Jurnal Fokus Elektroda : Energi Listrik, Telekomunikasi, Komputer, Elektronika dan Kendali)
Volume 07 No 04, Tahun 2022: Hal. 237-243
e-ISSN: 2502-5562. Open Access at: https://elektroda.uho.ac.id/
Penerbit : Jurusan Teknik Elektro Universitas Halu Oleo Kendari Sulawesi Tenggara,

Optimasi Normalisasi Kata Pada Data Twitter Untuk Meningkatkan Akurasi Analisis Sentimen (Studi Kasus Respon Masyarakat Terhadap Layanan Teman Bus)

M. Adnan Nur[1], Nurilmiyanti Wardhani[2]

[1,2] Teknik Informatika, STMIK Handayani Makassar

Coprespondent Author : M. Adnan Nur

Abstract **— Friends Bus is a service provided by the Indonesian Ministry of Transportation. To determine the level of community satisfaction with the Teman Bus service, sentiment analysis can be applied using Twitter social media data. Unstructured text on Twitter becomes a problem in sentiment analysis, especially word misspelling and slang usage. The purpose of this study is to optimize word normalization by applying spelling corrections and converting slang words into standard words. The research stages consist of data collection, pre-processing, sentiment analysis and accuracy testing. The data used consists of twitter data with the keywords bus friend, basic word dataset (standard words) and slang word dataset. For pre-processing, the steps include tokenizing, case folding, filtering, stemming, spelling correction with Levenshtein distance and conversion of slang words using the word2vec model. The distribution of training data and test data for testing the classification of sentiment analysis using k-fold cross validation. In the testing phase, 5 test scenarios were prepared by setting the Levenshtein distance and word2vec parameters and there was 1 test scenario that did not involve word normalization. The results obtained at the testing stage showed an increase in accuracy by applying word normalization of 0.776. This word normalization uses a Levenshtein distance ratio of 0.9 and a min-count word2vec of 10.**

Keyword **— optimization, word normalization, sentiment analysis, teman bu.**

Abstrak — Teman Bus merupakan layanan yang disediakan oleh Kementerian Perhubungan RI. Untuk mengetahui tingkat kepuasan masyarakat terhadap layanan Teman Bus, Analsis sentimen dapat diterapkan menggunakan data media sosial *twitter*. Teks yang tidak terstruktur pada *twitter* menjadi permasalahan dalam analsis sentimen khsusnya kesalahan ejaan kata dan penggunaan kata slang. Tujuan dari penelitian ini adalah melakukan optimasi normalisasi kata dengan menerapkan koreksi ejaan kata dan konversi kata slang menjadi kata baku. Tahapan penelitian terdiri atas pengumpulan data, prapemrosesan, analsis sentimen dan pengujian akurasi. Data yang digunakan terdiri atas data *twitter* dengan kata kunci teman bus, *dataset* kata dasar (kata baku) dan *dataset* kata slang. Untuk prapemrosesan, tahapannya meliputi *tokenizing*, *case folding*, *filtering*, *stemming*, koreksi ejaan kata dengan *levenshtein distance* dan konversi kata slang dengan model *word2vec*. Pembagian data latih dan data uji untuk pengujian klasifikasi analisis sentimen menggunakan *k-fold cross validation*. Tahap pengujian disiapkan 5 skenario pengujian dengan pengaturan parameter *levenshtein distance* dan *word2vec* serta terdapat 1 skenario pengujian yang tidak melibatkan normalisasi kata. Hasil yang diperoleh pada tahap pengujian menunjukkan peningkatan akurasi dengan menerapkan normalisasi kata sebesar 0,776. Normalisasi kata ini menggunakan *ratio levenshtein distance* sebesar 0,9 dan *min-count word2vec* sebesar 10.

Kata kunci — optimasi, normalisasi kata, analsis sentimen, teman bus.

I. PENDAHULUAN

Media sosial telah menjadi wadah dalam menyampaikan pendapat maupun mengekspresikan diri bagi para penggunanya. Penyampaian tingkat kepuasan terhadap produk maupun jasa pada media sosial telah menjadi informasi umum yang ditemukan pada halaman beranda media sosial. Informasi seperti ini tentunya dapat menjadi bahan evaluasi bagi penyedia produk maupun jasa untuk meningkatkan kualitas maupun layanannya dengan mengolahnya lebih lanjut menggunakan *Text Mining* berupa analisis sentimen. Salah satu layanan masyarakat yang disediakan oleh Pemerintah saat ini melalui Kementerian Perhubungan Republik Indonesia adalah Teman Bus. Teman Bus merupakan angkutan umum di kawasan perkotaan berbasis jalan yang menggunakan teknologi telematika yang andal dan berbasis non tunai untuk meningkatkan keselamatan dan keamanan serta kenyamanan mobilisasi [1]. Analisis sentimen terhadap tingkat kepuasan masyarakat pada layanan Teman Bus yang disampaikan melalui media sosial digunakan dalam penelitian ini sebagai studi kasus.

