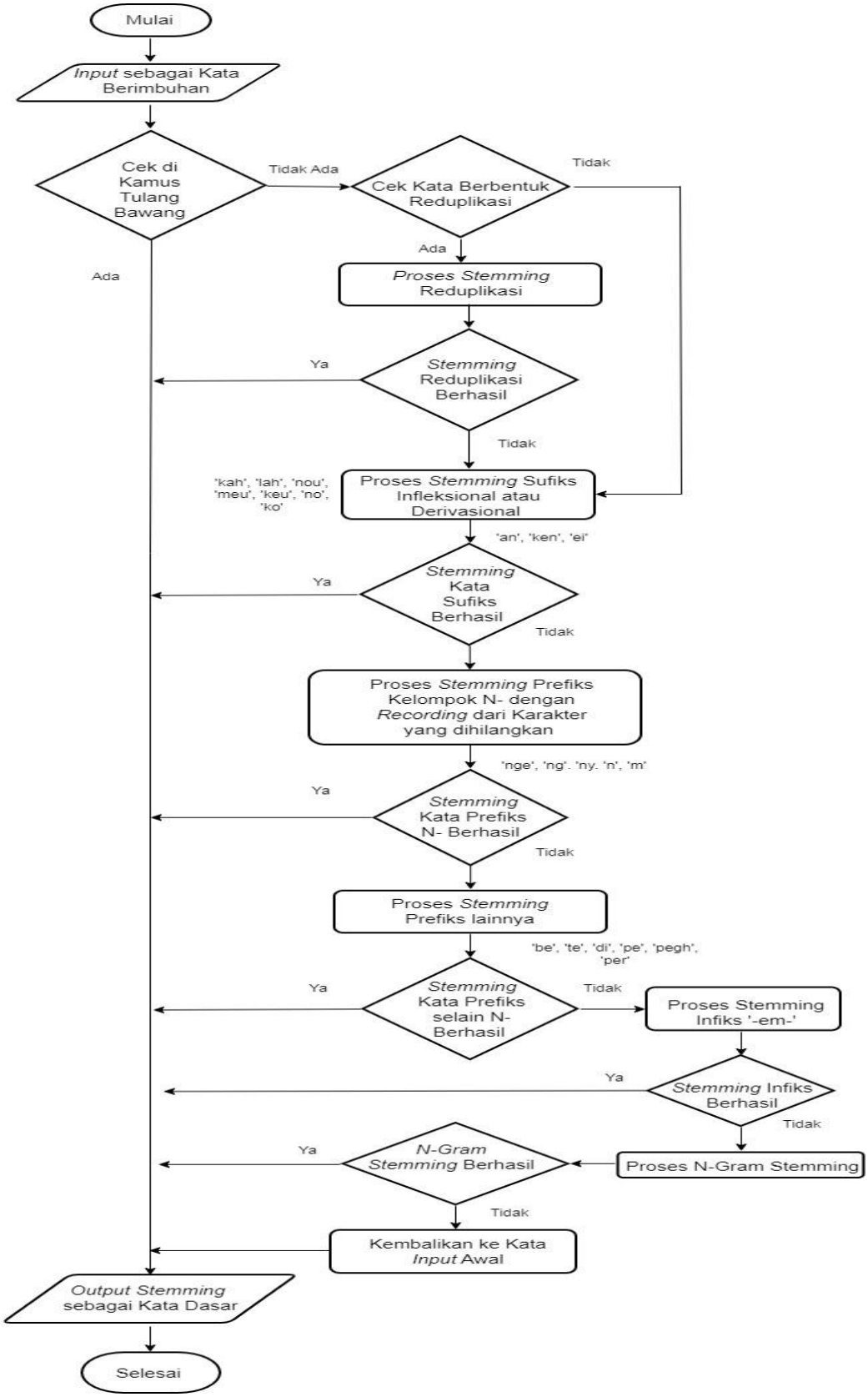
Langkah 3.

Lakukan proses *stemming* baik pada sufiks infleksi {*"kah", "lah", "nou", "meu", "keu", "no", "ko"*} atau sufiks derivasional {"an", "ken", "ei", "ko"} untuk dihapus. Jika ini berhasil, hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 4.

Langkah 4.

Lakukan proses pada prefiks pada kata yang akan dilakukan *stemming* dengan memperhatikan beberapa langkah berikut:

a. Prioritas satu adalah memproses kata yang mengandung prefiks kelompok N- = {*"nge", "ng", "ny", "n", "m"*} dan lakukan proses stemming sesuai Tabel 2.15, jika berhasil maka diproleh kata dasarnya jika tidak lakukan Langkah b. Pada bagian ini tersedia mekanisme *reco-*



Gambar 3.5 *Flow Chart* Modifikasi *Confix-Stripping* disertai *N-Gram Stemming* untuk Dialek Tulang Bawang

rding yaitu memberikan karakter tertentu yang hilang akibat proses *stemming* pada prefiks peN atau N- sesuai acuan pada Tabel 2.15.

b. Prioritas selanjutnya adalah memproses kata yang mengandung prefiks { *"be", "te", "di", "pe", "pegh", "per"*}, jika berhasil maka diperoleh kata dasarnya jika tidak maka selesai lanjut ke Langkah 5.

Langkah 5.

Lakukan proses infiks yaitu melihat suatu kata yaitu dengan menghapus urutan karakter kedua dan ketiga mengandung karakter infiks "-em". Gabungkan karakter awal dan sisanya, cek hasil penggabungan apakah membentuk kata dasar yang ada di kamus jika iya sisa karakter sebagai kata dasarnya jika tidak maka proses lanjut ke Langkah 6.

Langkah 6.

Lakukan proses *string similarity* menggunakan metode *n*-gram *stemming*, apabila tingkat kemiripannya memenuhi nilai ambang batas maka kata dasar ditampilkan. Jika belum berhasil lanjutkan ke Langkah 7.

Langkah 7.

Jika Langkah 1, 2, 3, 4, 5 dan 6 gagal maka metode mengembalikan kata awal sebagai *input* menjadi *output*.

3.2.4 Metode *Morphological-based* untuk Kata Dialek Tulang Bawang

Metode *Morphological-based stemming* kata dialek Tulang Bawang adalah metode CS dengan penambahan proses konfiks setelah pengecekan reduplikasi. Detail tabel konfiks disajikan pada Tabel 3.1.

Urutan langkah modifikasi metode *Morphological-based* untuk kata dialek Tulang Bawang sebagai berikut :

Langkah 1.

Masukkan kata yang akan dilakukan *stemming*. Kata yang akan dilakukan *stemming* dicari atau dicek di dalam kamus, jika ditemukan di dalam kamus, diasumsikan kata

tersebut adalah kata dasar, sehingga kata dikembalikan dan metode berhenti, jika tidak lanjutkan ke Langkah 2.

Tabel 3.1 Aturan Konfiks pada Dialek Tulang Bawang

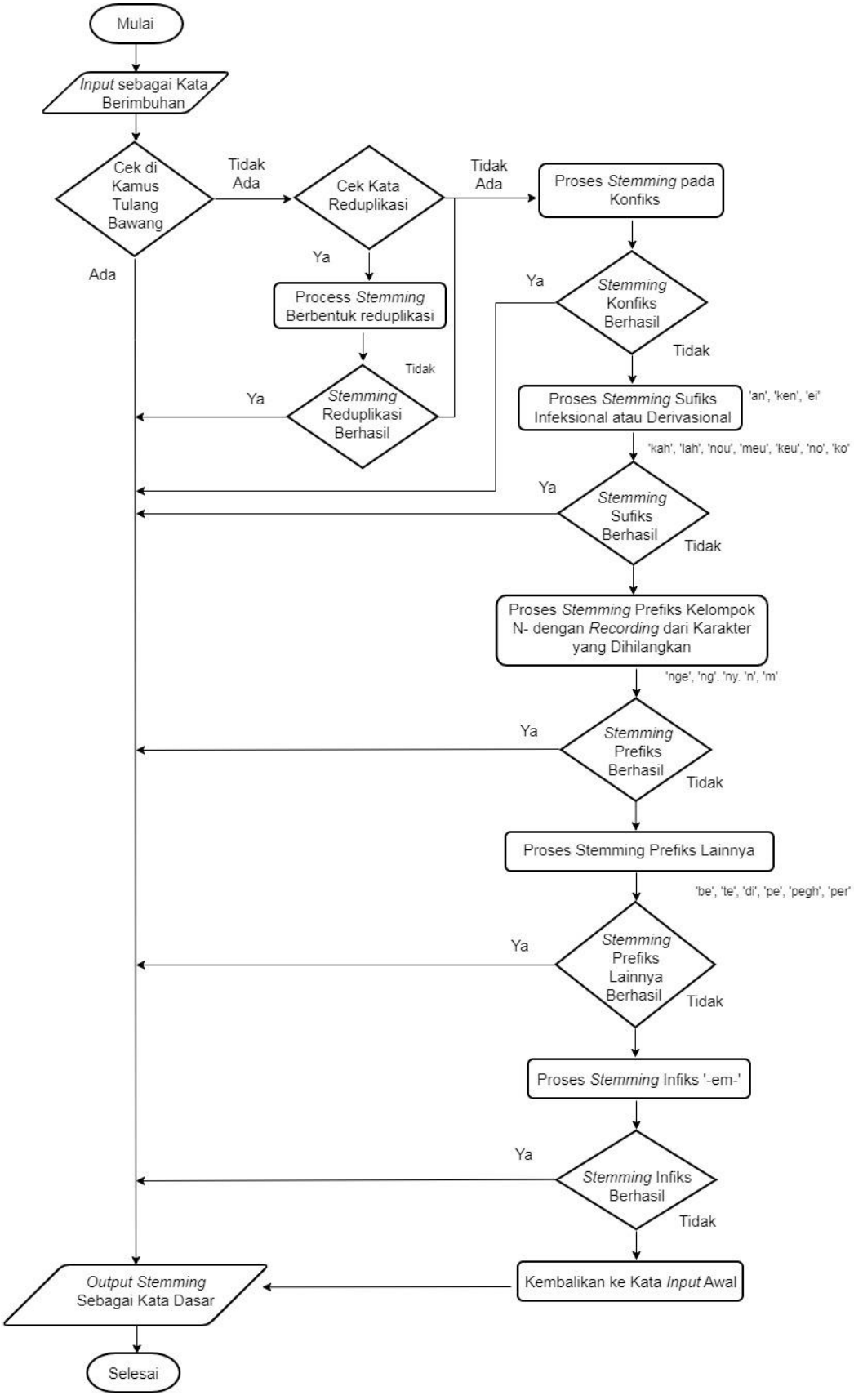
Aturan	Pola Konstruksi Konfiks	Pola Penghapusan Konfiks
1	nge{b|d|g|h|j|gh|l|w}AA...+ -ken	nge- +{b|d|g|h|j|gh|l|w}AA... + -ken
2	ng{a|i|u|e|o}CV... + -ken	ng- +{a|i|u|e|o}CV... + -ken | ng- + k{a|i|u|e|o}CV... + -ken
3	nyVC... + -ken	ny- + cVC... + -ken | ny- + sVC... + -ken
4	nVC... + -ken	n- + tVC... + -ken
5	mVC... + -ken	m- + pVC... + -ken
6	nge{b|d|g|h|j|gh|l|w}AA...+ -ei	nge- +{b|d|g|h|j|gh|l|w}AA... + -ei
7	ng{a|i|u|e|o}CV... + -ei	ng- +{a|i|u|e|o}CV... + -ei | ng- + k{a|i|u|e|o}CV... + -ei
8	nyVC... + -ei	ny- + cVC... + -ei | ny- + sVC... + -ei
9	nVC... + -ei	n- + tVC... + -ei
10	mVC... + -ei	m- + pVC... + -ei
11	beAA... +-ken	be- +AA...+ -ken
12	beAA... +-an	be- +AA...+ -an
13	diAA... +-ken	di- +AA...+ -ken
14	diAA... + -ei	di- +AA...+ -ei
15	perCV... + -ken	per- +CV...+ -ken
16	keCV... + -an	ke- +CV...+ -an

