Penerapan Normalisasi Kata Tidak Baku Menggunakan Levenshtein Distance pada Analisa Sentimen Layanan PT. KAI di Twitter

Imam Fahrur Rozi[1], Rizky Ardiansyah[2], Naomi Rebeka[3]

[1,2,3] Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang
[1] imam.rozi@polinema.ac.id, [2] rizky.computerscience@gmail.com, [3]naomirbc@gmail.com

***Abstrak*— Pertumbuhan media sosial saat ini, menjadi salah satu jalan bagi masyarakat untuk memberikan opini balik berupa *feedback* kepada pelayanan transportasi publik, seperti pada layanan Kereta Api Indonesia (KAI). Masyarakat akan memberikan opini atau *feedback* terhadap layanan dan infrastruktur yang diberikan. Namun, seiring bertambahnya *tweet* yang semakin banyak, pengelola Twitter PT. KAI dan masyarakat lainnya merasa kesulitan untuk membaca *tweet feedback* dan opini yang tidak memiliki arti atau kesalahan ejaan, juga kesulitan dalam menggolongkan *tweet* kedalam kategori-kategori tertentu, seperti positif, netral dan negatif. Padahal informasi *tweet* yang memiliki arti dan pengkategorian dari *feedback* masyarakat ini sangat diperlukan untuk evaluasi dan menentukan kebijakan KAI.**

Sehingga dibutuhkan suatu sistem yang dapat membantu memperbaiki kata-kata yang tidak memiliki arti serta mengklasifikasikan *tweet* ke dalam kategori positif, negatif dan netral secara otomatis. Sistem serupa sebelumnya telah dibuat menggunakan metode *Naïve Bayes* dengan nilai akurasi sebesar 75% tanpa seleksi fitur *Chi Square* dan 80% dengan seleksi fitur *Chi Square*. Pengembangan kali ini akan menggunakan *Naïve Bayes*, namun dengan normalisasi kata menggunakan *Levenshtein Distance*.

Penelitian menggunakan data training sebanyak 450 data dengan masing-masing kategori sebanyak 150 positif, 150 negatif dan 150 netral. Hasil pengujian pada 100 data uji dari penelitian sebelumnya menghasilkan tingkat akurasi sebesar 67.05% menggunakan *Levenshtein Distance* dan 63.83% tanpa *Levenhstein Distance* pada proses klasifikasi menggunakan *Naïve Bayes*.

Kata kunci—twitter, analisis sentimen, perbaikan kata tidak baku, levenshtein distance, naïve bayes

I. PENDAHULUAN

Penelitian yang dilakukan ini merupakan pengembangan dari penelitian terdahulu yang telah dilakukan oleh Dhitta Hananda dengan analisa sentimen pada Twitter menggunakan *Naïve Bayes Classifier* dan seleksi fitur *Chi Square* yang menunjukkan hasil tingkat akurasi sebesar 75% tanpa seleksi fitur *Chi Square* dan hasil tingkat akurasi sebesar 80% dengan seleksi fitur *Chi Square* [1]. Kekurangan dari penelitian yang sudah dilakukan oleh Dhitta Hananda adalah tidak adanya perbaikan kata tidak baku pada data *tweet* yang sudah diklasifikasikan menggunakan *Naïve Bayes Classifier* sehingga mengurangi hasil dari akurasi. Sehingga dibutuhkan suatu sistem perbaikan kata tidak baku yang dapat membantu memperbaiki dan menormalisasikan kata dalam *tweet* yang tidak memiliki arti menjadi kata baku sesuai dengan kamus Bahasa Indonesia secara otomatis.

Dari kekurangan dalam penelitian tersebut, terdapat sebuah metode yang dapat memperbaiki kata tidak baku dengan menggunakan metode *Levenshtein Distance*. Metode *Levenshtein Distance* ini dapat digunakan untuk memperbaiki dan menormalisasikan kata dengan kesalahan ejaan menjadi kata baku sehingga apabila ada kata yang tidak baku bisa ternormalisasi menjadi kata baku. Maka judul penelitian "Penerapan Normalisasi Kata Tidak Baku Menggunakan *Levenshtein Distance* pada Analisis Sentimen Layanan PT. KAI di Twitter" ini ditujukan untuk memperbaiki kata yang tidak baku dari komentar dan *feedback* yang diberikan oleh masyarakat. Perbaikan kata tidak baku ini digunakan untuk mengoreksi kesalahan ejaan yang diketik oleh pengguna yang dimana kata-kata tersebut tidak terdaftar dalam kamus Bahasa Indonesia. Dengan adanya penelitian ini diharapkan dapat memperbaiki tingkat akurasi dari penelitian yang sudah dilakukan sebelumnya.