Salah satu media sosial yang signifikan digunakan dalam beberapa penelitian terakhir terkait *Text Mining* khususnya analisis sentimen adalah *twitter* [2]. Pengguna *twitter* umumnya melakukan interaksi melalui teks dengan menggunakan bahasa sehari-hari yang tidak baku serta singkatan. Selain itu, memungkinkan pula terjadinya kesalahan penulisan kata dalam penyampaiannya [3]. Teks yang tidak terstruktur tersebut membutuhkan normalisasi kata pada tahapan prapemrosesan sebelum melakukan analisis sentimen. Prapemrosesan teks merupakan langkah penting pada analisis sentimen karena memilih metode dalam prapemrosesan yang tepat dapat meningkatkan akurasi [4]. Prapemrosesan teks yang menjadi standar dalam analisis sentimen terdiri atas *case folding*, *tokenizing*, *stop words* dan *stemming* [5]. Tahapan standar ini tidak melibatkan koreksi kata sehingga memungkinkan adanya kata yang memiliki kesalahan ejaan ikut dalam pembobotan kata proses analisis sentimen. Selain itu, adanya singkatan

dan kata slang juga mempengaruhi tingkat akurasi dari analsis sentimen. Kata slang merupakan kata atau frasa yang sangat informal, biasanya digunakan oleh kelompok sosial atau golongan umur tertentu yang cenderung menggambarkan hal negatif, tabu atau ekstrim [6]. Pada penelitian ini, tahapan normalisasi kata ditambahkan dalam prapemrosesan dan ditempatkan setelah tahapan *tokenizing* untuk meningkatkan akurasi. Tahapan ini terbagi menjadi beberapa proses, yaitu melakukan koreksi kesalahan ejaan kata, pembuatan kamus untuk mengubah kata singkatan menjadi kata baku serta melakukan konversi kata slang menjadi kata yang terdaftar pada Kamus Besar Bahasa Indonesia (KBBI). Adapun algoritma yang digunakan dalam koreksi kesalahan ejaan kata yaitu *levenshtein distance* dan untuk konversi kata slang menggunakan model *word2vec*.

II. TINJAUAN PUSTAKA

A. Analisis Sentimen

Analisis sentimen merupakan salah satu bidang pemodelan *mechine learning* yang menggunakan teks. Proses pelatihan pada analisis sentimen cenderung lebih sulit dibandingkan bidang *mechine learning* lainnya. Algoritma yang digunakan pada penelitian ini adalah *Naïve Bayes*. Algoritma tersebut telah digunakan pada beberapa penelitian analisis sentimen sebelumnya seperti yang dilakukan oleh Rustiana et al (2017) dengan judul penelitian analisis sentimen pasar otomotif mobil pada *twitter* menggunakan *naïve bayes* [7]. Akurasi yang dihasilkan penelitian tersebut mencapai 93%. Terdapat pula penelitian dengan judul analisis sentimen terhadap opini masyarakat tentang vaksin *Covid-19* menggunakan algoritma *naïve bayes classifier* oleh Yulita et al (2021) [8]. Penelitian tersebut menghasilkan akurasi yang sama yaitu 93%.

B. Prapemrosesan Analisis Sentimen

Analisis sentimen terhadap data *twitter* yang tidak terstruktur membutuhkan tahapan *preprocessing*. Terdapat beberapa penelitian yang membahas *preprocessing* analisis sentimen diantaranya penelitian dengan judul *the effect of preprocessing techniques on Twitter sentiment analysis* yang dibuat oleh Krouska et al (2016)[4]. Penelitian tersebut menggambarkan bahwa akurasi dapat ditingkatkan dengan pemilihan fitur dan representasi yang tepat pada tahap *preprocessing*. Penelitian serupa juga dilakukan oleh Khairunnis et al (2021) berjudul pengaruh *text preprocessing* terhadap analisis sentimen komentar masyarakat pada media sosial *twitter* (studi kasus pandemi *Covid-19*) [9]. Penelitian tersebut melibatkan beberapa tahapan dalam *preprocessing* yaitu *case folding*, normalisasi kata, *cleaning*, *stopword* dan *stemming*. Tahapan normalisasi kata memberikan peningkatan akurasi dari 68,51 % menjadi 72,68%. Hasil ini menunjukkan bahwa tahapan normalisasi kata dalam *preprocessing* mempengaruhi akurasi dari analisis sentimen. Dalam penelitian ini, penulis berfokus pada tahapan normalisasi dengan melakukan optimasi melalui koreksi ejaan kata dan konversi *slang word* menjadi kata baku.

