Pengaruh Kombinasi dan Urutan *Pre-Processing* pada *Tweets* Bahasa Indonesia

Sheila Shevira[a1], I Made Agus Dwi Suarjaya[a2], Putu Wira Buana [b3]
[1,2,3] Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana
e-mail: [1]shevira@student.unud.ac.id, [2]agussuarjaya@it.unud.ac.id, [3]wbhuana@it.unud.ac.id

Abstrak

Twitter merupakan jaringan microblog online yang dijadikan gaya hidup baru di kalangan masyarakat sebagai wadah pengganti untuk mencari dan menyebarkan informasi, sebagai tempat mencurahkan perasaan, ataupun menjalankan bisnis, dengan cara menuliskan tweet. Permasalahannya adalah tweet yang dituliskan mayoritas oleh remaja berumur 18-24 tahun, sehingga kata-kata yang dituliskan masih banyak mengandung karakter pengganggu, ejaan, kata gaul, atau kata yang bersifat non-baku. Data yang tidak bersih dan akurat akan berdampak buruk bagi hasil analisis. Pre-processing data dalam hal ini berperan penting untuk memperbaiki data agar menjadi lebih bersih dan akurat sebelum diproses. Penelitian ini fokus membahas mengenai beberapa skenario kombinasi pre-processing, serta dengan mengubah urutan proses cleaning, normalisasi, stemming, dan stop-word, untuk mendapatkan akurasi paling baik dan meningkatkan performa dalam klasifikasi. Hasil testing pada tweet menunjukkan akurasi tertinggi ada pada data yang melewati tahapan penuh pre-processing data dengan urutan kombinasi pre-processing adalah menaruh proses normalisasi sebelum melakukan proses stemming, yaitu sebesar 89.2%.

Kata kunci: *Normalisasi, Pre-Processing Data, Stemming, Stop-word, Twitter*

Abstract

Twitter is an online microblog network that is used as a new lifestyle among the community as a substitute for finding and sharing information, as a place to express feelings, or doing a business, by writing tweets. The problem is that the majority of tweets are written by teenagers aged 18-24 years, so the words written still contain a lot of distracting characters, spelling, slang, or non-standard words. Data that is not clean and accurate will have a bad impact on the results of the analysis. Pre-processing data in this case plays an important role in improving the data to be cleaner and more accurate before being analyzed. This study focuses on discussing several scenarios of pre-processing combinations, and changing the order of the cleaning, normalizing, stemming, and stop-word processes, to get the best accuracy and improve performance in classification. The results of testing on tweets show that the highest accuracy is in data that has passed the full stages of pre-processing data with the pre-processing combination order placing the normalization process before carrying out the stemming process, which is 89.2%.

Keywords: *Normalization, Pre-Processing Data, Stemming, Stop-word, Twitter*

1. Pendahuluan

Twitter merupakan jaringan *microblog online* yang sering dimanfaatkan masyarakat untuk menuliskan segala sesuatu yang berkaitan dengan pemikiran, suasana hati, aktivitas, komunikasi, kehidupan sosial, ataupun berbagi informasi berita, karena penggunanya dapat menuliskan pesan secara singkat, padat, dan jelas dibawah 280 karakter [1]. Banyak masyarakat saat ini menuliskan *tweet* di Twitter untuk menunjukkan perasaannya secara *real-time*, baik berupa *tweet* yang bersifat positif maupun negatif. *Tweet* yang tersebar di internet ini dapat digunakan sebagai data dalam melakukan analisis terhadap berbagai hal.

Data merupakan aspek penting bagi setiap individu, organisasi, maupun perusahaan, karena berperan dalam banyak hal, seperti pengambilan keputusan, sebagai dasar

perencanaan, ataupun sebagai bahan evaluasi [2]. Kualitas data dalam *data mining* berpengaruh signifikan pada tingginya kinerja model dan juga hasil analisis. Kualitas data yang buruk dapat berdampak negatif bagi produktivitas individu/organisasi/perusahaan sehingga capaian hasil yang diinginkan menjadi tidak maksimal.