Data yang digunakan pada penelitian kali ini berupa *tweet* masyarakat yang merupakan data dari hasil penelitian terdahulu yang sudah pernah melakukan pengujian dan *tweet* tahun 2019. Data *tweet* tersebut akan di normalisasi menggunakan metode *Levenshtein Distance* dan di klasifikasikan menggunakan metode *Naïve Bayes Classifier*. Pengujian akan dilakukan berdasarkan hasil dari nilai *accuracy*, dengan bertujuan untuk mengetahui tingkat kedekatan antara nilai prediksi dengan nilai aktual. Dan untuk mengetahui nilai akurasi normalisasi kata tidak baku menggunakan *Levenshtein Distance* dan yang tidak menggunakan *Levenshtein Distance*.

II. LANDASAN TEORI

A. *Preprocessing*

Pada *text mining* diperlukan beberapa tahapan untuk mengolah teks menjadi lebih terstruktur. Salah satu tahapan pada *text mining* adalah *text preprocessing*. Tahap ini adalah tahapan yang mana data disiapkan agar menjadi data yang siap untuk di analisis [2]. Pada data yang belum dilakukan *preprocessing* masih menjadi data mentah, yaitu data yang masih belum siap untuk dilakukan analisis, karena masih banyak mengandung kata-kata yang tidak memiliki arti dan kata yang belum terstruktur. Sehingga diperlukan *text preprocessing* untuk mengolah teks menjadi lebih terstruktur. Pada tahapan *preprocessing* meliputi proses *case folding*, *cleansing*, *tokenizing*, *stemming* dan *stopword*.

B. *Perbaikan Kata Tidak Baku*

Perbaikan kata tidak baku atau normalisasi bahasa adalah proses yang digunakan untuk mengubah kata-kata yang tidak baku menjadi kata baku sesuai dengan Kamus Besar Bahasa Indonesia (KBBI). Pada penelitian sebelumnya proses normalisasi meliputi [3] :

1. Merenggangkan tanda baca (*punctuation*) dan simbol selain *alphabet*
   Pada proses ini merenggangkan tanda baca dilakukan dengan cara memberikan jarak terhadap tanda baca dari kata-kata sebelumnya atau sesudahnya, dengan tujuan agar tanda baca dan simbol selain alphabet tidak menjadi satu dengan kata-kata pada saat proses tokenisasi.
2. Normalisasi Kata
   Pada proses normalisasi kata dilakukan dengan cara mengubah kata yang tidak baku menjadi baku sesuai dengan pedoman yang ada pada KBBI. Sebagai contoh ketika seseorang melakukan *Tweet* terkadang masih banyak yang tidak menggunakan kata baku, misalnya menuliskan kata "terima kasih" menjadi "makasi" (terima kasih→kata baku, makasi→kata tidak baku).
3. Menghilangkan huruf yang berulang
   Pada proses ini dilakukan dengan menghilangkan huruf yang berulang. Sebagai contoh ketika seseorang merasa senang atau kesal, terkadang mereka melakukan *Tweet* dengan mengulang huruf yang sama pada kata tersebut, seperti penulisan "Keeereeeennnn" maka akan dinormalisasi menjadi "Keren".
4. Menghilangkan *emoticon*
   Pada twitter, penggunaan *emoticon* sering dilakukan untuk mengekspresikan perasaan pengguna. Namun dalam penggunaan *emoticon* tersebut terkadang tidak sesuai dengan maksud dari apa yang sebenarnya. Sebagai contoh ketika pengguna melakukan *Tweet* terhadap respon film "film ini bagus banget :(" kata opini bagus namun digunakan *emoticon* :(. Sehingga dalam penelitian ini *emoticon* akan diabaikan saja atau dihapus.