Langkah 2.

Lakukan proses *stemming* pada kata yang mengandung perulangan atau reduplikasi. Reduplikasi utuh atau reduplikasi sebagian atau reduplikasi yang mendapatkan penambahan afiks atau reduplikasi dwipurwa. Jika ini berhasil, hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 3.

Langkah 3.

Lakukan proses *stemming* pada konfiks yaitu *{"nge-ken", "ng-ken", "ny-ken", "n-ken", "m-ken", "nge-ei", "ng-ei", "ny-ei", "n-ei", "m-ei","be-an", "be-ken", "di-ken", "di-ei", "ke-an", "per-ken"}* untuk dihapus sesuai pola yang tertera pada Tabel 3.1. Jika ini berhasil, hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 4.



Gambar 3.6 *Flowchart Stemming* berbasis *Morphological* untuk Dialek Tulang Bawang

Langkah 4.

Lakukan proses *stemming* baik pada sufiks infleksi yaitu sufiks infleksi {*"kah"*, *"lah"*, *"nou"*, *"meu"*, *"keu"*, *"no"*, *"ko"*} atau sufiks derivasional {"an", "ken", "ei", "ko"} untuk dihapus. Jika ini berhasil maka hasilnya adalah sebagai kata dasar jika tidak maka proses lanjut ke Langkah 5.

Langkah 5.

Lakukan proses pada prefiks pada kata yang akan dilakukan *stemming* dengan memperhatikan beberapa langkah berikut:

a. Prioritas satu adalah memproses kata yang mengandung prefiks kelompok N- = {*"nge"*, *"ng"*, *"ny"*, *"n"*, *"m"*} dan lakukan proses stemming sesuai Tabel 2.15, jika berhasil maka diproleh kata dasarnya jika tidak lakukan Langkah b. Pada bagian ini tersedia mekanisme *recording* yaitu memberikan karakter tertentu yang hilang akibat proses stemming pada prefiks peN atau N- sesuai acuan pada Tabel 2.15.
b. Prioritas selanjutnya adalah memproses kata yang mengandung prefiks { *"be"*, *"te"*, *"di"*, *"pe"*, *"pegh"*, *"per"*}, jika berhasil maka diproleh kata dasarnya jika tidak maka selesai lanjutkan ke Langkah 6.

Langkah 6.

Lakukan proses infiks yaitu melihat suatu kata yaitu dengan menghapus urutan karakter kedua dan ketiga mengandung karakter infiks "-em". Gabungkan karakter awal dan sisanya, cek hasil penggabungan apakah membentuk kata dasar yang ada di kamus jika iya sisa karakter sebagai kata dasarnya jika tidak maka proses lanjut ke Langkah 7.

Langkah 7.

Jika Langkah 1, 2, 3, 4, 5 dan 6 gagal maka metode mengembalikan kata awal sebagai *input* menjadi *output*.

3.2.5 *N-Gram Stemming* untuk Kata Dialek Tulang Bawang

Berikut ini diberikan prosedur metode *N-Gram stemming* untuk kata bahasa Lampung dialek Tulang Bawang.

1. *Input* Data

   Masukkan kata yang akan dilakukan *stemming*, ubah kata menjadi huruf kecil, tentukan nilai N (ukuran *N-gram* yang diinginkan), siapkan kamus kata dasar untuk perbandingan.

2. Pemberian *Padding*

   Tambahkan karakter *underscore* (_) di awal kata, tambahkan karakter *underscore* (_) di akhir kata, simpan kata yang sudah diberi *padding*

3. Pembuatan *N-gram*
   a. Mulai dari karakter pertama kata yang sudah diberi *padding*, selanjutnya disebut kata1
   b. Ambil N karakter berurutan,
   c. Simpan potongan N karakter sebagai satu *N-gram*,
   d. Geser satu karakter ke kanan
   e. Ulangi Langkah b-d sampai mencapai akhir kata
   f. Kumpulkan semua *N-gram* yang terbentuk

4. Perbandingan dengan Kata Dasar
   a. Ambil satu kata dari kamus kata dasar, selanjutnya disebut kata2
   b. Lakukan proses *padding* pada kata dari kamus
   c. Buat *N-gram* dari kata kamus tersebut
   d. Hitung jumlah *N-gram* yang sama (*intersection*)
   e. Hitung total *N-gram* dari kedua kata
   f. Hitung nilai kemiripan dengan rumus *Dice Coefficient*:
      DC = (2 × jumlah *N-gram* sama) / (total *N-gram* kata1 + total *N-gram* kata2)
   g. Simpan nilai kemiripan dan kata dasar
   h. Ulangi Langkah a-g untuk semua kata dalam kamus

5. Pemilihan Hasil
   a. Tentukan nilai *threshold* minimal (misal 0,5)
   b. Periksa semua nilai kemiripan yang sudah dihitung
   c. Pilih kata dasar dengan nilai kemiripan tertinggi
   d. Bandingkan nilai kemiripan tertinggi dengan *threshold*
   e. Jika nilai kemiripan > *threshold*:

Gunakan kata dasar terpilih sebagai hasil *stemming*

f. Jika nilai kemiripan ≤ *threshold*:

Gunakan kata asli sebagai hasil *stemming*

6. Hasil atau *Output*
   a. Tampilkan kata hasil *stemming*
   b. Tampilkan nilai kemiripan (opsional)
   c. Selesai

Contoh:

Kata yang akan di-*stemming* yaitu "pamelok", N = 3 (*trigram*)

Padding: "*_pamelok_*"

Trigram: ["*_pa", "pam", "ame", "mel", "elo", "lok*", “ok_”]

Bandingkan dengan kata dasar "melok":

Kata dasar dengan *padding*: "*melok*"

Trigram kata dasar: ["*_me", "mel", "elo", "lok*", “ok_”]

N-gram sama: [*"mel", "elo", "lok*", “ok_”]

DC = (2 × 4) / (7 + 5) = 8/12= 0.67

Jika 0.67 > *threshold*, maka hasil *stemming* adalah "melok"

Kata yang akan di-*stemming* yaitu "pamelok", N = 2 (*Bigram*)

Padding: "*_pamelok_*"

Bigram: ["*_p", "pa", "am", "me", "el", "lo*", “ok”, “k_”]

Bandingkan dengan kata dasar "melok":

Kata dasar dengan *padding*: "*melok*"

Bigram kata dasar: ["*_m", "me", "el", "lo*", “ok”, “k_”]

N-gram sama: [*"me", "el", "lo*", “ok”, “k_”]

DC = (2 × 5) / (8+6) = 10/14= 0,71

Jika 0,71 > *threshold*, maka hasil *stemming* adalah "melok"

Perbandingan hasil *N-Gram stemming* dengan nilai *threshold* 0.5 pada *Trigram* diperoleh nilai 0.67 > *threshold* sedangkan pada *Bigram* diperoleh nilai 0.71 > *threshold*.

3.3 *Text Data Augmentation* Metode Permutasi

Ekperimen memperbanyak kalimat teks latin bahasa Lampung secara otomatis menggunakan komputer belum pernah dilakukan. Minimnya distribusi mono korpus tulisan atau teks kalimat bahasa Lampung di internet menjadi faktor utama belum pernah ada penelitian tentang TDA bahasa Lampung. Adapun upaya yang pernah dilakukan pada tahun 2017 yaitu pengetikan teks bahasa Lampung dan terjemahannya dilakukan secara manual (dengan mengetik ulang bersumber dari buku pelajaran bahasa Lampung tingkat SD dan SMP) guna kebutuhan riset tesis S2 penulis di STEI ITB Bandung sebanyak 3000 kalimat bahasa Lampung dialek *Api* dan *Nyo* berikut terjemahannya.