C. Koreksi Kesalahan Ejaan Kata

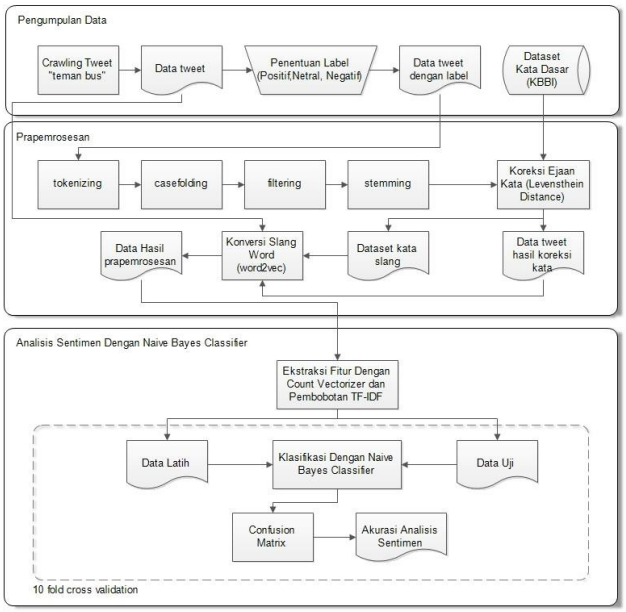
Terdapat beberapa metode maupun algoritma yang dapat digunakan dalam koreksi ejaan kata dan salah satunya adalah *levenshtein distance*. Penelitian yang memanfaatkan algoritma *levenshtein distance* pernah dibuat oleh Sadiah et al (2020) dengan judul *autocorrect* pada modul pencarian *drugs e-dictionary* menggunakan algoritma *levenshtein distance*. Hasil penelitian tersebut memiliki nilai akurasi, *recall* dan *precision* yang tinggi yaitu 90% [10]. Penulis juga telah melakukan penelitian sebelumnya pada tahun 2021 yang membandingkan algoritma *levenshtein distance* dan *jaro-winkler distance* dalam koreksi kata *preprocessing* analisis sentimen [11]. Hasil penelitian menunjukkan algoritma *levenshtein distance* menghasilkan tingkat akurasi analisis sentimen yang lebih tinggi. Namun, dalam penelitian tersebut, normalisasi kata yang dilakukan hanya pada kesalahan ejaan kata. Pengembangan dari penggunaan algoritma ini dilakukan oleh Wibawa et al (2020), *levenshtein distance* dikolaborasikan dengan metode empiris untuk melakukan koreksi ejaan bahasa indonesia [12]. Penambahan metode empiris ini meningkatkan akurasi dari 73% menjadi 97%. Berdasarkan penelitian tersebut, penulis menambahkan metode empiris yang dikombinasikan dengan algoritma *levenshtein distance* untuk melakukan koreksi ejaan kata dalam mengoptimasi normalisasi kata.

d. Normalisasi Kata Slang

Penggunaan *twitter* yang melibatkan berbagai kelompok sosial dan budaya membuat *tweet* banyak diisi dengan kata slang sesuai dengan budaya dan bahasa dari penggunanya. Salah satu penelitian yang menjadikan kata slang sebagai sobjek penelitian dibuat oleh Rosalina et al (2020) dengan judul penggunaan bahasa slang di media sosial *twitter* [13]. Penelitian tersebut mengklasifikasikan penggunaan kata slang menjadi tiga yaitu slang dalam bentuk singkatan, slang berdasarkan ruang lingkup penggunaan, dan slang berdasarkan fungsinya. Hasil klasifikasi tersebut dapat digunakan dalam penelitian ini untuk mengelompokkan *dataset* konversi kata slang. Untuk penelitian yang berkaitan dengan konversi kata slang menjadi kata baku pernah dibuat oleh Riyaddulloh et al (2021). Penelitiannya menggunakan model *word2vec* untuk menormalisasi kata slang menjadi kata baku dengan *dataset*. Akurasi yang dihasilkan dari evaluasi penelitian tersebut mencapai 91% [14].

III. METODE PENELITIAN

Penelitian melibatkan beberapa tahapan dalam penerapan optimasi normalisasi kata. Tahapan tersebut diuraikan pada diagram alir metode penelitian berikut.



Gambar 1. Diagram Alir metode Penelitian

A. Pengumpulan Data

Data yang digunakan dalam penelitian ini terdiri atas data *tweet*, *dataset* kata dasar dan *dataset* kata slang. Berikut metode dan hasil pengumpulan data yang dilakukan dalam penelitian.

1) Data *Tweet* Teman Bus
   Dataset tweet dengan kata kunci teman bus merupakan data yang digunakan dalam penelitian ini sebagai contoh kasus untuk menguji peningkatan akurasi analisis sentimen dari metode yang diterapkan. Tahapan dalam mendapatkan data ini dimulai dengan membuat perangkat lunak *crawling tweet* menggunakan bahasa pemrograman *python* dengan *library tweepy*. *Tweepy* merupakan salah satu *library* bahasa pemrograman *python* dalam mengakses *application programming interface* yang disediakan oleh *twitter* untuk melakukan *crawling tweet* [15]. Aplikasi *crawling* ini selanjutnya digunakan untuk mendapatkan *dataset tweet* dengan menggunakan kata kunci teman bus. Terdapat 1.445 *tweet* yang diambil dari bulan juni hingga juli tahun 2022. Setiap *tweet* yang terdapat pada *dataset* ini diberi label positif, netral dan negatif untuk kebutuhan klasifikasi analisis sentimen. *Dataset* dengan label positif berjumlah 316, netral 833 dan negatif 296.