Tweet yang dituliskan setiap orang pada media sosial Twitter, dapat dikatakan sebagai salah satu data yang bersifat kurang baik, karena mayoritas pengguna Twitter adalah di kalangan masyarakat remaja, yakni berumur 18-24 tahun [3]. Hal ini membuat *tweet* yang tersebar di dunia maya lebih banyak menggunakan kata yang memiliki ejaan/singkatan ataupun kata-kata gaul. Tahap *pre-processing* dalam hal ini penting dilakukan untuk membersihkan dan memperbaiki struktur pada *tweet* agar data menjadi lebih seragam dan akurat ketika dianalisis.

Pre-processing adalah tahap untuk melakukan transformasi data agar sesuai dengan format seharusnya dan dapat diproses. Tahapan *pre-processing* ada beragam tergantung pada kebutuhan, namun secara umum diantaranya, yaitu *cleaning tweet*, *lowering case*, normalisasi, *stop-word*, *stemming*, dan *tokenizing*. *Cleaning tweet* dilakukan agar karakter yang bersifat *noise* dapat dihilangkan [4]. *Lowering case* merupakan proses untuk mengubah bentuk data menjadi huruf kecil seluruhnya [5]. Normalisasi *tweet* merupakan proses untuk mengubah *tweet* yang mengandung ejaan dan kata *slang* untuk mempermudah proses analisis [6]. Kata *slang* yang dimaksudkan adalah kata yang sulit dimengerti, kata yang memiliki makna beda dari makna aslinya, kata yang bersifat non-baku, ataupun kata gaul yang digunakan remaja modern [7]. Proses normalisasi bertujuan agar pembaca dapat memahami arti kata yang dimaksudkan secara jelas, begitu juga dengan sistem ketika melakukan analisis klasifikasi. *Stemming* merupakan proses dalam mentransformasi kata menjadi kata dasar dengan menghilangkan imbuhannya, sehingga bisa meminimalkan dimensi kata yang berbeda bentuk namun dengan makna yang sama [8]. *Stop-word* dilakukan untuk meminimalkan jumlah kata dengan nilai informasi rendah sehingga proses klasifikasi lebih cepat dan akurat [9]. *Tokenizing* adalah proses memecah kalimat per-kata/token [10].

Penelitian terdahulu yang sejenis oleh Danny Sebastian dan Kristian Adi Nugraha tahun 2019 adalah melakukan pengembangan dataset kata-kata singkatan bahasa Indonesia, yang kemudian dapat digunakan untuk menormalkan kata-kata menggunakan *Crowdsourcing*. Akurasi yang didapatkan dari penelitian ini sekitar 90,85%.[11]

Dwi Wahyudi, dkk melakukan penelitian di tahun 2017 untuk membandingkan 2 algoritma *stemming* pada *task* bahasa Indonesia. Fokus penelitian yaitu ada pada proses *stemming* yang membandingkan algoritma Nazief & Adriani dengan algoritma Porter untuk menentukan algoritma yang lebih akurat. Hasil menunjukkan bahwa algoritma Nazief & Adriani lebih unggul daripada algoritma Porter, yaitu dengan akurasi 95,26% dan 79,13%. [12]

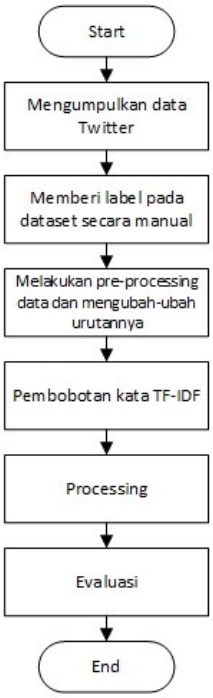
Penelitian terkait *pre-processing* telah dilakukan oleh Siti Khomsah & Agus Sasmito Aribowo tahun 2020. Berbagai macam model *pre-processing* dilakukan pada data komentar YouTube, lalu dibandingkan untuk mendapatkan akurasi terbaiknya. Hasil menunjukkan bahwa *pre-processing* dengan melakukan *cleaning* data, *stop-words*, dan konversi kata gaul berdasarkan kamus bahasa Indonesia menaikkan akurasi sebanayk 3,5% pada kata *unigram* [13].