Sebagai bagian dari tahapan awal penelitian TDA serta sesuai kondisi saat ini pada bahasa Lampung maka eksperimen yang paling memungkinkan adalah dilakukan augmentasi data teks kalimat bahasa Lampung dengan metode permutasi (Haralabopoulos, et al., 2021). Metode ini melakukan simulasi pembangkitan teks kalimat bahasa Lampung secara otomatis, misalkan untuk kalimat berpola subjek predikat objek (SPO). Langkah-Langkah eksperimen permutasi untuk kalimat berpola subjek predikat objek (SPO) sebagai berikut :

a. Sediakan suatu kalimat berpola SPO. Misalkan "*nyak nginum teh"*. *nyak* sebagai subjek, *nginum* sebagai predikat dan *teh* sebagai objek. Jika dilakukan tokenisasi pada kalimat "*nyak nginum teh"* maka diperoleh tiga *token* yaitu {"*nyak", "nginum", "teh"*}.
b. Lakukan permutasi pada kalimat "*nyak nginum teh",* artinya ada tiga token yang akan menempati posisi 1 yaitu n_1, posisi 2 yaitu n_2 dan posisi 3 yaitu n_3. Posisi 1 berpotensi diisi oleh 3 *token*, posisi 2 berpotensi diisi oleh 2 *token* dan posisi 3 berpotensi diisi oleh 1 *token* ditulis dalam simbol $n_1 x n_2 x n_3 = 3x2x1 = 6$.
c. Tuliskan hasil semua hasil permutasi dengan susunan sebagai berikut: {{*nyak, nginum, teh*}, {*nyak, teh, nginum*}, {*nginum, nyak, teh*}, {*nginum, teh, nyak*}, {*teh, nginum, nyak*}, {*teh, nyak, nginum*}}.
d. Susun ulang dan tulis sebagai suatu kalimat. Hasil dari pemutasi {*nyak, nginum, teh*} sebagai berikut:

nyak nginum teh,
nyak teh nginum,
nginum nyak teh,
nginum teh nyak,
teh nginum nyak,
teh nyak nginum

Saat ini *data set* yang tersedia untuk melakukan penelitian augmentasi data teks bahasa Lampung adalah sebanyak 3000 kalimat dialek *Api* dan 9076 kalimat dialek *Nyo*. Langkah-Langkah eksperimen yang dapat dilakukan yaitu : (1) membangkitkan 6076 kalimat dialek Api dengan metode permutasi agar *data set* dialek *Api* dan dialek *Nyo* menjadi seimbang, (2) membangun model *Lampung Language Dialect Identification* (LLDI) menggunakan metode *Naive Bayes*, *Support Vector Machine*, *Logistic Regression*, *Random Forest*. Eksperimen dilakukan pada dua kondisi yaitu data *imbalanced* dan data *balanced*, (3) membandingkan hasil eksperimen dua kondisi tersebut melalui pengamatan nilai *accuracy*, *precision, recall* dan *F1-score*.

3.3.1 Membangkitkan 6076 Kalimat Dialek *Api*

Fakta awal pada ekperimen membangun model LLDI dengan *data set* berupa 3000 kalimat dialek *Api* dan 9076 kalimat dialek *Nyo* adalah terdapat selisih sebesar 6076 kalimat antara kedua dialek tersebut. Pada *data set* 3000 kalimat dialek *Api*, peneliti melakukan pengamatan secara seksama secara manual untuk memilih sampel kalimat yang dapat digunakan untuk membangkitkan 6076 kalimat. Tabel 3.2 menunjukkan sampel kalimat berikut jumlah potensi kalimat yang dapat dibangkitkan secara otomatis dengan metode permutasi. Berikut contoh upaya membangun kalimat berpola subjek predikat (SP) atau kalimat berpola subjek predikat objek (SPO) atau kalimat berpola subjek predikat objek keterangan (SPOK).

Tabel 3.2 Sampel Kalimat Dialek *Api* yang Berpotensi dibuat TDA dengan Permutasi

No	Sampel Kalimat	Jumlah Kata	Potensi Jumlah Permutasi yang Dapat Dibuat
1	*abah macul di sabah*	4	24
2	*abang adi maccul paghit*	4	24
3	*abang belajagh di peghpustakaan*	4	24
4	*abang diughau ia mak nyahut*	5	120

5	*abang macul di sabah*	4	24
6	*adik lagi nginum susu*	4	24
7	*adik miwang ditepagh kaka*	4	24
8	*adik miwang kilui mubil-mubilan*	4	24
9	*adik miwang ulah keawesan*	4	24
10	*adik ngusung buku*	3	6
11	*adik nyimpen buku di lemaghi*	5	120
12	*adik tanom kikim*	3	6
13	*adikku gelaghni hilda*	3	6
14	*adikku si kusayangi*	3	6
15	*adin ngejamukko buku bahasa lampungku*	5	120
16	*adu bingi nyak haga mulang*	5	120
17	*agegali pakai tembilang*	3	6
18	*ah mak ApiApi*	3	6
19	*ahmad murid kelas iv*	4	24
20	*ahmad nugal huma*	3	6
21	*akuk wai pakai ngebasuh pungu*	5	120
22	*akukko nyak wai*	3	6
23	*akukko nyak wai aus temon*	5	120
24	*alamat sikam di kampung baru*	5	120
25	*alamat sikam gegoh jama boy*	5	120
26	*aLangkah bangik ni hughik gham*	5	120
27	*aLangkah indahni desaku*	3	6
28	*ali lapah mit sekula*	4	24
29	*amak lagi bubalah jama mamak*	5	120
30	*amak ngosegh putti di sabah*	5	120
31	*amakni sai ngeracuni anak kandungni*	5	120
32	*aminah lagi nyapu*	3	6
33	*aminah ngiyau pighing*	3	6
34	*aminah nulis sughat bakal ghikni*	5	120
35	*aminah nyambel delan*	3	6
36	*amini mekik ngaliak maling*	4	24
37	*amir mak sekula ummi*	4	24
38	*amir ninjuk manuk haga ditikol*	5	120
39	*amun sekula pakailah sepatu*	4	24
40	*anakanak di sekula ku pandaipandai*	5	120
41	*anakni ngelawan hulun tuhani*	4	24
42	*anakni nyukak jama inakni*	4	24
43	*anakni sai tuha begeghal lukman*	5	120
44	*andahni janji mungkeri sakik nihan hatiku sinji*	7	5040
45	*angahkon banguk mu*	3	6
46	*ani tulung akukko wai sina*	5	120
47	*apak di sabah*	3	6
48	*apak ghisok ngejala di kulam*	5	120

49	*apak nanom paghi di sabah*	5	120
50	*apak ngusung deghian*	3	6

3.3.2 Membangun Model *Lampung Language Dialect Identification*

Langkah-langkah dalam membangun model *Lampung Language Dialect Identification* dengan metode *Naive Bayes* (Eisenstein, 2019; Jurafsky & Martin, 2026; Kedia & Rasu, 2020) :

1. Sediakan *Data set* berupa 3000 kalimat bahasa Lampung dialek *Api* dan 9076 kalimat dialek *Nyo*
2. Lakukan Ekstraksi Fitur dari *data training*
   - *Character n-gram* (n=1,2,3)
     a. *Unigram*: mengekstrak karakter tunggal (a, b, c)
     b. *Bigram*: mengekstrak 2 karakter berurutan (ab, bc)
     c. *Trigram*: mengekstrak 3 karakter berurutan (abc)
   - *Word n-gram* (n=1,2)
     a. *Unigram*: mengekstrak kata tunggal
     b. *Bigram*: mengekstrak 2 kata berurutan
3. Perhitungan TF-IDF
   - *Term Frequency* (TF): hitung frekuensi kemunculan setiap fitur
   - *Document Frequency* (DF): hitung jumlah dokumen yang memuat fitur
   - IDF = log(N/DF), N = jumlah total dokumen
   - TF-IDF = TF × IDF
4. Pemodelan *Naive Bayes*
   - Hitung *Prior Probability* setiap kelas (Dialek)
   - Hitung *Conditional Probability* dengan *Laplace smoothing*
   - P(kata|kelas) = (*count* + 1)/(total_kata + *vocab_size*)
5. Ujicoba Klasifikasi Teks Baru
   - Ekstrak fitur dari teks uji
   - Hitung probabilitas untuk setiap kelas
   - Normalisasi probabilitas
   - Pilih kelas dengan probabilitas tertinggi
6. Evaluasi dan Analisis Hasil

- Identifikasi kekuatan dan kelemahan model
- Analisis hasil klasifikasi

Langkah-langkah dalam membangun model *Lampung Language Dialect Identification* dengan metode *Logistic Regression* (Eisenstein, 2019; Kedia & Rasu, 2020) :

1. Sediakan *Data set* berupa 3000 kalimat bahasa Lampung dialek *Api* dan 9076 kalimat dialek *Nyo*
2. Lakukan Ekstraksi Fitur dari *data training*
   - *Character n-gram* (n=1,2,3)
     a. *Unigram*: mengekstrak karakter tunggal (a, b, c)
     b. *Bigram*: mengekstrak 2 karakter berurutan (ab, bc)
     c. *Trigram*: mengekstrak 3 karakter berurutan (abc)
   - *Word n-gram* (n=1,2)
     a. *Unigram*: mengekstrak kata tunggal
     b. *Bigram*: mengekstrak 2 kata berurutan
3. Perhitungan TF-IDF
   - *Term Frequency* (TF): hitung frekuensi kemunculan setiap fitur
   - *Document Frequency* (DF): hitung jumlah dokumen yang memuat fitur
   - IDF = log(N/DF), N = jumlah total dokumen
   - TF-IDF = TF × IDF
4. Pemodelan *Logistic Regression*
   - Setiap teks diubah menjadi vektor fitur berdasarkan nilai TF-IDF dari *character n-gram* dan *word n-gram* yang diekstrak.
   - *Logistic regression* menghitung probabilitas bahwa teks termasuk dalam kelas *Api* menggunakan fungsi *sigmoid*:

$$P(Api|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}}$$

Dimana :

$x_1, x_2, x_3, \ldots, x_n$ adalah nilai TF-IDF dari fitur.

$\beta_0, \beta_1, \ldots, \beta_n$ adalah parameter model yang dipelajari selama pelatihan.

- Model dilatih untuk menyesuaikan parameter β agar teks dari dialek *Api* memiliki probabilitas mendekati 1, dan teks dari dialek *Nyo* mendekati 0.