TABEL I. CONTOH KAMUS_KATABAKU

Kata Tidak Baku	Kata Baku
abis	habis
ancur	hancur
gokil	gila
kentel	kental
lo	kamu
nonton	tonton
nyesel	sesal

C. *Levenshtein Distance*

Levenshtein distance atau yang biasa disebut dengan *edit distance* adalah suatu metode yang dapat digunakan untuk mengatasi terjadinya kesalahan ejaan. Kesalahan ejaan terjadi apabila kata yang diketik oleh pengguna tidak terdapat pada daftar kamus Bahasa Indonesia. Fungsi metode *Levenshtein Distance* yaitu untuk menghitung jarak kedekatan dari dua buah *string* melalui penambahan karakter, pengubahan karakter, dan penghapusan karakter hingga kedua *string* tersebut cocok.

Pada algoritme *Levenshtein Distance* terdapat 3 macam operasi utama yang dilakukan yaitu [4] :

1. Operasi Penambahan Karakter
   Operasi penambahan karakter yaitu operasi yang digunakan untuk menambahkan karakter ke dalam *string*. Contoh pada penulisan *string* 'kern' maka diubah menjadi *string* 'keren' dengan menambahkan karakter 'e'.
2. Operasi Pengubahan Karakter
   Operasi pengubahan karakter yaitu operasi yang digunakan untuk mengubah karakter dengan cara menukar sebuah karakter dengan karakter lain. Contoh pada penulisan *string* 'hidsp' diubah menjadi *string* 'hidup' dengan mengubah karakter 's' menjadi karakter 'u'.
3. Operasi Penghapusan Karakter
   Operasi penghapusan karakter yaitu operasi yang digunakan untuk menghapus suatu karakter pada *string.* Contoh pada penulisan *string* 'hebatt' di ubah menjadi *string* 'hebat' dengan menghilangkan karakter 't'.
   Persamaan yang digunakan untuk mencari *Distance* adalah sebagai berikut:

$$Dist_{a,b}(i,j) = Min\{$$
$$Dist_{a,b}((i,j-1)+1)$$
$$Dist_{a,b}((i-1,j)+1)$$
$$Dist_{a,b}((i-1,j-1)+1)\ (_{ai\neq bj})) \quad (1)$$

Keterangan :

a : *kata pertama*
b : *kata kedua*
i : *iterasi kata pertama*
j : *iterasi kata kedua*
Dist : *jarak*

Pada algoritma *Levenshtein Distance* dilakukan tahapan proses yang dimulai dari atas pojok kiri dari sebuah array dua dimensi yang telah diisi karakter *string* input dan *string* target. Selain itu juga terdapat nilai *cost* didalamnya. Nilai *cost* yang

berada pada bawah pojok kanan merupakan nilai *Edit Distance* yang menggambarkan jumlah perbedaan dari kedua *string*. Contoh perhitungan *Levenshtein Distance* dapat dilihat pada tabel dibawah ini :

TABEL II. MATRIKS PERHITUNGAN EDIT DISTANCE

		Y	**A**	**N**	**G**
	0	1	2	3	4
Y	1	0	1	2	3
A	2	1	0	1	2
N	3	2	1	0	**1**

Contoh pada perhitungan *Levenshtein Distance* diatas menggunakan 2 kata yang berbeda kemudian dilakukan perhitungan seperti pada Tabel 2.2. Nilai *Edit Distance* ditunjukkan pada Tabel 2.2 dengan warna merah. Hasil dari perhitungan *Levenshtein Distance* antara kata 'YAN' dan 'YANG' adalah 1. Pengecekan dimulai dari iterasi awal dari kedua kata kemudian dilakukan operasi pengubahan, penambahan dan penghapusan karakter. Pada contoh diatas hanya terdapat 1 penyisipan karakter, yaitu: karakter 'G' pada kata 'YAN' sehingga menjadi 'YANG'.

D. Naïve Bayes Classifier

Naïve Bayes Classifier merupakan salah satu metode yang populer untuk keperluan data mining karena penggunaannya yang mudah dan dalam pemroresan memiliki waktu yang cepat, mudah diimplementasikan dengan strukturnya yang cukup sederhana dan untuk tingkat efektifitasnya memiliki efektifitas yang tinggi [5]. Klasifikasi yang berdasar pada teorema bayes sangat cocok digunakan untuk dimensi masukan yang sangat besar. Klasifikasi *Naïve Bayes* juga memperlihatkan tingginya akurasi dan cepat ketika digunakan untuk dataset dengan jumlah besar. Secara umum proses dari klasifikasi *Naïve Bayes Classifier* dapat dilihat pada persamaan 2.2.