Penelitian lain telah dilakukan juga oleh Riri Riyaddulloh dan Ade Romadhony tahun 2021. Penelitian ini melakukan normalisasi pada *tweet* yang dituliskan akun resmi beberapa *merk gadget* menggunakan model *word2vec*. Hasil pengujian yang didapatkan, yaitu adanya peningkatan akurasi untuk *tweet* yang mengalami normalisasi dibandingkan dengan *tweet* yang tidak mengalami normalisasi, yaitu 91% berbanding 88%. [7]

Penelitian kali ini memfokuskan pada *pre-processing* data dengan mencoba beberapa skenario urutan *pre-processing* pada proses normalisasi, *stop-word*, dan *stemming*. Dilakukannya penelitian ini bertujuan agar dapat melihat pengaruh yang dihasilkan dari urutan *pre-processing* data dan juga kombinasi proses di dalamnya sebelum diekstraksi fitur dan melakukan analisis.

2. Metode Penelitian

Analisis pengaruh kombinasi dan urutan tahap *pre-processing* pada *tweet* bahasa Indonesia memiliki metode penelitian yang dipaparkan ke dalam alur sebagai berikut.



Gambar 1. Alur penelitian

Alur yang digunakan dari penelitian ini terbagi dalam beberapa tahapan. Pengumpulan data dilakukan pada Twitter menggunakan Twitter API, dan diberi label secara manual. Tahap *pre-processing* data dilakukan beberapa kali, yaitu dengan kombinasi dan urutan proses *cleaning*, normalisasi, *stop-word*, dan *stemming* yang berbeda-beda. *Dataset* yang sudah di-*cleansing* diekstrak dengan menggunakan TF-IDF untuk mentransformasikan tulisan ke bentuk angka, sebelum diuji performa modelnya menggunakan Naïve Bayes, sehingga didapatkan hasil akurasi terbaik.

3. Kajian Pustaka

Penelitian ini dilakukan berdasarkan beberapa teori penunjang yang dijabarkan dalam kajian pustaka sebagai berikut.

3.1. Twitter dan Twitter API

Twitter adalah jaringan microblog *online* yang mengizinkan penggunanya untuk menulis cuitan atau pesan singkat di bawah 280 karakter [1]. Manfaat Twitter sangat beragam mulai dari bertukar indormasi, melatih kreativitas menulis, hingga menjalankan bisnis [14]. Twitter memiliki API tersendiri yang dibangun untuk mempermudah *developer* dalam mengambil data untuk diolah, dengan cara *login* melalui akun *developer* hingga mendapatkan akses API berupa `api_key, api_secret_key`, `access_token`, dan `access_token_secret`.

3.2. Python

Python merupakan bahasa pemrograman yang memiliki banyak *library* dan digunakan oleh *data scientists* dan *machine learning engineers* untuk mengembangkan model serta aplikasi yang berhubungan dengan *data science*. *Library* Python yang digunakan dijabarkan sebagai berikut.

a. Snscrape, untuk mengumpulkan data Twitter dengan Twitter API
b. Regex, untuk proses *cleaning*, dan normalisasi.
c. NLTK, untuk proses *stop-word*.
d. Sastrawi, untuk membantu proses *stemming*
e. Sklearn, untuk klasifikasi

3.3. Text Pre-Processing

Text *pre-processing* terdiri dari beberapa proses di dalamnya yang terbagi sebagai berikut.