2) *Dataset* Kata Dasar (KBBI)
   Dataset kata dasar yang terdaftar pada Kamus Besar Bahasa Indonesia (KBBI) dibutuhkam dalam prapemrosesan khususnya untuk *stemming*, koreksi ejaan kata dan konversi *slang word* menjadi kata baku. Kata dasar ini diperoleh dari *library sastrawi* untuk bahasa pemrograman *python*. *Library sastrawi* bertujuan dalam mengubah kata berimbuhan menjadi kata dasar dengan menerapkan algoritma *Nazief* dan *Andriani* serta ditingkatkan dengan *Enhanced Confix Stripping* [16]. Dalam *library* ini juga terdapat *dataset* kata dasar yang berjumlah 29.933 kata.

3) *Dataset Slang Words*
   Kata slang merupakan kata yang tidak terdaftar pada Kamus Besar Bahasa Indonesia (KBBI) yang dapat berupa singkatan atau istilah yang digunakan dalam kelompok masyarakat tertentu [5]. Dalam penelitian ini, kata slang ditempatkan pada *dataset* yang bertujuan untuk menampung kata slang yang terdapat pada *dataset tweet* namun tidak ada pada *dataset* kata dasar. Kata slang yang didapatkan dari *dataset tweet* berjumlah 174 kata. Dataset kata slang ini selanjutnya digunakan dalam tahap prapemrosesan konversi kata slang.

B. Prapemrosesan

1) *Tokenizing*
   Tahap prapemrosesan diawali dengan memisahkan kata pada setiap kalimat *tweet* menjadi *token* kata [17]. *Tokenizing* ini dilakukan untuk memudahkan dalam tahapan prapemrosesan selanjutnya. Hasil *tokenizing* pada bahasa pemrograman *python* yang digunakan ditempatkan pada struktur data *list.*

2) *Casefolding*
   Dalam tahap prapemprosesan ini terdapat proses untuk menghitung tingkat kemiripan kata sehingga dibutuhkan *casefolding* untuk setiap *token* kata. *Casefolding* bertujuan untuk mengubah seluruh huruf kapital menjadi huruf kecil pada setiap *tweet* [18]. Penerapan *casefolding* dalam penelitian ini menggunakan fungsi *lower()* yang disediakan bahasa pemrograman *python* pada setiap *token* kata.

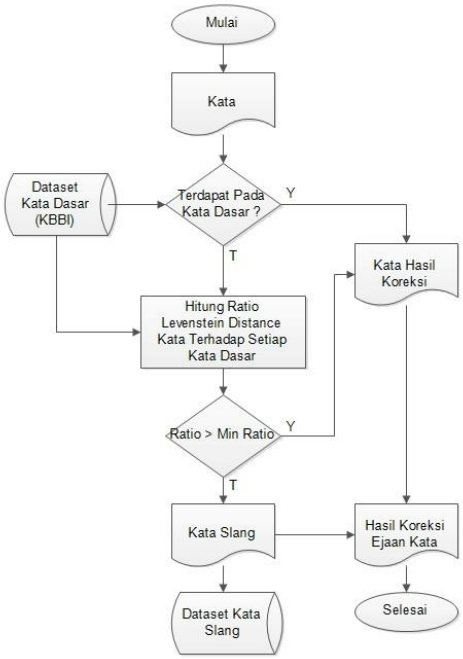
3) *Filtering*
   Dalam penelitian ini, terdapat beberapa kata atau istilah yang dihilangkan untuk mengoptimalkan waktu dalam proses klasifikasi. Adapun kata atau istilah yang dihilangkan meliputi *username* yang diawali dengan simbol @, tagar yang diawali dengan simbol #, alamat *website* yang diawali dengan *http://* atau *https://* atau *www* serta kata yang mengandung angka. Selain itu, pada tahap *filtering* ini juga diterapkan *stopword removal* untuk mereduksi jumlah kata pada *dataset*. *Stopword* adalah kata yang dapat dihilangkan dalam prapemrosesan bahasa alami karena tidak mempengaruhi hasil dari analisis teks [19]. Dalam menerapkan *stopword removal* ini digunakan *dataset stopword* yang berasal dari *class StopWordRemoverFactory* dari *library sastrawi* untuk bahasa pemrograman *python.*

4) Stemming

   Stemming bertujuan untuk mengubah kata berimbuhan menjadi kata dasar. Penerapan *stemming* dalam tahap

prapemrosesan dapat meningkatkan performansi dari klasifikasi [20][21]. *Library sastrawi* digunakan dalam penelitian ini untuk menjalankan tahap *stemming*.