5. Ujicoba atau Prediksi Identifikasi Teks Baru
   - Ekstrak fitur *character n-gram* (n=1,2,3) dan *word n-gram* (n=1,2) dari teks tersebut.
   - Hitung nilai TF-IDF untuk fitur-fitur tersebut berdasarkan *data training*.
   - Masukkan vektor TF-IDF ke dalam model *Logistic Regression*.
   - Model menghitung probabilitas $P(Api|x)$
   - Klasifikasikan teks berdasarkan aturan:
     - Jika $P(Api|x) > 0.5$, teks diklasifikasikan sebagai dialek *Api*.
     - Jika $P(Api|x) \leq 0.5$, teks diklasifikasikan sebagai dialek *Nyo*.
6. Evaluasi dan Analisis Hasil
   - Identifikasi kekuatan dan kelemahan model
   - Analisis hasil klasifikasi

Logistic Regression bekerja pada *dialect identification* untuk dialek *Api* dan *Nyo* dengan cara:

1. Mengekstrak fitur dari teks menggunakan *character n-gram* (*unigram*, *bigram*, *trigram*) dan *word n-gram* (*unigram*, *bigram*).
2. Menghitung bobot fitur dengan TF-IDF untuk menentukan fitur yang paling relevan.
3. Melatih model untuk memprediksi probabilitas kelas berdasarkan vektor TF-IDF.
4. Mengklasifikasikan teks baru sebagai *Api* atau *Nyo* berdasarkan probabilitas yang dihasilkan.

Metode ini memanfaatkan perbedaan pola karakter dan kata antara dialek *Api* dan *Nyo* untuk melakukan identifikasi yang akurat.

Langkah-langkah dalam membangun model *Lampung Language Dialect Identification* dengan metode *Support Vector Machine* (Géron, 2022; Kedia & Rasu, 2020) :

1. Sediakan *Data set* berupa 3000 kalimat bahasa Lampung dialek *Api* dan 9076 kalimat dialek *Nyo*
2. Lakukan Ekstraksi Fitur dari *data training*
   - *Character n-gram* (n=1,2,3)
     a. *Unigram*: mengekstrak karakter tunggal (a, b, c)
     b. *Bigram*: mengekstrak 2 karakter berurutan (ab, bc)
     c. *Trigram*: mengekstrak 3 karakter berurutan (abc)
   - *Word n-gram* (n=1,2)
     a. *Unigram*: mengekstrak kata tunggal
     b. *Bigram*: mengekstrak 2 kata berurutan
3. Perhitungan TF-IDF
   - *Term Frequency* (TF): hitung frekuensi kemunculan setiap fitur
   - *Document Frequency* (DF): hitung jumlah dokumen yang memuat fitur
   - IDF = log(N/DF), N = jumlah total dokumen
   - TF-IDF = TF × IDF
4. Pemodelan *Support Vector Machine*, Vektor TF-IDF dari *training data* digunakan sebagai *input* untuk melatih model SVM. Cara kerja SVM pada tahap ini adalah:

   - SVM mencari *hyperplane* (bidang pemisah) terbaik yang dapat memisahkan data dari dua kelas (*Api* dan *Nyo*) dengan *margin* (jarak) maksimum antara *hyperplane* dan titik-titik data terdekat dari kedua kelas.
   - Jika data tidak dapat dipisahkan secara *linear* di ruang asli, SVM menggunakan *kernel trick* (misalnya, kernel RBF) untuk memetakan data ke dimensi yang lebih tinggi, di mana pemisahan *linear* menjadi mungkin.

5. Ujicoba atau Prediksi Klasifikasi Teks Baru,
   Untuk mengidentifikasi dialek dari teks baru, langkah-langkah berikut

dilakukan:

- Ekstrak fitur (*character n-gram* dan *word n-gram*) dari teks baru menggunakan metode yang sama seperti pada *data training*.
- Hitung vektor TF-IDF berdasarkan fitur-fitur tersebut dengan memanfaatkan informasi dari data latih (misalnya, DF yang sudah dihitung sebelumnya).
- Masukkan vektor TF-IDF ke dalam model SVM yang telah dilatih.
- Model SVM akan menentukan kelas dialek (*Api* atau *Nyo*) berdasarkan posisi teks relatif terhadap *hyperplane* yang telah terbentuk dari model.

6. Evaluasi dan Analisis Hasil
   - Identifikasi kekuatan dan kelemahan model
   - Analisis hasil klasifikasi

SVM berpotensi cocok untuk *dialect identification* karena:

- Kemampuannya menangani data berdimensi tinggi, seperti vektor TF-IDF yang dihasilkan dari ekstraksi fitur teks.
- Kapasitasnya untuk menemukan batas keputusan yang optimal antara dua kelas, bahkan ketika data kompleks atau tidak sepenuhnya terpisah secara *linear*.

Dengan langkah-langkah ini, SVM dapat berpotensi secara akurat mengklasifikasikan teks ke dalam dialek *Api* atau *Nyo* berdasarkan pola linguistik yang ditangkap dari *data training*.

Langkah-langkah dalam membangun model *Lampung Language Dialect Identification* dengan metode *Random Forest* (Géron, 2022; Kedia & Rasu, 2020) :

1. Sediakan *Data set* berupa 3000 kalimat bahasa Lampung dialek *Api* dan 9076 kalimat dialek *Nyo*
2. Lakukan Ekstraksi Fitur dari *data training*
   - *Character n-gram* (n=1,2,3)

a. *Unigram*: mengekstrak karakter tunggal (a, b, c)

b. *Bigram*: mengekstrak 2 karakter berurutan (ab, bc)

c. *Trigram*: mengekstrak 3 karakter berurutan (abc)

- *Word n-gram* (n=1,2)

a. *Unigram*: mengekstrak kata tunggal

b. *Bigram*: mengekstrak 2 kata berurutan

3. Perhitungan TF-IDF

- *Term Frequency* (TF): hitung frekuensi kemunculan setiap fitur
- *Document Frequency* (DF): hitung jumlah dokumen yang memuat fitur
- IDF = log(N/DF), N = jumlah total dokumen
- TF-IDF = TF × IDF

4. Pelatihan Model *Random Forest*

Random Forest adalah metode *ensemble learning* yang terdiri dari banyak pohon keputusan (*decision trees*). Cara kerjanya pada tahap pelatihan adalah:

- Dataset dibagi menjadi beberapa subset acak menggunakan metode *bagging* (*bootstrap aggregating*).
- Setiap pohon keputusan dilatih pada *subset* data yang berbeda dan hanya menggunakan *subset* acak dari fitur (*random feature selection*).
- Pada setiap pohon, data dipisahkan berdasarkan fitur yang paling informatif (misalnya, fitur dengan nilai TF-IDF yang paling membedakan *Api* dan *Nyo*).
- Proses ini diulang hingga semua pohon selesai dilatih, menghasilkan kumpulan pohon yang beragam.

5. Ujicoba atau Prediksi Klasifikasi Teks Baru,

Untuk mengklasifikasikan teks baru ke dalam dialek *Api* atau *Nyo*, Langkah-Langkahnya adalah:

- Ekstrak fitur dari teks baru (*character n-gram* dan *word n-gram*).
- Hitung vektor TF-IDF berdasarkan fitur yang diekstrak.

- Masukkan vektor TF-IDF ke dalam setiap pohon keputusan di *Random Forest*.
- Setiap pohon memberikan prediksi kelas (*Api* atau *Nyo*).
- Tentukan kelas akhir dengan *majority voting*: kelas yang dipilih oleh mayoritas pohon adalah hasil prediksi akhir.

6. Evaluasi dan Analisis Hasil
   - Identifikasi kekuatan dan kelemahan model
   - Analisis hasil klasifikasi

Random Forest berpotensi cocok untuk *dialect identification* karena:

- *Ensemble Learning*: Dengan menggabungkan prediksi dari banyak pohon, *Random Forest* mengurangi risiko *overfitting* dan meningkatkan akurasi dibandingkan satu pohon keputusan saja.
- Kemampuan menangani data berdimensi tinggi: Vektor TF-IDF sering kali memiliki dimensi besar (banyak fitur), dan *Random Forest* dapat menanganinya dengan baik.
- Identifikasi Fitur Penting: *Random Forest* dapat menunjukkan fitur mana (misalnya, karakter atau kata tertentu) yang paling berperan dalam membedakan dialek *Api* dan *Nyo*.

Dengan metode ini, *Random Forest* dapat secara efektif mengklasifikasikan teks ke dalam dialek *Api* dan *Nyo* berdasarkan pola linguistik yang dipelajari dari *data training*.

3.4 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan pada penelitian *word stemming* (WS) dan *text data augmentation* (TDA) sebagai berikut:

1. Tinjauan pustaka

   Dalam penelitian WS dan TDA bahasa Lampung, metode penelusuran terkait pustaka yang digunakan yaitu meliputi berbagai jurnal, prosiding, *textbook* berkaitan dengan WS, TDA dan buku kajian bahasa Lampung.
2. Observasi pengumpulan data teks bahasa Lampung

Dalam penelitian WS dan TDA bahasa Lampung, dilakukan observasi langsung dan diketik secara manual bersumber pada buku bahasa Lampung tingkat SD dan SMP serta kamus bahasa Lampung edisi kedua yang diterbitkan oleh kantor bahasa provinsi Lampung.

Bahasa Lampung dialek Tulang Bawang merupakan bahasa daerah yang minim sumber data digital. Minimnya data digital ini menyulitkan pencarian kata berimbuhan dalam bahasa Lampung dialek Tulang Bawang. Data yang digunakan dalam penelitian ini adalah kumpulan kata dasar dari Kamus Bahasa Lampung-Indonesia edisi kedua yang dibuat oleh Balai Bahasa Provinsi Lampung, 500 kata uji, dan 200 kata uji mandiri yang diambil dari buku "Sistem Morfologi Verba Bahasa Lampung Dialek Tulang Bawang". Distribusi afiks pada 500 kata disajikan pada Tabel 3.3 sedang pada 200 kata disajikan pada Tabel 3.4.

Tabel 3.3 Distribusi Afiks pada 500 Kata Uji Dialek Tulang Bawang

Distribusi Afiks	Jumlah
Infix	9
Konfiks	179
Prefiks	174
Reduplikasi	37
Sufiks	101
Total	500

Tabel 3.4 Distribusi Afiks pada 200 Kata Uji Dialek Tulang Bawang

Distribusi Afiks	Jumlah
Infix	1
Konfiks	36
Prefiks	90
Reduplikasi	18
Sufiks	55
Total	200

3.5 Kebutuhan *Software* dan *Hardware*

Kebutuhan *software* dan *hardware* yang digunakan untuk membangun model *word stemming* dan *text data augmentation* sebagai berikut :

1. *Software*

   Software yang digunakan Python, Google Colab, Microsoft Word.