a. *Cleaning Text*, untuk membersihkan *tweet* dari simbol, angka, URL, *hashtag*. Contohnya menghilangkan tanda koma (,) pada kalimat "Pengen menyerah, capek dgn smuanya" sehingga menjadi "Pengen menyerah capek dgn smuanya".
b. *Lowering Case*, untuk menyeragamkan ke dalam huruf kecil. Contohnya kata "Pengen" menjadi "pengen".
c. Normalisasi, untuk mengubah kata non-baku menjadi baku. Contohnya kata "pengen" menjadi "ingin", kata "dgn" menjadi "dengan", kata "smuanya" menjadi "semuanya".
d. *Stop-word removal*, untuk menghilangkan kata yang nilai informasinya rendah. Contohnya kata "ada", "di", "dan", "dengan".
e. *Stemming*, untuk mengubah kata menjadi bentuk dasarnya. Contohnya kata "semuanya" menjadi "semua", kata "perasaan" menjadi "rasa".
f. *Tokenization*, untuk memecah kalimat per-*token* atau kata.

Penelitian ini menggunakan 20 skenario kombinasi dan urutan *pre-processing*, yaitu: melakukan *cleaning text* saja, normalisasi saja, *stopword* saja, *stemming* saja, *cleaning text* dan normalisasi, *cleaning text* dan *stopword*, *cleaning text* dan *stemming*, normalisasi dan *stopword*, normalisasi dan *stemming*, *stopword* dan *stemming*, *cleaning text*-normalisasi-*stopword*, *cleaning text*-normalisasi-*stemming*, normalisasi-*stopword-stemming*, *full pre-processing* 1 dengan urutan (*cleaning text-lowercase*-normalisasi-*stopword-stemming-tokenizing*), *full pre-processing* 2 (*cleaning text-lowercase*-normalisasi-*stemming-stopword-tokenizing*), *full pre-processing* 3 (*cleaning text-lowercase-stopword-stemming*-normalisasi-*tokenizing*), *full pre-processing* 4 (*cleaning text-lowercase-stopword*-normalisasi-*stemming-tokenizing*), *full pre-processing* 5 (*cleaning text-lowercase-stemming*-normalisasi-*stopword-tokenizing*), *full pre-processing* 6 (*cleaning text-lowercase-stemming-stopword*-normalisasi-*tokenizing*).

3.4. TF-IDF

Dataset yang masih berbentuk tulisan, sebelum melewati pemrosesan harus diekstrak terlebih dahulu dengan TF-IDF. TF-IDF digunakan sebagai salah satu bagian *information retrieval system* dalam mengubah data non-terstruktur menjadi bentuk yang dapat memenuhi kebutuhan informasi. Tujuannya agar tulisan dapat direpresentasikan ke dalam bentuk angka sebelum diolah dengan Naïve Bayes. Ekstraksi tulisan dapat dikalkulasikan menggunakan rumus berikut.

$$w_{ij} = tf_{ij}\ x \log\left(D/df_j\right) \tag{1}$$

Menentukan bobot setiap kata *(w)* dilakukan dengan mengalikan banyaknya kata i dalam suatu dokumen j (*tf*) dan *idf*, yang mana (*idf*) didapat dari logaritma pembagian jumlah seluruh dokumen (*D*) dengan banyaknya dokumen yang mengandung kata j (*df*).

3.5. Evaluasi Performa

Pengukuran *performance* kinerja dari beberapa kombinasi dan urutan *pre-processing* dilakukan dengan menghitung keakuratan dengan rumus berikut.

$$Akurasi = \frac{Total\ Benar}{Jumlah\ Keseluruhan}\ x\ 100\% \tag{2}$$

4. Hasil dan Pembahasan

Hasil yang didapat dari penelitian ini akan dibahas dengan pemaparan berdasarkan gambaran umum sistem yang ada.

4.1. Dataset

Dataset didapat dengan mengumpulkan data Twitter menggunakan *library* Snscrape dan disimpan ke Mongo *database*. *Keyword* yang digunakan berkaitan dengan kata-kata yang mengindikasikan gangguan mental, yaitu "capek", "stres", "depresi", "mau mati", dan "tidak ada

yang peduli". Data *tweet* yang sudah dikumpulkan selanjutnya diberi label secara manual sesuai kebutuhan penelitian. Beberapa *tweet* yang diambil dari Twitter ada pada Tabel 1.