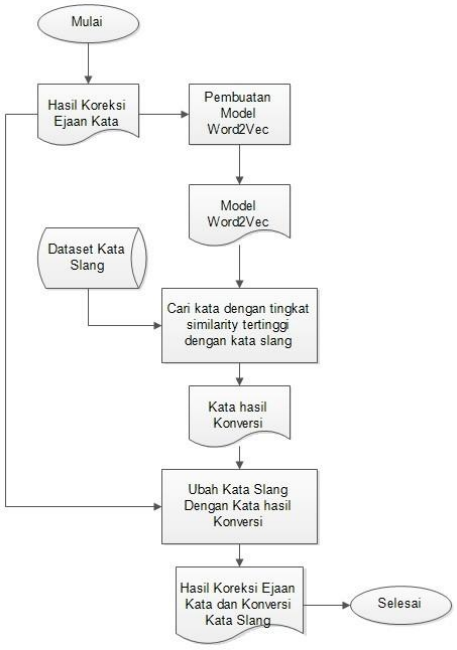
5) Koreksi Kesalahan Ejaan Kata
Setelah melewati tahap *tokenizing* hingga *stemming*, kata yang ada pada *dataset* selanjutnya masuk pada tahap normalisasi kata yang terdiri atas koreksi ejaan kata dan konversi kata slang. Berikut alur dari tahap koreksi ejaan kata.



Gambar 2. Alur Koreksi Ejaan Kata

Setiap *token* kata yang ada pada *dataset* diperiksa tingkat kemiripannya dengan setiap kata yang ada pada *dataset* kata dasar atau baku. Jika *token* kata tersebut ada pada *dataset* kata dasar maka *token* kata langsung dimasukkan pada *dataset* hasil koreksi. Namun, jika *token* kata tersebut tidak ditemukan maka dilakukan perhitungan tingkat kemiripan menggunakan metode *lenvenshtein distance* antara *token* kata dengan setiap kata pada dataset kata dasar. Jika terdapat kata yang *ratio* tingkat kemiripannya diatas *min ratio* yang telah ditentukan maka kata tersebut dimasukkan pada *dataset* hasil koreksi. Untuk *token* kata yang yang *ratio* kemiripannya terhadap seluruh kata pada *dataset* kata dasar dibawah *min ratio* dimasukkan pada *dataset* kata slang dan hasil koreksi. Dalam penerapan koreksi ejaan kata ini, terdapat 2 nilai *min ratio* yang diujikan yaitu 0,75 dan 0,9. Pada *min ratio* 0,9 terdapat 414 *token* kata yang dikoreksi sedangkan untuk *min ratio* 0,75 berjumlah 3.896 *token* kata yang dikoreksi.

6) Konversi Kata Slang
Kata slang merupakan kata atau istilah tidak baku yang digunakan oleh kelompok tertentu dalam berkomunikasi dan umumnya digunakan dalam jangka waktu tertentu. Penggunaan kata slang ini banyak ditemukan pada media sosial [13]. Berdasarkan hal tersebut maka dalam penelitian ini dilakukan konversi kata slang menjadi kata baku untuk meningkatkan akurasi klasifikasi dengan alur seperti gambar dibawah ini.



Gambar 3. Alur Konversi Kata Slang

Tahap konversi kata slang diawali dengan pembuatan model *word2vec* dengan menggunakan *dataset* hasil koreksi ejaan kata. Model *word2vec* ini selanjutnya digunakan untuk mencari kata baku yang memiliki tingkat *similarity* tertinggi dengan kata slang sebagai kata hasil konversi kata slang. Terdapat 2 parameter *min count* kata baku hasil konversi yang diujikan yaitu 5 dan 10. Kata hasil konversi ini menggantikan kata slang yang ada pada *dataset* hasil konversi ejaan kata.

C. Analisis Sentimen Dengan Naïve Bayes Classifier

Dataset hasil prapemrosesan digunakan sebagai data latih dan data uji dalam klasifikasi analisis sentimen. Tahap ini diawali dengan melakukan pembobotan kata menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF). Penentuan parameter minimum dan maksimum *document frequency* (*min-df* dan *max-df*) dalam TF-IDF mempengaruhi hasil akurasi klasifikasi [22]. Dalam penelitian ini, *min-df* yang digunakan adalah 0,0005 dan *max-df* diatas 0,5.

Dataset hasil hasil pembobotan kata dengan TF-IDF selanjutnya dibagi menjadi data latih dan data uji menggunakan *k-fold cross validation*. Metode ini mengevaluasi kinerja model yang digunakan dengan mengelompokkan dan melakukan perulangan data latih dan data uji sebanyak K [23]. Nilai K dalam penelitian ini ditentukan dengan nilai 10. Pada setiap perulangan, klasifikasi analisis sentimen diterapkan menggunakan algoritma *naïve bayes*. Algoritma ini merupakan salah satu algoritma *machine learning* yang memiliki akurasi tinggi dibandingkan beberapa algoritma lainnya [24][25][26]. Model klasifikasi *naïve bayes* dibentuk menggunakan data latih. Setelah model klasifikasi terbentuk maka data uji digunakan untuk menghitung tingkat akurasinya.