$$P(c_j|w_i) = \frac{P(c_j)P(w_i|c_j)}{P(w_i)} \quad (2)$$

Keterangan:
$P(c_j \mid w_i)$: Peluang kategori j ketika terdapat kemunculan kata *i*
$P(w_i \mid c_j)$: Peluang sebuah kata *i* masuk ke dalam kategori *j*
$P(c_j)$: Peluang kemunculan sebuah kategori *j*
$P(w_i)$: Peluang kemunculan sebuah kata

III. IMPLEMENTASI PENGUJIAN

A. Pengujian Performa Sistem

Pengujian performa sistem dilakukan dengan cara menghitung nilai dari *accuracy*. Selain menguji tingkat akurasi pengujian juga akan menghitung tingkat *precision* pada sistem. Pengujian tingkat akurasi dilakukan untuk mengetauhi pengaruh sebelum dan sesudah penggunaan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance*. Selain untuk mengetahui pengaruh perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* pada klasifikasi *Naïve Bayes*. Selanjutnya akan dilakukan beberapa pengujian yang melibatkan pengunaan data dengan komposisi yang berbeda. Seperti pengujian tingkat akurasi dengan penggunakan data *training* sebanyak 70% dan penggunaan data *training* dengan kategori yang tidak seimbang.

B. Pengujian Pengaruh Penggunaan Perbaikan Kata Tidak Baku dan Normalisasi Levenshtein Distance

Pengujian ini menjelaskan tentang pengujian untuk mengetahui pengaruh proses perbaikan kata tidak baku. Pada perbaikan kata tidak baku dilakukan dua proses. Proses pertama dilakukan dengan pencocokan kata tidak baku pada kamus_katabaku dan proses kedua adalah dengan normalisasi *Levenshtein Distance*. Pada pengujian ini dilakukan perbaikan kata tidak baku dengan seluruh proses, data latih dan data uji dilakukan variasi proses *preprocessing* dan dilakukan proses perbaikan kata tidak baku dan normalisasi *Levenshtein Distance*. Pada pengujian ini data yang digunakan merupakan data hasil dari penelitian terdahulu mengenai layanan PT. KAI di Twitter. *Dataset* yang digunakan sebanyak 550 *tweet* yang terdiri dari 450 untuk data *training* dan 100 untuk data uji. Data *training* terbagi menjadi tiga kategori yang masing-masing 150 positif, 150 negatif dan 150 netral. Hasil pengujian pengaruh penggunaan perbaikan kata tidak baku dan normalisasi *Levenshtein Distance* dapat dilihat pada Tabel III dan Tabel IV.

Pada Tabel III menunjukkan hasil pengujian menggunakan 100 data testing dari penelitian sebelumnya menghasilkan tingkat akurasi sebesar 63.83% sebelum dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* dan menunjukkan akurasi sebesar 67.05% setelah dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance*. Hal ini dikarenakan *Levenshtein Distance* telah memperbaiki kata-kata yang tidak baku dan tidak memiliki arti sesuai dengan Kamus Besar Bahasa Indonesia (KBBI) yang digunakan pada klasifikasi *Naïve Bayes*. Sehingga klasifikasi *Naïve Bayes* hanya melakukan perhitungan terhadap kata-kata yang sesuai dengan KBBI saja.

Hasil perbandingan tingkat presisi antara sebelum dan sesudah menggunakan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* menghasilkan tingkat presisi yang relatif lebih besar. Ini menunjukkan bahwa hasil klasifikasi setelah menggunakan perbaikan kata tidak baku dan normalisasi *Levenshtein Distance* memiliki tingkat kemiripan yang tinggi antara pelabelan secara sistem dengan pelabelan secara manual. Tingkat presisi pada pengujian ini menggunakan *Levenshtein Distance* dan menghasilkan nilai sebesar 0.63 setelah menggunakan *Levenshtein Distance*.