Tabel 1. *Dataset* Twitter

Full Tweet	Label
Maaf mngecewakan km, udh cape bgt..	0
@dyonisuss2_ Yallah capek ketawa	1
Rasanya mau mati	0
hadiah paling susah menurutku itu doa yg tulus jadi bersyukur sekali bisa dapat teman2 yang tidak ada hentinya doakan yg baik2. terima kasih jg untuk mama papa yg ajar caca untuk peduli sekitar dengan hal paling simple : selalu doakan sesama manusia sebaik mungkin.	1
@bertanyarl Jujur aku kalau lagi capek emang kepikiran itu cuma kalau entar aku udah mati gimana perasaan orang tua aku? :) Mereka pasti kecewa banget, anak yang mereka besarkan sepenuh hati bukannya berjuang untuk hidup malah lebih milih nyerah. Kena dosa juga karena belum waktunya dipanggil	0
Cinta sejati akan dibawa sampai mati.	1
SUMPAH GUE SENENG BANGET GILA HAHAHAA	1
Ini hari paling buruk bikin hilang minat dan lelah nangis :)))	0
gue jadi manusia bodoh bgt ! jadi paling bego.tolol.banget. gue mending gak lahir kykny	1
Guee bersyukur bgt ada Bangtan yang jadi sumber kebahagiaan, penerang, dan semangat untuk gue…	0
Ini jg ngikut jd cowok mamba, gila ga kuat gua knp ganteng banget sih	1

Data *tweet* yang dikumpulkan berjumlah 2.400 data dan dilabelkan secara manual. *Dataset* sebanyak 2.400 terbagi atas 1.151 (48%) diberi label “0” untuk merepresentasikan *tweet* negatif dan 1.249 (52%) diberi label “1” untuk merepresentasian *tweet* positif.

4.2. Pre-Processing Data

Tahap ini dilakukan untuk membersihkan data *tweet* sebelum dilakukan analisis agar data menjadi lebih akurat. Berikut pada Tabel 2. adalah contoh hasil tahap *pre-processing* data.

Tabel 2. Hasil *pre-processing* data

Proses	Tweet Sebelum Proses	Tweet Setelah Proses
Cleaning Text	Maaf mngecewakan km, udh cape bgt..	Maaf mngecewakan km udh cape bgt
Lowering Case	Maaf mngecewakan km udh cape bgt	maaf mngecewakan km udh cape bgt
Normalisasi	maaf mngecewakan km udh cape bgt	maaf mengecewakan kamu sudah capek banget
Stopword Removal	maaf mengecewakan kamu sudah capek banget	maaf mengecewakan capek banget
Stemming	maaf mengecewakan capek banget	maaf kecewa capek banget
Tokenizing	maaf kecewa capek banget	“maaf”, “kecewa”, “capek”, “banget”

Data *tweet* melewati tahap *cleaning text* untuk menghilangkan karakter *noise* dengan menghapus angka, simbol, URL, *hashtag*, lalu diseragamkan seluruhnya ke dalam huruf kecil pada tahap *lowering case*. *Tweet* yang sudah bersih melewati tahap normalisasi dengan mengubah kata gaul, ejaan, ataupun kata non-baku menjadi bentuk sebenarnya. Tahap normalisasi dilakukan berdasarkan kamus *slang* yang dibuat manual yang terdiri dari 250 kata. Berdasarkan Tabel 2. dapat diketahui bahwa normalisasi dilakukan pada pada “mngecewakan”

menjadi “mengecewakan”, “km” menjadi “kamu”, “udh” menjadi “sudah”, “cape” menjadi “capek”, dan “bgt” menjadi “banget”. *Tweet* yang sudah melewati tahap normalisasi, selanjutnya dipilih kata-kata yang memiliki nilai informasi rendah untuk dihilangkan. Tahap *stop-word removal* dilakukan berdasarkan kamus *stop-word* yang didapat dari jurnal Fadillah Z Tala, yang terdiri dari 750 kata [15]. Contohnya, yaitu menghapuskan kata ganti orang “kamu”, dan kata hubung “sudah”. Data *tweet* yang sudah disaring kemudian melewati tahap *stemming* untuk mengubah kata demi kata menjadi ke bentuk dasar, yaitu kata “mengecewakan” yang dihilangkan *prefix*, *infix*, serta *suffix-nya* sehingga menjadi “kecewa”. Terakhir yaitu memecah kalimat menjadi *token* pada tahap *tokenizing*.