IV. HASIL DAN PEMBAHASAN

Terdapat 5 skenario yang diterapkan dalam pengujian dengan rentang tingkat akurasi 0 hingga 1. Berikut hasilnya:

a. Skenario 1: Tanpa Normalisasi Kata
Untuk skenario pertama ini, tahap prapemrosesan tidak melibatkan proses normalisasi kata. Berikut hasil perhitungan akurasi analisis sentimennya.

Tabel 1. Hasil Pengujian Skenario 1

10-Fold Cross Validation					
Akurasi	*fold-1*	*fold-2*	*fold-3*	*fold-4*	*fold-5*
	0.676	0.634	0.690	0.690	**0.752**
	fold-6	*fold-7*	*fold-8*	*fold-9*	*fold-10*
	0.708	0.653	0.722	**0.590**	0.708

Dalam 10 percobaan yang dilakukan, diperoleh akurasi tertinggi sebesar 0,752 pada *fold-5*. Untuk akurasi terendah sebesar 0,590 pada *fold-9*.

b. Skenario 2: Normalisasi Kata dengan *Ratio Levenshtein Distance* 0,75 dan *Min-Count Word2Vec* 10
Pada skenario ini, normalisasi kata diterapkan. Untuk koreksi ejaan kata menggunakan *ratio levenshtein distance* 0,75 dan konversi kata slang dengan *min-count word2vec* 10. Berikut hasil perhitungan akurasinya:

Tabel 2. Hasil Pengujian Skenario 2

10-Fold Cross Validation					
Akurasi	*fold-1*	*fold-2*	*fold-3*	*fold-4*	*fold-5*
	0.657	0.643	0.692	0.706	**0.769**
	fold-6	*fold-7*	*fold-8*	*fold-9*	*fold-10*
	0.748	0.657	0.678	**0.601**	0.678

Akurasi tertinggi yang diperoleh pada skenario ini sebesar **0,769**. Akurasi tersebut diperoleh pada *fold-5* sedangkan akurasi terendah ada pada *fold-9* sebesar *0,601*.

c. Skenario Pengujian 3: *Ratio Levenshtein Distance* 0,75 dan *Min-Count Word2Vec* 5
Skenario pengujian ke-3 ini menetapkan *ratio levenstein distance* untuk koreksi ejaan kata sama dengan pengujian ke-2 sebesar 0,75. Namun untuk *min-count word2vec* diturunkan menjadi 5. Berikut hasil pengujian untuk skenario 3:

Tabel 3. Hasil Pengujian Skenario 3

10-Fold Cross Validation					
Akurasi	*fold-1*	*fold-2*	*fold-3*	*fold-4*	*fold-5*
	0.643	0.650	0.692	0.706	**0.776**
	fold-6	*fold-7*	*fold-8*	*fold-9*	*fold-10*
	0.727	0.657	0.685	**0.587**	0.678

Skenario pengujian menghasilkan akurasi tertinggi sebesar 0,776 yang posisinya sama dengan skenario sebelumnya yaitu pada *fold-5*. Akurasi terendah yang diperoleh sebesar 0,587 pada *fold-9*.

d. Skenario Pengujian 4: *Ratio Levenshtein Distance* 0,9 dan *Min-Count Word2Vec* 10
Pada skenario ke-4, *ratio levenshteon distance* ditingkatkan sebesar 0,9. Untuk *min-count word2vec* sebesar 10. Parameter tersebut menghasilkan tingkat akurasi pada tabel berikut:

Tabel 4. Hasil Pengujian Skenario 4

10-Fold Cross Validation					
Akurasi	*fold-1*	*fold-2*	*fold-3*	*fold-4*	*fold-5*
	0.615	0.643	0.678	0.692	**0.741**
	fold-6	*fold-7*	*fold-8*	*fold-9*	*fold-10*
	0.727	0.636	0.713	**0.601**	0.650

Akurasi tertinggi yang diperoleh sebesar 0,741 pada *fold-5*. Sedangkan akurasi terendah sebesar 0,601 di *fold-9*.

e. Skenario Pengujian 5: *Ratio Levenshtein Distance* 0,9 dan *Min-Count Word2Vec* 10
Koreksi ejaan kata dengan *ratio* 0,9 serta konversi kata slang dengan *min-count* 10 diterapkan pada pengujian ini sebagai skenario pengujian terakhir dari kombinasi parameter yang ditetapkan. Berikut hasil akurasi yang diperoleh.