2. *Hardaware*

   Hardaware utama yang digunakan adalah satu buah laptop Core i7 dengan RAM 8 Giga dan harddisk 250 Giga.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Eksperimen yang telah dilakukan pada *word stemming* (WS) dialek Tulang Bawang (TB) serta eksperimen *text data augmentation* (TDA) kalimat dialek *Api* menghasilkan beberapa kesimpulan sebagai berikut:

1. *Word stemming* dialek Tulang Bawang pada prinsip dasarnya adalah mengikuti seperti yang dilakukan WS pada bahasa Indonesia tetapi dengan melakukan penyesuaian sesuai kebutuhan pada masing-masing metode. (a) metode modifikasi Nazief-Adriani dilakukan dengan cara *stemming* dengan urutan sufiks, prefiks dan mekanisme *recording*, (b) metode modifikasi *Confix-Stripping* dilakukan dengan cara *stemming* dengan urutan reduplikasi, sufiks, prefiks disertai mekanisme *recording*, serta infiks, (c) metode modifikasi *Confix-Stripping* disertai *N-Gram Stemming* mengikuti urutan proses yang sama persis dengan metode modifikasi *Confix-Stripping*, dan jika masih gagal maka lakukan *N-Gram stemming* khusus pada kata yang gagal, (d) metode *Morhological-based* dibangun dengan mengerahkan semua afiksasi yang ada yaitu dengan cara *stemming* dengan urutan reduplikasi, konfiks, sufiks, prefiks disertai mekanisme *recording*, serta infiks, sedangkan (e) metode yang terakhir yaitu *N-Gram stemming* dilakukan tanpa melibatkan unsur morfologi melainkan mencari tingkat *similarity* atau *dice coefficient bi-gram* dan *tri-gram* pada kata uji dan kata acuan yang ada di kamus dialek Tulang Bawang dengan nilai *threshold* 0,5. Hasil eksperimen menunjukkan bahwa modifikasi *Confix-Stripping* disertai *N-Gram stemming* adalah metode *stemming* terbaik untuk dialek Tulang Bawang berdasarkan nilai *Gold Standar Asessment* (GSA) sebesar 98,8 %.
2. Eksperimen yang dilakukan untuk mendapatkan metode WS terbaik yaitu dengan melakukan eksperimen *stemming* pada 500 kata data uji dan 200 kata uji independen pada lima modifikasi metode yaitu MNA, MCS, MCS disertai

N-Gram stemming, *Morhological-based* serta *N-Gram stemming*. MCS disertai *N-Gram stemming* sebagai metode terbaik dan Langkah selanjutnya metode terbaik ditanamkan pada aplikasi *Direct Machine Translation* (DMT) Tulang Bawang – Indonesia dan mendapatkan nilai *Bilingual Evaluation Understudy* (BLEU) *Score* sebesar 80,07 %.

3. Membangun model TDA metode permutasi dilakukan dengan cara memilih satu kalimat yang berisi *n* token. Lalu dilakukan permutasi pada kalimat tersebut dan menghasilkan sebanyak *n*! kalimat yang digunakan sebagai *data set* baru hasil sintesis secara komputasi. Kata-kata yang tersusun dengan bantuan program Python sehingga dibangkitkan data kalimat baru dan dipergunakan untuk *data set* tambahan pada *task Lampung Language Dialect Identification*.

4. Membangun model *Lampung Language Dialect Identification* (LLDI) pada bahasa Lampung dengan kondisi komparatif melibatkan Langkah-Langkah sebagai berikut: pertama, melakukan *text preprocessing* pada 3000 kalimat dialek *Api* dan 9078 kalimat dialek *Nyo* melalui tokenisasi, normalisasi, dan ekstraksi fitur (seperti *n-gram* karakter/kata); kemudian, *data set* dibagi menggunakan *5-fold cross validation* untuk memastikan evaluasi yang konsisten dan menghindari *overfitting*; selanjutnya, empat metode klasifikasi dibandingkan – *Naive Bayes* (efisien dengan asumsi independensi fitur), *Logistic Regression* (model probabilistik dengan interpretabilitas tinggi), *Support Vector Machine* (efektif pada data dimensi tinggi dengan pemisahan optimal), dan *Random Forest* (*ensemble* yang tangguh terhadap *noise* dan *overfitting*); terakhir, performa model-model tersebut dievaluasi menggunakan metrik seperti akurasi, presisi, *recall*, dan *F1-score*, dengan perhatian khusus pada ketidakseimbangan *data set* (rasio 1:3 antara dialek *Api* dan *Nyo*) yang memerlukan penambahan *data set* melalui TDA metode permutasi untuk mengoptimalkan hasil klasifikasi.
5. Eksperimen TDA digunakan untuk menambahkan *data set* pada *training data task dialect identification*. Pada *dialect identification* ini dievaluasi melalui performa empat model klasifikasi—*Random Forest, Logistic Regression,* SVM, dan *Naive Bayes*—dalam *task* identifikasi dialek *Api* dan *Nyo* pada bahasa

Lampung, dengan kondisi kelas seimbang dan tidak seimbang, menggunakan metrik seperti *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), *True Negative* (TN), *Total Instance, Accuracy, Precision* (*Api*), *Recall* (*Api*), *F1-Score* (*Api*), dan *Specificity* (*Nyo*). Model SVM *Confusion Balanced Class* mencapai performa tertinggi dengan akurasi 97,4%, *precision* 97,4%, *recall* 97,7%, *F1-Score* 97,5%, dan *specificity* 97,4%, menunjukkan kemampuan optimal dalam mengklasifikasikan kedua dialek. *Random Forest Balanced Class* juga tampil kuat dengan akurasi 96,9% dan *F1-Score* 96,8%, diikuti oleh *Logistic Regression Balanced Class* dengan akurasi 96,3% dan *F1-Score* 97,0%. Sebaliknya, *Naive Bayes Unbalanced Class* memiliki performa terendah dengan akurasi 85,9%, *recall* 42,5%, dan *F1-Score* 59,5%, meskipun *precision* 99,3%, menandakan banyak *instance* dialek *Api* yang terlewat karena dominasi dialek *Nyo*. Pada kondisi tidak seimbang, semua model cenderung memiliki *precision* tinggi namun *recall* rendah untuk dialek *Api*, sementara penyeimbangan kelas terbukti meningkatkan deteksi dialek minoritas (*Api*) tanpa mengorbankan performa pada dialek mayoritas (*Nyo*), dengan SVM dan *Random Forest* dengan kondisi *balanced* sebagai pilihan terbaik.

5.2 Saran

Pada penelitian *stemming* kata pada bahasa Lampung dialek Tulang Bawang, dua hal utama yang diperhatikan dengan seksama adalah adanya detail aturan morfologi dan kamus kata dasar dalam jumlah yang tidak sedikit karena keduanya merupakan komponen utama pada *stemming* kata. Eksprimen selanjutnya yang berpotensi dilakukan adalah mengkombinasikan antara metode *Morphological-based* dengan *N-Gram stemming*. Alternatif evaluasi hasil *stemming* yaitu menggunakan metode *Damerau Levenshtein Distance* (DLD).

Pada eksperimen *Text Data Augmentation*, salah satu metode yang dapat digunakan yaitu *template-based slot filling* dengan menyiapkan naskah korpus agar dapat dibuat dalam kondisi sudah melalui tokenisasi dan berlabel subjek, predikat, objek, keterangan. Kelemahan metode permutasi adalah kalimat menjadi tidak tersusun sesuai kaidah sintaksis dan metode *template-based slot filling* berpotensi mempertahankan dari aspek sintaksis.

DAFTAR PUSTAKA

Abidin, Z., Junaidi, A., & Wamiliana. (2024). Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review. *Journal of Information Systems Engineering and Business Intelligence*, *10*(2), 217–231. https://doi.org/10.20473/jisebi.10.2.217-231

Abidin, Z., Junaidi, A., Wamiliana, Togatorop, F. M., Ahmad, I., & Puspaningrum, A. S. (2023). Direct Machine Translation Indonesian-Batak Toba. *Proceedings of the 7th 2023 International Conference on New Media Studies, CONMEDIA 2023*, 82–87. https://doi.org/10.1109/CONMEDIA60526.2023.10428332

Abidin, Z., Wijaya, A., & Pasha, D. (2021a). Aplikasi Stemming Kata Bahasa Lampung Dialek Api Menggunakan Metode Brute-Force dan Pemograman C#. *Jurnal Media Informatika Budidarma*, *5*(1), 1. https://doi.org/10.30865/mib.v5i1.2483

Abidin, Z., Permata, P., & Ariyani, F. (2021b). Translation of the Lampung Language Text Dialect of Nyo into the Indonesian Language with DMT and SMT Approach. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, *5*(1), 58–71. https://doi.org/10.29407/intensif.v5i1.14670

Abdurrahman & Purwarianti, A. (2019, October). Effective use of augmentation degree and language model for synonym-based text augmentation on Indonesian text classification. In *2019 International Conference on Advanced Computer Science and information Systems (ICACSIS)* (pp. 217-222). IEEE.

Agus, M., Subali, P., & Fatichah, C. (2019). *Kombinasi Metode Rule-based dan N-Gram Stemming untuk Mengenali Stemmer Bahasa Bali*. *6*(2), 219–228. https://doi.org/10.25126/jtiik.201961105

Aini, L. R., Nurfadhilah, E., Jarin, A., Santosa, A., & Uliniansyah, M. T. (2023). Enhancing Sentiment Analysis Models through Multi-Technique Data Augmentation: A Study with IndoBERT. *Proceedings - 2023 10th International Conference on Computer, Control, Informatics and Its Applications: Exploring the Power of Data: Leveraging Information to Drive Digital Innovation, IC3INA 2023*, 137–142. https://doi.org/10.1109/IC3INA60834.2023.10285775

Alyousf, M., & Alhalabi, M. F. (2025). A Survey of Document Stemming Algorithms in Information Retrieval Systems. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *24*(4), 1-28.

Amin, F., Hadikurniawati, W., Wibisono, S., Februariyanti, H., & Wibowo, J. S. (2017). A Hybrid Method of Rule-based and String Matching Stemmer for Javanese Language. *Journal of Theoretical and Applied Information Technology*, *15*, 19. www.jatit.org

Amin, F., Razaq, J. A. (2018). Implementasi Stemmer Bahasa Jawa dengan Metode Rule Base Approach pada Sistem Temu Kembali Informasi Dokumen Teks Berbahasa Jawa. Prosiding SENDI 2018.