TABEL III. HASIL DARI PENGUJIAN DATA TESTING LAMA

	***Precision* Positif**	***Precision* Netral**	***Precision* Negatif**	***Accuracy***
Tanpa *Levenshtein Distance*	0.704545454 54545	0.777777777 77778	0.379310344 82759	63.83 %
Menggunakan *Levenshtein Distance*	0.756097560 97561	0.757575757 57576	0.384615384 61538	67.05 %

TABEL IV. HASIL DARI PENGUJIAN DATA TESTING BARU

	***Precision* Positif**	***Precision* Netral**	***Precision* Negatif**	***Accuracy***
Tanpa *Levenshtein Distance*	0.628571428 57143	0.296296296 2963	0.333333333 33333	41.84 %
Menggunakan *Levenshtein Distance*	0.6875	0.354838709 67742	0.365853658 53659	45.92 %

TABEL V. PENGUJIAN DATA 70% DAN DATA TESTING LAMA

	***Precision* Positif**	***Precision* Netral**	***Precision* Negatif**	***Accuracy***
Tanpa *Levenshtein Distance*	0.744186046 51163	0.733333333 33333	0.380952380 95238	67 %
Menggunakan *Levenshtein Distance*	0.775	0.710526315 78947	0.363636363 63636	67.02 %

TABEL VI. PENGUJIAN DATA 70% DAN DATA TESTING BARU

	***Precision* Positif**	***Precision* Netral**	***Precision* Negatif**	***Accuracy***
Tanpa *Levenshtein Distance*	0.615384615 38462	0.346153846 15385	0.384615384 61538	45.92 %
Menggunakan *Levenshtein Distance*	0.724137931 03448	0.384615384 61538	0.416666666 66667	48.98 %

TABEL VII. PENGUJIAN DATA KATEGORI TIDAK SEIMBANG DAN DATA TESTING LAMA

	***Precision* Positif**	***Precision* Netral**	***Precision* Negatif**	***Accuracy***
Tanpa *Levenshtein Distance*	0.513157894 73684	0.6	0.428571428 57143	51.07 %
Menggunakan *Levenshtein Distance*	0.593220338 98305	0.818181818 18182	0.421052631 57895	61.71 %

TABEL VIII. PENGUJIAN DATA KATEGORI TIDAK SEIMBANG DAN DATA TESTING BARU

	***Precision* Positif**	***Precision* Netral**	***Precision* Negatif**	***Accuracy***
Tanpa *Levenshtein Distance*	0.571428571 42857	0.368421052 63158	0.482758620 68966	51.02 %
Menggunakan *Levenshtein Distance*	0.586956521 73913	0.333333333 33333	0.411764705 88235	46.94 %

Pada Tabel IV menunjukkan hasil pengujian menggunakan 100 data testing terbaru tahun 2019 menghasilkan tingkat akurasi sebesar 41.84% sebelum dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* dan menunjukkan akurasi sebesar 45.92% setelah dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance*.

Hasil perbandingan tingkat presisi antara sebelum dan sesudah menggunakan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* menghasilkan tingkat presisi yang relatif lebih kecil. Tingkat presisi pada pengujian ini menghasilkan nilai dengan rata-rata sebesar 0.42 sebelum menggunakan Levenshtein Distance dan menghasilkan nilai sebesar 0.47 setelah menggunakan *Levenshtein Distance*.

C. Pengujian dengan Data Training Sebanyak 70%

Pengujian dengan menggunakan jumlah data *tweet* yang berbeda dimaksudkan untuk mengetahui efek dari penggunaan jumlah data *training* yang digunakan. Pengujian dilakukan menggunakan 70% data *tweet* dari total 450 data *training* yang ada. Sehingga total data *tweet* yang digunakan sebanyak 315 *tweet* dengan masing-masing kategori sebanyak 105 positif, 105 negatif dan 105 netral. Hasil pengujian pengaruh penggunaan perbaikan kata tidak baku dan normalisasi *Levenshtein Distance* dapat dilihat pada Tabel V dan Tabel VI.

Pada Tabel 6.4 menunjukan hasil tingkat akurasi dengan menggunakan 100 data testing dari penelitian sebelumnya sebesar 67% sebelum dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* dan menunjukkan akurasi sebesar 67.02% setelah dilakukan perbaikan kata tidak baku dan normalisasi *Levenshtein Distance*. Hal ini menunjukan jumlah data *training* mempengaruhi tingkat akurasi yang dihasilkan. *Naïve Bayes* juga merupakan metode pengklasifikasian yang berbasis data *learning*. Semakin sedikit data yang digunakan untuk *learning* maka tingkat akurasi yang dihasilkan juga semakin menurun.
Hasil pengujian tingkat presisi menunjukan kategori negatif memiliki tingkat akurasi paling rendah sedangkan kategori positif memiliki tingkat presisi yang tinggi. Tingkat presisi pada pengujian ini menghasilkan nilai dengan rata-rata sebesar 0.61 sebelum menggunakan *Levenshtein Distance* dan menghasilkan nilai sebesar 0.62 setelah menggunakan *Levenshtein Distance*.