Tabel 5. Hasil Pengujian Skenario 5

10-Fold Cross Validation					
Akurasi	*fold-1*	*fold-2*	*fold-3*	*fold-4*	*fold-5*
	0.657	0.657	0.650	0.692	0.734
	fold-6	*fold-7*	*fold-8*	*fold-9*	*fold-10*
	0.776	0.657	0.699	**0.601**	0.664

Pada skenario ini, terdapat perbedaan posisi *fold* dari akurasi tertinggi yang didapatkan yaitu pada *fold*-6. Nilai akurasi tertinggi yang diperoleh sebesar 0,776. Untuk akurasi terendah posisinya sama dengan skenario lainnya pada *fold*-6 sebesar 0,601.

Berdasarkan 5 skenario pengujian yang dilakukan pada penelitian ini, diperoleh tingkat akurasi tertinggi sesesar 0,776 pada skenario ke-3 dan skenario ke-5. Akurasi tersebut didapatkan setelah menerapkan normalisasi kata dengan *ratio levenshtein distance* untuk koreksi ejaan kata sebesar 0,75 dan 0,9 serta *min-count word2vec* 5 dan 10. Namun dilihat dari akurasi terendah dari 10 kali percobaan, skenario ke-3 memperoleh akurasi terendah dibandingkan skenario lainnya. Dari hasil tersebut dapat dilihat bahwa akurasi terbaik diperoleh pada skenario ke-5 dengan *ratio levenshtein distance* sebesar 0,9 dan *min-count word2vec* sebesar 10.

VII. KESIMPULAN

Penerapan normalisasi kata pada tahap prapemrosesan analsis sentimen yang meliputi koreksi kesalahan ejaan kata dan konversi kata slang dapat meningkatkan akurasi klasifikasi. Kombinasi parameter *ratio levenshtein distance* untuk koreksi kesalahan ejaan kata dan *min-count word2vec* untuk konversi kata slang yang memiliki akurasi terbaik masing-masing adalah 0,9 dan 10.

DAFTAR ACUAN

[1] "Teman Bus #KamiAdaUntukAnda." https://temanbus.com/ (accessed Jan. 29, 2022).

[2] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi, "Twitter and Research: A Systematic Literature Review through Text Mining," *IEEE Access*, vol. 8, pp. 67698–67717, 2020, doi: 10.1109/ACCESS.2020.2983656.

[3] S. A. Salloum, M. Al-Emran, A. A. Monem, and K. Shaalan, "A survey of text mining in social media: Facebook and Twitter perspectives," *Adv. Sci. Technol. Eng. Syst.*, vol. 2, no. 1, pp. 127–133, 2017, doi: 10.25046/aj020115.

[4] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," *IISA 2016 - 7th Int. Conf. Information, Intell. Syst. Appl.*, Dec. 2016, doi: 10.1109/IISA.2016.7785373.

[5] S. Khomsah and Agus Sasmito Aribowo, "Text-Preprocessing Model Youtube Comments in Indonesian," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 4, pp. 648–654, Aug. 2020, doi: 10.29207/resti.v4i4.2035.

[6] W. Trimastuti, "AN ANALYSIS OF SLANG WORDS USED IN SOCIAL MEDIA," *J. Dimens. Pendidik. dan Pembelajaran*, vol. 5, no. 2, pp. 64–68, Jul. 2017, doi: 10.24269/DPP.V5I2.497.

[7] D. Rustiana and N. Rahayu, "ANALISIS SENTIMEN PASAR OTOMOTIF MOBIL: TWEET TWITTER MENGGUNAKAN NAÏVE BAYES," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 8, no. 1, pp. 113–120, Apr. 2017, doi: 10.24176/simet.v8i1.841.

[8] W. Yulita, "Analisis Sentimen Terhadap Opini Masyarakat Tentang Vaksin Covid-19 Menggunakan Algoritma Naïve Bayes Classifier," *J. Data Min. dan Sist. Inf.*, vol. 2, no. 2, pp. 1–9, Aug. 2021, doi: 10.33365/JDMSI.V2I2.1344.

[9] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 2, pp. 406–414, Apr. 2021, doi: 10.30865/MIB.V5I2.2835.

[10] H. T. Sadiah, M. Saad, N. Ishlah, and N. N. Rokhmah, "Autocorrect on Drugs e-Dictionary Search Module Using Levenshtein Distance Algorithm," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 1, pp. 64–69, Feb. 2020, doi: 10.29207/RESTI.V4I1.1401.

[11] M Adnan Nur, "Perbandingan Levenshtein Distance Dan Jaro-Winkler Distance Untuk Koreksi Kata Dalam Preprocessing Analisis Sentimen Pengguna Twitter," *J. Fokus Elektroda Energi List. Telekomun. Komputer, Elektron. dan Kendali)*, vol. 6, no. 2, pp. 88–93, Jun. 2021, doi: 10.33772/JFE.V6I2.17751.