Amisani, D. (1991). *Sistem morfologi nomina dan adjektiva bahasa Lampung dialek pesisir*. Pusat Pembinaan dan Pengembangan Bahasa, Departemen Pendidikan dan Kebudayaan.

Andriani, Y. F., Utami, E., & Suwanto, S. (2019). Modifikasi Metode Porter Stemmer Untuk Stemming Bahasa Sasak. *Jurnal Informa: Jurnal Penelitian dan Pengabdian Masyarakat*, *5*(3), 61-64.

Ardiyanti Suryani, A., Hendratmo Widyantoro, D., Purwarianti, A., & Sudaryat, Y. (2018). The rule-based sundanese stemmer. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *17*(4). https://doi.org/10.1145/3195634

Arif Siswandi, A., Permana, Y., & Emarilis, A. (2021). Stemming Analysis Indonesian Language News Text with Porter Algorithm. *Journal of Physics: Conference Series*, *1845*(1). https://doi.org/10.1088/1742-6596/1845/1/012019

Arifin, A. Z., Mahendra, I. P. A. K., & Ciptaningtyas, H. T. (2009, April). Enhanced confix stripping stemmer and ants algorithm for classifying news document in indonesian language. In *The International Conference on Information & Communication Technology and Systems* (Vol. 5, pp. 149-158).

Arimbawa, I. G. A. P., & Era, N. A. S. (2017). Lemmatization in Balinese language. *Jurnal Elektronik Ilmu Komputer Udayana p-ISSN*, *2301*, 5373.

Ariyani, F. (2014). Distribusi Verba Berprefiks {N-} pada Bahasa Lampung dalam Kitab Kuntara Raja Niti dan Buku Ajar: Kajian Morfologi. *Ranah: Jurnal Kajian Bahasa*, *3*(2), 124-134.

Ariyani, F., Putrawan, G. E., Riyanda, A. R., Idris, A. R., Misliani, L., & Perdana, R. (2022). Technology and minority language: an Android-based dictionary development for the Lampung language maintenance in Indonesia. *Tapuya: Latin American Science, Technology and Society*, *5*(1). https://doi.org/10.1080/25729861.2021.2015088

Ariyani, F., Rusminto, N. E., Sumarti, Idris, A. R., & Misliani, L. (2022). Examining the Forms and Variations of the Lampung Script in Ancient Manuscripts. *WSEAS Transactions on Environment and Development*, *18*, 204–217. https://doi.org/10.37394/232015.2022.18.22

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M., & Williams, H. E. (2007). Stemming Indonesian: A confix-stripping approach. *ACM Transactions on Asian Language Information Processing (TALIP)*, *6*(4), 1-33.

Asian, J., Williams, H. E., & Tahaghoghi, S. M. M. (2005). Stemming Indonesian. *Conferences in Research and Practice in Information Technology Series*, *38*, 307–314. https://doi.org/10.1145/1316457.1316459

Trianto, R. B., Nugroho, A. S., & Supriyadi, E. (2023). Klasterisasi Menggunakan Metode K-Means Dan Elbow Pada Opini Masyarakat Tentang Kebijakan Sekolah Luring Tahun 2022. *Jurnal Inovtek Polbeng Seri Informatika*, *8*(1), 1-13.

Bayer, M., Kaufhold, M. A., Buchhold, B., Keller, M., Dallmeyer, J., & Reuter, C. (2023). Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International Journal of Machine Learning and Cybernetics*, *14*(1), 135–150. https://doi.org/10.1007/s13042-022-01553-3

Bayer, M., Kaufhold, M. A., & Reuter, C. (2022b). A Survey on Data Augmentation for Text Classification. *ACM Computing Surveys*, *55*(7). https://doi.org/10.1145/3544558

Bencke, L., & Moreira, V. P. (2024). Data augmentation strategies to improve text classification: a use case in smart cities. *Language Resources and Evaluation*, *58*(2), 659–694. https://doi.org/10.1007/s10579-023-09685-w

Bunyamin, Huda, A. F., & Suryani, A. A. (2021). Indonesian Stemmer for Ambiguous Word based on Context. *2021 International Conference on Data Science and Its Applications, ICoDSA 2021*, 91–96. https://doi.org/10.1109/ICoDSA53588.2021.9617514

Cahyani, D. E., Utami, L. M. T., & Setiadi, H. (2019). Clustering of Javanese News in Krama Alus Level with Javanese Stemming. In *2019 International Conference on Information and Communications Technology (ICOIACT)* (pp. 462-467). IEEE.

Chen, J., Tam, D., Raffel, C., Bansal, M., & Yang, D. (2023). An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, *11*, 191-211.

Hidayatullah, N., Wibawa, A. P., & Rosyid, H. A. (2017). Penerapan ECS Stemmer untuk Modifikasi Nazief & Adriani Berbahasa Jawa. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)* Vol **.** 3 No. 3 (2019) 343 - 348.

Dwiharyono, H., & Suyanto, S. (2022). Stemming for Better Indonesian Text-to-Phoneme. *Ampersand*, *9*. https://doi.org/10.1016/j.amper.2022.100083

Eisenstein, J. (2019). *Introduction to natural language processing*. MIT press.

Enni Lindrawati, Ema Utami, & Yaqin, A. (2023). ANoM STEMMER: Nazief & Andriani Modification for Madurese Stemming. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(6), 1341–1347. https://doi.org/10.29207/resti.v7i6.5086

Fadilah, N., & Priyanta, S. (2022). Automatic Essay Scoring Using Data Augmentation in Bahasa Indonesia. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *16*(4), 401. https://doi.org/10.22146/ijccs.76396

Fahmi, S., Purnamawati, L., Shidik, G. F., Muljono, M., & Fanani, A. Z. (2020).

Sentiment analysis of student review in learning management system based on sastrawi stemmer and SVM-PSO. *Proceedings - 2020 International Seminar on Application for Technology of Information and Communication: IT Challenges for Sustainability, Scalability, and Security in the Age of Digital Disruption, ISemantic 2020*, 643–648. https://doi.org/10.1109/iSemantic50169.2020.9234291

Faidha, Y. F., Shidik, G. F., & Fanani, A. Z. (2021). Study Comparison Stemmer to Optimize Text Preprocessing in Sentiment Analysis Indonesian E-Commerce Reviews. *2021 International Conference on Data Analytics for Business and Industry, ICDABI 2021*, 135–139. https://doi.org/10.1109/ICDABI53623.2021.9655867

Fathan Hidayatullah, A., Ratnasari, C. I., & Wisnugroho, S. (2016). Analysis of Stemming Influence on Indonesian Tweet Classification. *Telkomnika*, *14*(2), 1693–6930. https://doi.org/10.12928/TELKOMNIKA.v14i1.3113

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). *A Survey of Data Augmentation Approaches for NLP*. *arXiv preprint arXiv:2105.03075*.

Gede Surya Cipta Nugraha, P., & Wayan Wardani, N. (2020). *Stemming Dokumen Teks Bahasa Bali Dengan Metode Rule Base Approach* (Vol. 7, Issue 3). http://jurnal.mdp.ac.idjatisi@mdp.ac.idceivedjune1ssedju

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.".

Guterres, A., Gunawan, & Santoso, J. (2019). Stemming Bahasa Tetun Menggunakan Metode Rule Based. *Teknika*, *8*(2), 142–147. https://doi.org/10.34148/teknika.v8i2.224

Hadiwinoto, P. N., & Lestari, D. P. (2020). Data augmentation on spontaneous Indonesian automatic speech recognition using statistical machine translation. *IOP Conference Series: Materials Science and Engineering*, *803*(1). https://doi.org/10.1088/1757-899X/803/1/012030

Haralabopoulos, G., Torres, M. T., Anagnostopoulos, I., & McAuley, D. (2021). Text data augmentations: Permutation, antonyms and negation. *Expert Systems with Applications*, *177*. https://doi.org/10.1016/j.eswa.2021.114769

Hedderich, M. A., Lange, L., Adel, H., Strötgen, J., & Klakow, D. (2021, June). A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 2545-2568).

Hermawan, W., Eko R, N., Udin, N., Akhyar, W., & Sanusi, E. (2001). *Sistyem morfologi verba bahasa Lampung dialek Tulang Bawang*. Pusat Bahasa.

Hrp, N. H., Fikry, M., & Yusra, Y. (2023). Metode Stemming Teks Bahasa Batak Angkola Berbasis Aturan Tata Bahasa. *Journal of Computer System and Informatics (JoSYC)*, *4*(3), 642–648.

https://doi.org/10.47065/josyc.v4i3.3458

Imin, G., Ablimit, M., Yilahun, H., & Hamdulla, A. (2022). A Character String-Based Stemming for Morphologically Derivative Languages. *Information (Switzerland)*, *13*(4). https://doi.org/10.3390/info13040170

Indra Winata, G., Fikri Aji, A., Cahyawijaya, S., Mahendra, R., Koto, F., Romadhony, A., Kurniawan, K., Moeljadi, D., Eko Prasojo, R., Fung, P., Baldwin, T., Han Lau, J., Sennrich, R., & Ruder, S. (n.d.). *NusaX: Multilingual Parallel Sentiment Dataset for 10 Indonesian Local Languages*. https://github.com/

Indrahimawan, M. R., Santosa, P. I., & Adji, T. B. (2023). Handling Data Imbalance Using Text Augmentation for Classifying Public Complaints. *Proceedings - 2023 10th International Conference on Computer, Control, Informatics and Its Applications: Exploring the Power of Data: Leveraging Information to Drive Digital Innovation, IC3INA 2023*, 284–289. https://doi.org/10.1109/IC3INA60834.2023.10285813

Jabbar, A., Illahi, M., Iqbal, S., Khan, A. R., Elhakim, N., & Saba, T. (2023). PWMStem: A Corpus-Based Suffix Identification and Stripping Algorithm for Multi-lingual Stemming. *Journal of Advances in Information Technology*, *14*(4), 863–875. https://doi.org/10.12720/jait.14.4.863-875

Jabbar, A., Iqbal, S., Alaulamie, A. A., & Ilahi, M. (2024). Building a Multilevel Inflection Handling Stemmer to Improve Search Effectiveness for Urdu Language. *IEEE Access*, *12*, 39313–39329. https://doi.org/10.1109/ACCESS.2024.3373714

Jabbar, A., Iqbal, S., Tamimy, M. I., Hussain, S., & Akhunzada, A. (2020). Empirical evaluation and study of text stemming algorithms. *Artificial Intelligence Review*, *53*(8), 5559–5588. https://doi.org/10.1007/s10462-020-09828-3

Jabbar, A., Iqbal, S., Tamimy, M. I., Rehman, A., Bahaj, S. A., & Saba, T. (2023). An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems. *IEEE Access*, *11*, 133681–133702. https://doi.org/10.1109/ACCESS.2023.3332710

Jauhari, A., Suzanti, I. O., Pramudita, Y. D., Husni, & Diantisari, N. P. W. (2020). Enhanced confix stripping stemmer and cosine similarity for search engine in the holy qur"an translation. *Proceeding - 6th Information Technology International Seminar, ITIS 2020*, 207–212. https://doi.org/10.1109/ITIS50118.2020.9321041

Junaidi, A. (2016). *Lampung handwritten character recognition* (Doctoral dissertation, Dissertation, Dortmund, Technische Universität, 2016).