Pada Tabel VI menunjukkan hasil pengujian menggunakan 100 data testing terbaru tahun 2019 menghasilkan tingkat akurasi sebesar 45.92% sebelum dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* dan menunjukkan akurasi sebesar 48.98% setelah dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance*.

Hasil perbandingan tingkat presisi antara sebelum dan sesudah menggunakan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* menghasilkan tingkat presisi yang relatif lebih kecil. Tingkat presisi pada pengujian ini menghasilkan nilai dengan rata-rata sebesar 0.45 sebelum menggunakan *Levenshtein Distance* dan menghasilkan nilai sebesar 0.51 setelah menggunakan *Levenshtein Distance*.

D. Pengujian dengan Jumlah Kategori Tidak Seimbang

Pengujian ini dimaksudkan untuk menguji adakah efek dari penggunan data *training* dengan kategori yang tidak seimbang. Pengujian akan dilakukan dengan menggunakan 450 data *tweet* dengan masing-masing kategori sebanyak 250 positif, 100 negatif dan 100 netral. Data uji yang digunakan yaitu sebesar 100 data *tweet* dari penelitian sebelumnya dan 100 data *tweet* tahun 2019. Hasil pengujian pengaruh penggunaan perbaikan kata tidak baku dan normalisasi

Levenshtein Distance dapat dilihat pada Tabel VII dan Tabel VIII.

Pada Tabel 6.6 menunjukkan hasil tingkat akurasi dengan menggunakan 450 data *tweet* tidak seimbang dan 100 data testing dari penelitian sebelumnya adalah sebesar 51.07% sebelum dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* dan menunjukkan akurasi sebesar 61.71% setelah dilakukan perbaikan kata tidak baku dan normalisasi *Levenshtein Distance*. Hasil pengujian kali ini mengalami penurunan dibandingkan dengan dua pengujian sebelumnya. Hal ini menunjukkan jumlah yang tidak seimbang turut mempengaruhi tingkat akurasi yang dihasilkan. Karena selain memperhitungkan probabilitas pada tiap kata, metode *Naïve Bayes* juga memperhitungkan probabilitas pada tiap-tiap kategori.

Hasil pengujian tingkat presisi menunjukan kategori negatif memiliki tingkat akurasi paling rendah sedangkan kategori netral memiliki tingkat presisi yang paling tinggi. Tingkat presisi pada pengujian ini menghasilkan nilai dengan rata-rata sebesar 0.513 sebelum menggunakan *Levenshtein Distance* dan menghasilkan nilai sebesar 0.611 setelah menggunakan *Levenshtein Distance*.

Pada Tabel 6.7 menunjukkan hasil pengujian menggunakan 100 data testing terbaru tahun 2019 menghasilkan tingkat akurasi sebesar 51.02% sebelum dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* dan menunjukkan akurasi sebesar 46.94% setelah dilakukan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance*.

Hasil dari perbandingan tingkat presisi antara sebelum dan sesudah menggunakan perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* menghasilkan tingkat presisi yang relatif sama. Tingkat presisi pada pengujian ini menghasilkan nilai dengan rata-rata sebesar 0.44 sebelum menggunakan *Levenshtein Distance* dan menghasilkan nilai sebesar 0.48 setelah menggunakan *Levenshtein Distance.*

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian dan pembahasan yang telah dilakukan pada penelitian mengenai penerapan normalisasi kata tidak baku menggunakan Levenshtein Distance pada analisis sentimen layanan PT. KAI di Twitter, dapat disimpulkan sebagai berikut :