[12] A. P. Wibawa, P. Yuliawati, P. Santoso, R. Shalahuddin, and I. M. Wirawan, "Damerau Levenshtain Distance dengan Metode Empiris untuk Koreksi Ejaan Bahasa Indonesia," *Ilk. J. Ilm.*, vol. 12, no. 3, pp. 176–182, Dec. 2020, doi: 10.33096/ilkom.v12i3.600.176-182.

[13] R. Rosalina, A. Auzar, and H. Hermandra, "Penggunaan Bahasa Slang di Media Sosial Twitter," *J. TUAH Pendidik. dan Pengajaran Bhs.*, vol. 2, no. 1, pp. 77–84, Jun. 2020, doi: 10.31258/JTUAH.2.1.P.77-84.

[14] R. Riyadulloh and A. Romadhony, "Normalisasi Teks Bahasa Indonesia Berbasis Kamus Slang Studi Kasus: Tweet Produk Gadget Pada Twitter," *eProceedings Eng.*, vol. 8, no. 4, Aug. 2021, Accessed: Feb. 04, 2022. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15246/14969

[15] R. T. Swaminathan, V. Balaji, and S. Subramanian, "Sentiment Analysis of Twitter Data using Tweepy and TextBlob," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 12, pp. 785–789, Dec. 2021, doi: 10.22214/ijraset.2021.39391.

[16] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing For Student Complaint Document

Classification Using Sastrawi," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 874, no. 1, p. 012017, Jun. 2020, doi: 10.1088/1757-899X/874/1/012017.

[17] A. A. Kurniawan and M. Mustikasari, "Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia," *J. Inform. Univ. Pamulang*, vol. 5, no. 4, pp. 544–552, Dec. 2020, doi: 10.32493/INFORMATIKA.V5I4.6760.

[18] E. Wita, "Penerapan Natural Language Processing Untuk Mengidentifikasi Kalimat Ambigu pada Surat Kabar Menerapkan Metode Shift Reduce Parsing," *BEES Bull. Electr. Electron. Eng.*, vol. 2, no. 2, pp. 63–66, Nov. 2021, doi: 10.47065/BEES.V2I2.996.

[19] A. P. Wibawa, F. Miftahuddin, and S. Suyono, "K-Medoids Clustering untuk Pembentukan Database Stopword Bahasa Jawa," *Ranah J. Kaji. Bhs.*, vol. 10, no. 2, pp. 261–269, Dec. 2021, doi: 10.26499/RNH.V10I2.2125.

[20] M. A. Al Farisi, W. Astuti, and A. Adiwijaya, "Klasifikasi Multi-label Pada Hadis Sahih Bukhari Terjemahan Bahasa Indonesia Menggunakan Convolutional Neural Networks," *eProceedings Eng.*, vol. 8, no. 5, Oct. 2021, Accessed: Sep. 19, 2022. [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15674/15387

[21] F. N. Zamzami, A. Adiwijaya, and M. D. P, "Analisis Sentimen Terhadap Review Film Menggunakan Metode Modified Balanced Random Forest dan Mutual Information," *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 2, pp. 415–421, Apr. 2021, doi: 10.30865/MIB.V5I2.2844.

[22] N. Umar, M. A. Nur, T. Informatika, and H. Makassar, "Application of Naïve Bayes Algorithm Variations On Indonesian General Analysis Dataset for Sentiment Analysis," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, pp. 585–590, Aug. 2022, doi: 10.29207/RESTI.V6I4.4179.

[23] L. Mardiana, D. Kusnandar, and N. Satyahadewi, "ANALISIS DISKRIMINAN DENGAN K FOLD CROSS VALIDATION UNTUK KLASIFIKASI KUALITAS AIR DI KOTA PONTIANAK," *Bimaster Bul. Ilm. Mat. Stat. dan Ter.*, vol. 11, no. 1, pp. 97–102, Jan. 2022, doi: 10.26418/BBIMST.V11I1.51608.

[24] R. Ardianto, T. Rivanie, Y. Alkhalifi, F. S. Nugraha, and W. Gata, "Sentiment Analysis On e-Sposrts For Education Curriculum Using Naive Bayes And Support Vector Machine," *J. Ilmu Komput. dan Inf.*, vol. 13, no. 2, pp. 109–122, Jul. 2020, doi: 10.21609/JIKI.V13I2.885.

[25] S. Dyah Anggita and Ikmah, "Algorithm Comparation of Naive Bayes and Support Vector Machine based on Particle Swarm Optimization in Sentiment Analysis of Freight Forwarding Services," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 2, pp. 362–369, Apr. 2020, doi: 10.29207/RESTI.V4I2.1840.

[26] S. Tamrakar, B. K. Bal, and R. B. Thapa, "Aspect Based Sentiment Analysis of Nepali Text Using Support Vector Machine and Naive Bayes," *Tech. J.*, vol. 2, no. 1, pp. 22–29, Nov. 2020, doi: 10.3126/TJ.V2I1.32824.

.