Junaidi, A., Grzeszick, R., Fink, G. A., & Vajda, S. (2013). Statistical modeling of the relation between characters and diacritics in lampung script. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 663–667. https://doi.org/10.1109/ICDAR.2013.136

Junaidi, A., Vajda, S., & Fink, G. A. (2011). Lampung - A new handwritten character benchmark: Database, labeling and recognition. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/2034617.2034632

Jurafsky, D. & Martin, J. H. 2026. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 3rd edition. Online manuscript released January 6, 2026. https://web.stanford.edu/~jurafsky/slp3.

Tamrizal, A.M. (2022). *Metode Stemming untuk Bahasa Kaili*. (Tesis). Universitas Amikom Yogyakarta. 138 p.

Kartika, B. V., Alfredo, M. J., & Kusuma, G. P. (2023). Fine-Tuned IndoBERT based model and data augmentation for indonesian language paraphrase identification. *Revue d''Intelligence Artificielle*, *37*(3), 733–743. https://doi.org/10.18280/ria.370322

Kedia, A., & Rasu, M. (2020). *Hands-On Python Natural Language Processing: Explore tools and techniques to analyze and process text with a view to building real-world NLP applications*. Packt Publishing Ltd.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

Kim, H. S., Kang, Y., & Lee, J. H. (2024). STAGE: Simple Text Data Augmentation by Graph Exploration. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 15238-15256).

Kim, H. S., & Lee, J. H. (2024). Need Text Data Augmentation? Just One Insertion Is Enough. *International Journal of Fuzzy Logic and Intelligent Systems*, *24*(2), 83–92. https://doi.org/10.5391/IJFIS.2024.24.2.83

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering - A systematic literature review. In *Information and Software Technology* (Vol. 51, Issue 1, pp. 7–15). https://doi.org/10.1016/j.infsof.2008.09.009

Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., & Linkman, S. (2010). Systematic literature reviews in software engineering-A tertiary study. In *Information and Software Technology* (Vol. 52, Issue 8, pp. 792–805). Elsevier B.V. https://doi.org/10.1016/j.infsof.2010.03.006

Kusuma Wardana, H., Swanita, I., & Yohanes, B. W. (2019). Sistem Pemeriksa Pola Kalimat Bahasa Indonesia berbasis Metode Left-Corner Parsing dengan Stemming. In *JNTETI* (Vol. 8, Issue 3).

Li, B., Hou, Y., & Che, W. (2022). Data augmentation approaches in natural language processing: A survey. *AI Open*, *3*, 71–90. https://doi.org/10.1016/j.aiopen.2022.03.001

Lindrawati, E., Utami, E., & Yaqin, A. (2023). Comparison of Modified Nazief&Adriani and Modified Enhanced Confix Stripping algorithms for Madurese Language Stemming. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, *7*(2), 276–289. https://doi.org/10.29407/intensif.v7i2.20103

Maesya, A., Arifin, Y., Zahra, A., & Budiharto, W. (2023). Development of Sundanese Stemmer Based on Morphophonemics. *10th International Conference on ICT for Smart Society, ICISS 2023 - Proceeding*. https://doi.org/10.1109/ICISS59129.2023.10291840

Maesya, A., Ramadhan, A., Abdurachman, E., Trisetyarso, A., & Zarlis, M. (2022). Stemming Algorithm for the Indonesian Language: A Scientometric View. *2022 IEEE Creative Communication and Innovative Technology, ICCIT 2022*. https://doi.org/10.1109/ICCIT55355.2022.10119050

Magueresse, A., Carles, V., & Heetderks, E. (2020). *Low-resource Languages: A Review of Past Work and Future Challenges*. http://arxiv.org/abs/2006.07264

Maulidi, R. (2016). Stemmer Untuk Bahasa Madura Dengan Modifikasi Metode Enhanced Confix Stripping Stemmer. In *Prosiding Seminar Nasional FDI* (Vol. 2016, pp. 12-15).

Maulana, F. I., Heryadi, Y., Kusuma, G. P., & Budiharto, W. (2025). Data augmentation English-Indonesia-Madurese parallel corpus dataset using neural machine translation. *Data in Brief*, 112046.

Melia, S. I., Sholihah, J., Nisak, D., Juniaristha, I. S., & Ni"mah, A. T. (2023). The Ngoko Javanese Stemmer uses the Enhanced Confix Stripping Stemmer Method. *Rekayasa*, *16*(1), 107–112. https://doi.org/10.21107/rekayasa.v16i1.19308

Muchtar, M. A., Jaya, I., Nababan, M., Andayani, U., Siregar, L. N., Nababan, E. B., & Sitompul, O. S. (2019). Separation of Basic Words in Angkola Batak Text Documents using Enhanced Confix Stripping Stemmer Case: Mandailing Ethnic. *IOP Conference Series: Materials Science and Engineering*, *648*(1). https://doi.org/10.1088/1757-899X/648/1/012024

Muftie, F., & Haris, M. (2023). IndoBERT Based Data Augmentation for Indonesian Text Classification. *2023 International Conference on Information Technology Research and Innovation, ICITRI 2023*, 128–132. https://doi.org/10.1109/ICITRI59340.2023.10250061

Mulyana, I., Suhendra, A., Ernastuti, & Bheta Agus, W. (2019). Development of indonesian stemming algorithms through modification of grouping, sequencing and removing of affixes based on morphophonemic. *International Journal of Recent Technology and Engineering*, *8*(2 Special Issue 7), 179–184. https://doi.org/10.35940/ijrte.B1044.0782S719

Natasya, & Girsang, A. S. (2023). Modified EDA and Backtranslation Augmentation in Deep Learning Models for Indonesian Aspect-Based Sentiment Analysis. *Emerging Science Journal*, *7*(1), 256–272.

https://doi.org/10.28991/ESJ-2023-07-01-018

Nicolas, J., Gunawan, V., & Sutanto, A. (2025). Evaluation of the effect of back-translation direction on machine translation quality for low-resource sundanese-english language. *Procedia Computer Science*, *269*, 784-796.

Nigatu, H. H., Tonja, A. L., Rosman, B., Solorio, T., & Choudhury, M. (2024). The Zeno"s Paradox ofLow-Resource"Languages. *arXiv preprint arXiv:2410.20817*.

Noor, A. Z. M., Gernowo, R., & Nurhayati, O. D. (2023). Data Augmentation for Hoax Detection through the Method of Convolutional Neural Network in Indonesian News. *Jurnal Penelitian Pendidikan IPA*, *9*(7), 5078–5084. https://doi.org/10.29303/jppipa.v9i7.4214

Nq, M. A., Manik, L. P., & Widiyatmoko, D. (2020). Stemming Javanese: Another Adaptation of the Nazief-Adriani Algorithm. *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2020*, 627–631. https://doi.org/10.1109/ISRITI51436.2020.9315420

Pakray, P., Gelbukh, A., & Bandyopadhyay, S. (2025). Natural language processing applications for low-resource languages. *Natural Language Processing*, *31*(2), 183-197.

Pande, B. P., Tamta, P., & Dhami, H. S. (2019). Generation, implementation, and appraisal of an N-gram-based stemming algorithm. *Digital Scholarship in the Humanities*, *34*(3), 558–568. https://doi.org/10.1093/llc/fqy053

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

Paskahningrum, Y. K., Utami, E., & Yaqin, A. (2023). A Systematic Literature Review of Stemming in Non-Formal Indonesian Language. In *International Journal of Innovative Science and Research Technology* (Vol. 8, Issue 1). www.ijisrt.com

Pellicer, L. F. A. O., Ferreira, T. M., & Costa, A. H. R. (2023). Data augmentation techniques in natural language processing. *Applied Soft Computing*, *132*. https://doi.org/10.1016/j.asoc.2022.109803

Permata, P., & Abidin, Z. (2020). Statistical Machine Translation Pada Bahasa Lampung Dialek Api Ke Bahasa Indonesia. *Jurnal Media Informatika Budidarma*, *4*(3), 519. https://doi.org/10.30865/mib.v4i3.2116

Pramana, R., Debora, Subroto, J. J., Gunawan, A. A. S., & Anderies. (2022). Systematic Literature Review of Stemming and Lemmatization Performance for Sentence Similarity. *Proceedings of the 2022 IEEE 7th International Conference on Information Technology and Digital Applications, ICITDA 2022*. https://doi.org/10.1109/ICITDA55840.2022.9971451

Putra, O. V., Wasmanson, F. M., Harmini, T., & Utama, S. N. (2020). Sundanese

Twitter Dataset for Emotion Classification. *CENIM 2020 - Proceeding: International Conference on Computer Engineering, Network, and Intelligent Multimedia 2020*, 391–395. https://doi.org/10.1109/CENIM51130.2020.9297929

Putu, I., Wirayasa, M., Made, I., Wirawan, A., Pradnyana, A., Kunci, K., Stemming, :, Bali, B., & Bastal, A. (2019). Metode Bastal: Adaptasi Metode Nazief & Adriani Untuk Stemming Teks Bahasa Bali. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*. (Vol. 8).