1. Metode klasifikasi Naïve Bayes dengan perbaikan kata tidak baku menggunakan Levenshtein Distance dapat diterapkan pada proses analisis sentimen tentang layanan PT. KAI pada dokumen Twitter berbahasa Indonesia. Data latih dan data uji dilakukan proses preprocessing terlebih dahulu, kemudian setelah proses preprocessing terdapat tambahan perbaikan kata tidak baku menggunakan kamus_katabaku yang sudah ada.
2. Pada penggunaan perbaikan kata tidak baku dengan penambahan normalisasi Levenshtein Distance terhadap hasil klasifikasi analisis sentimen layanan PT. KAI pada dokumen Twitter berbahasa Indonesia memberikan pengaruh akurasi yang lebih baik, yaitu sebesar 67.05%, 67.02% dan 61.71% dari menggunakan 450 data tweet, 315 data tweet dan 450 data tweet tidak seimbang dengan data uji 100 tweet dari penelitian sebelumnya. Sedangkan, dengan menggunakan 100 tweet tahun 2019 sebagai data uji juga memberikan pengaruh akurasi yang lebih baik pada penggunaan perbaikan kata tidak baku dengan penambahan normalisasi Levenshtein Distance, yaitu sebesar 45.92%, 48.98% dan 46.94%.
3. Berdasarkan pengujian yang telah dilakukan, diperoleh hasil akurasi terbaik dari analisis sentimen tentang layanan PT. KAI pada dokumen Twitter berbahasa Indonesia menggunakan Naïve Bayes dengan perbaikan kata tidak baku dan normalisasi Levenshtein Distance adalah sebesar 67.05% dari menggunakan 450 data training dan 100 data testing dari penelitian sebelumnya. Sedangkan pengujian yang menggunakan 315 tweet dan 450 tweet kategori tidak seimbang sebagai data training dan 100 data testing menghasilkan tingkat akurasi yang lebih kecil. Hal ini menunjukkan bahwa keseimbangan jumlah kategori pada data training juga mempengaruhi bagi tingkat akurasinya.

B. Saran

Berdasarkan penelitian yang telah dilakukan, penulis menganggap bahwa penelitian yang sudah selesai dilakukan masih jauh dari kata sempurna, sehingga terdapat beberapa saran yang dapat digunakan untuk kepentingan penelitian selanjutnya yang berkaitan dengan *text mining* analisis sentimen atau penerapan *Naïve Bayes* dengan perbaikan kata tidak baku adalah sebagai berikut :

1. Pada perbaikan kata tidak baku penulis menggunakan kamus_katabaku yang sudah ada dan kata bahasa *modern* (*slang*) maupun kata singkatan yang sering muncul. Sehingga dapat ditambahkan kosakata lain pada kamus_katabaku.
2. Pada penelitian selanjutnya dapat ditambahkan fitur selain *bag-of-words*. Yaitu fitur *lexicon*, pembobotan emoticon, atau daftar kata positif dan negatif.
3. Pada penelitian ini untuk perbaikan kata tidak baku dan normalisasi menggunakan *Levenshtein Distance* masih membutuhkan waktu yang lama. Maka dari itu dapat dicoba menggunakan metode lain untuk mengurangi waktu prosesnya.

DAFTAR PUSTAKA

[1] Hananda, Dhitta., Rozi, Imam Fahrur & Apriyani, Meyti Eka., *Implementasi Analisa Sentimen pada Twitter Layanan PT. KAI Menggunakan Metode Naïve Bayes dengan Seleksi Fitur Chi Square*. Malang: Teknik Informatika Politeknik Negeri Malang, 2018.

[2] Hadna, N. et al., Studi Literatur tentang perbandingan metode proses analisis sentimen di twitter. Fakultas Teknik, Universitas Gadjah Mada, 2016.

[3] Buntoro, G. A., Adji, T. B., & Purnamasari, A. E., *Sentiment Analysis Twitter dengan Kombinasi Lexicon Based dan Double Propagation*. CTIEE. Halaman 39-43, 2014.

[4] Adriyani, N. M., Santiyasa, I. W. & Muliantara, A., *Implementasi Algoritma Levenshtein Distance dan Metode Impiris untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Doukem Berbahasa Indonesia*. Fakultas Ilmu Komputer, Universitas Udayana, 2012.

[5] Asrofi, G. (2015). Analisis Sentimen Calon Presiden Indonesia 2014 dengan Lima Class Attribute. Universitas Gadjah Mada. Available at:

http://etd.repository.ugm.ac.id/index.php?mod=penelitian_detail&sub=PenelitianDetail&act=view&typ=html&buku_id=80122&obyek_id=4

[Diakses 2 Desember 2018]

[6] Rohmah, Maya Shoburu., Astininingrum, Mungki & Saputra, Pramana Yoga., *Implementasi NLP dengan Konversi Kata pada Sistem Chatbot Konsultasi Laktasi*. Malang: Teknik Informatika Politeknik Negeri Malang, 2018.