Rachman, F. H., Ifada, N., Wahyuni, S., Ramadani, G. D., & Pawitra, A. (2022). ModifiedECS (mECS) Algorithm for Madurese-Indonesian Rule-Based Machine Translation. *2022 International Conference of Science and Information Technology in Smart Administration, ICSINTESA 2022*, 51–56. https://doi.org/10.1109/ICSINTESA56431.2022.10041470

Rifai, W., & Winarko, E. (2019). Modification of Stemming Algorithm Using A Non Deterministic Approach To Indonesian Text. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *13*(4), 379. https://doi.org/10.22146/ijccs.49072

Rizki, A. S., Tjahyanto, A., & Trialih, R. (2019). Comparison of stemming algorithms on Indonesian text processing. *Telkomnika (Telecommunication Computing Electronics and Control)*, *17*(1), 95–102. https://doi.org/10.12928/TELKOMNIKA.v17i1.10183

Rusminto, N. E. (2000). *Kata tugas bahasa Lampung dialek Tulang Bawang*. Pusat Pembinaan dan Pengembangan Bahasa, Departemen Pendidikan Nasional.

Septianingtias, V., Wahya, Nur, T., & Ariyani, F. (2024). Lexical variation in the Lampung language, Indonesia. *Cogent Arts and Humanities*, *11*(1). https://doi.org/10.1080/23311983.2024.2309740

Setiawan, I., & Kao, H. Y. (2024). SUSTEM: An Improved Rule-based Sundanese Stemmer. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *23*(6). https://doi.org/10.1145/3656342

Setiawan, R., Kurniawan, A., Budiharto, W., Kartowisastro, I. H., & Prabowo, H. (2016, September 6). Flexible affix classification for stemming Indonesian Language. *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2016*. https://doi.org/10.1109/ECTICon.2016.7561257

Shakib, M. S. S., Ahmed, T., & Azharul Hasan, K. M. (2019, September 1). Designing a Bangla Stemmer using rule based approach. *2019 International Conference on Bangla Speech and Language Processing, ICBSLP 2019*. https://doi.org/10.1109/ICBSLP47725.2019.201533

Shaukat, S., Asad, M., & Akram, A. (2023). Developing an Urdu Lemmatizer Using a Dictionary-Based Lookup Approach. *Applied Sciences*

(Switzerland), *13*(8). https://doi.org/10.3390/app13085103

Shi, S., Hu, K., Xie, J., Guo, Y., & Wu, H. (2024). Robust scientific text classification using prompt tuning based on data augmentation with L2 regularization. *Information Processing and Management*, *61*(1). https://doi.org/10.1016/j.ipm.2023.103531

Shorten, C., Khoshgoftaar, T. M., & Furht, B. (2021). Text Data Augmentation for Deep Learning. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00492-0

Siagh, A., Laallam, F. Z., Kazar, O., Salem, H., & Benglia, M. E. (2024). IDA: An Imbalanced Data Augmentation for Text Classification. *Communications in Computer and Information Science*, *1940 CCIS*, 241–251. https://doi.org/10.1007/978-3-031-46335-8_19

Singh, G., Bhandari, R., & Singh, P. (2024). Development and Implementation of a Novel Hybrid Stemmer for Punjabi NLP: Integrating Rule-Based and Dictionary-Based Algorithms. *2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science, AMATHE 2024*. https://doi.org/10.1109/AMATHE61652.2024.10582252

Singh, J., & Gupta, V. (2016). Text stemming: Approaches, applications, and challenges. *ACM Computing Surveys*, *49*(3). https://doi.org/10.1145/2975608

Singh, J., & Gupta, V. (2017). A systematic review of text stemming techniques. *Artificial Intelligence Review*, *48*(2), 157–217. https://doi.org/10.1007/s10462-016-9498-2

Sovia, R., Defit, S., & Yuhandri. (2022). Development of the Minangkabau Local Language Translation Machine Based on Stemming. *Proceeding - 2022 International Symposium on Information Technology and Digital Innovation: Technology Innovation During Pandemic, ISITDI 2022*, 195–198. https://doi.org/10.1109/ISITDI55734.2022.9944457

Sovia, R., Defit, S., Yuhandri, & Sulastri. (2023). Development of natural language processing on morphology-based Minangkabau language stemming algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, *31*(1), 542–552. https://doi.org/10.11591/ijeecs.v31.i1.pp542-552

Soyusiawaty, D., Jones, A. H. S., & Lestariw, N. L. (2020). The Stemming Application on Affixed Javanese Words by using Nazief and Adriani Algorithm. *IOP Conference Series: Materials Science and Engineering*, *771*(1). https://doi.org/10.1088/1757-899X/771/1/012026

Suci, F. W., Hayatin, N., & Munarko, Y. (2022). In-idris: Modification of idris stemming algorithm for indonesian text. *IIUM Engineering Journal*, *23*(1), 82–94. https://doi.org/10.31436/IIUMEJ.V23I1.1783

Sujana, Y., & Kao, H. Y. (2023). LiDA: Language-Independent Data Augmentation for Text Classification. *IEEE Access*, *11*, 10894–10901. https://doi.org/10.1109/ACCESS.2023.3234019

Sutedi, A., Elsen, R., & Nasrulloh, M. R. (2021). Sundanese Stemming using Syllable Pattern. *Jurnal Online Informatika*, *6*(2), 218. https://doi.org/10.15575/join.v6i2.812

Sutedi, A., Nasrulloh, M. R., & Elsen, R. (2022). Multi Rule-based and Corpus-based for Sundanese Stemmer. *Jurnal Online Informatika*, *7*(2), 184–192. https://doi.org/10.15575/join.v7i2.846

Suyanto, S., SuNyoto, A., Ismail, R. N., Rachmawati, E., & Maharani, W. (2022). Stemmer and phonotactic rules to improve n-gram tagger-based indonesian phonemicization. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 3807–3814. https://doi.org/10.1016/j.jksuci.2021.01.006

Taheri, A., Zamanifar, A., & Farhadi, A. (2024). Enhancing aspect-based sentiment analysis using data augmentation based on back-translation. *International Journal of Data Science and Analytics*. https://doi.org/10.1007/s41060-024-00622-w

Vajda, S., Junaidi, A., & Fink, G. A. (2011). A semi-supervised ensemble learning approach for character labeling with minimal human effort. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 259–263. https://doi.org/10.1109/ICDAR.2011.60

Vista, C. B., Lestari, D. P., & Widyantoro, D. H. (2019). Text Corpus Augmentation to Represent Filled Pause in Indonesian Spontaneous Speech Recognition System. *Proceedings - 2019 International Conference on Advanced Informatics: Concepts, Theory, and Applications, ICAICTA 2019*. https://doi.org/10.1109/ICAICTA.2019.8904168

Wardani, N. W., & Nugraha, P. G. S. C. (2020). Stemming Teks Bahasa Bali dengan Metode Enhanced Confix Stripping. *International Journal of Natural Science and Engineering*, *4*(3), 103–113. https://doi.org/10.23887/ijnse.v4i3.30309

Wei, J., & Zou, K. (2019). *EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Wibawa, A. P., Dwiyanto, F. A., Zaeni, I. A. E., Nurrohman, R. K., & Afandi, A. (2020). Stemming javanese affix words using nazief and adriani modifications. *Jurnal Informatika*, *14*(1), 36. https://doi.org/10.26555/jifo.v14i1.a17106

Wibawa, A. P., & Hakim, M. N. (2021). Stemming Bahasa Jawa Menggunakan Damerau Levenshtein Distance (Dld). *Jurnal Teknik Informatika*, *14*(1), 22-27. https://doi.org/10.15408/jti.v14i1.15010

Wibowo, S. H., Toyib, R., Muntahanah, M., & Darnita, Y. (2022). Time

complexity in rejang language stemming. *Jurnal Infotel*, *14*(3), 174–179. https://doi.org/10.20895/infotel.v14i3.764

Wibowo, S. H., & Wibowo, S. (2019). Development of Stemming Algorithm for Rejang Language Stemmer Based on Rejang Language Morphology View project Development of Stemming Algorithm for Rejang Language Stemmer Based on Rejang Language Morphology. *Article in Journal of Advanced Research in Dynamical and Control Systems*, *11*. https://www.researchgate.net/publication/341307354

Widjaja, M., & Hansun, S. (2015). Implementation of porter"s modified stemming algorithm in an Indonesian word error detection plugin application. *International Journal of Technology*, *6*(2), 139–150. https://doi.org/10.14716/ijtech.v6i2.456

Wijono, S. H., Alhamidi, M. R., Hilman, M. H., & Jatmiko, W. (2021). Canonical Segmentation Using Affix Characters as a Unit on Transformer for Javanese Language. *Proceedings - IWBIS 2021: 6th International Workshop on Big Data and Information Security*, 67–72. https://doi.org/10.1109/IWBIS53353.2021.9631839

Winarti, T., Kerami, D., Lussiana, E. T. P., & Sudiro, S. A. (2017). Improving stemming algorithm using morphological rules. *International Journal on Advanced Science, Engineering and Information Technology*, *7*(5), 1758-1764.

Winarti, T., Kerami, J., & Arief, S. (2017). Determining Term on Text Document Clustering using Algorithm of Enhanced Confix Stripping Stemming. In *International Journal of Computer Applications* (Vol. 157, Issue 9).

Xu, L., Xie, H., Wang, F. L., & Wang, W. (2022). Recent data augmentation techniques in natural language processing: A brief survey. *The IEEE Intelligent Informatics Bulletin*, *22*(1), 29-37.

Yusra, Fikry, M., & Hendri. (2021). Stemmer bahasa melayu riau berdasarkan aturan morfologi. In *Seminar Nasional Teknologi Informasi Komunikasi dan Industri* (pp. 118-124).

Zaiton, H., & Alansary, S. (2025). Natural Language Processing Approaches to Text Data Augmentation: A Computational Linguistic Analysis. *International Journal of Arabic-English Studies (IJAES)*, *25*(1).

Zampieri, M., Nakov, P., & Scherrer, Y. (2020). Natural language processing for similar languages, varieties, and dialects: A survey. *Natural Language Engineering*, *26*(6), 595-612.