4.3. Pembobotan TF-IDF

Dataset hasil *cleansing* didapatkan berupa atribut tulisan, sehingga perlu dilakukan pembobotan agar *dataset* dapat diproses dengan algoritma Naïve Bayes. Hasil pembobotan adalah berupa matriks numerik. Contoh kalkulasi TF-IDF pada 3 buah dokumen (D) sebagai berikut.

Tweet Clean:
maaf kecewa capek banget (negatif)
yallah capek ketawa (positif)
rasa mau mati (negatif)

Tabel 3. Matriks Kalkulasi TF-IDF

Kata	tf			df	D/df	IDF (log D/df)	W (TF-IDF)		
	D1	D2	D3				D1	D2	D3
maaf	1	0	0	1	3	0,477121	0,477121	0	0
kecewa	1	0	0	1	3	0,477121	0,477121	0	0
capek	1	1	0	2	1,5	0,176091	0,176091	0,176091	0
banget	1	0	0	1	3	0,477121	0,477121	0	0
yallah	0	1	0	1	3	0,477121	0	0,477121	0
ketawa	0	1	0	1	3	0,477121	0	0,477121	0
rasa	0	0	1	1	3	0,477121	0	0	0,477121
mau	0	0	1	1	3	0,477121	0	0	0,477121
mati	0	0	1	1	3	0,477121	0	0	0,477121

4.4. Evaluasi

Evaluasi performa dari kombinasi dan urutan *pre-processing* dilakukan dengan algoritma Naïve Bayes. Pembagian *dataset* pada data hasil klasifikasi diakukan dengan menggunakan perbandingan persentase data uji berbanding data latih, yaitu 90:10.

Tabel 4. Evaluasi performa *pre-processing* data

Skn	Urutan	Akurasi
1	*Cleaning text* saja	77.1%
2	Normalisasi saja	79.6%
3	*Stop-word removal* saja	78.3%
4	*Stemming* saja	78.7%
5	*Cleaning text* dan Normalisasi	80.7%
6	*Cleaning text* dan *Stop-word removal*	79.6%
7	*Cleaning text* dan *Stemming*	79.6%
8	Normalisasi dan *Stop-word removal*	84.1%
9	Normalisasi dan *Stemming*	85.7%
10	*Stop-word removal* dan *Stemming*	82.6%
11	*Cleaning text*, Normalisasi, dan *Stop-word removal*	85.5%
12	*Cleaning text,* Normalisasi, dan *Stemming*	85.7%
13	*Cleaning text, Stop-word removal*, dan *Stemming*	79.7%
14	Normalisasi, *Stop-word removal*, dan *Stemming*	85.7%

Skn	Urutan	Akurasi
15	*Full 1(Cleaning-Lowercase*-Normalisasi-*Stopword-Stemming-Tokenizing)*	89.2%
16	*Full 2(Cleaning-Lowercase*-Normalisasi-*Stemming-Stopword-Tokenizing)*	89.2%
17	*Full 3(Cleaning-Lowercase-Stopword-Stemming*-Normalisasi-*Tokenizing)*	86.8%
18	*Full 4(Cleaning-Lowercase-Stopword*-Normalisasi-*Stemming-Tokenizing)*	88.5%
19	*Full 5(Cleaning-Lowercase-Stemming*-Normalisasi-*Stopword-Tokenizing)*	87.1%
20	*Full 6(Cleaning-Lowercase-Stemming-Stopword*-Normalisasi-*Tokenizing)*	87.3%

Hasil akurasi dengan kinerja sistem tertinggi diperoleh oleh skenarip 15, dan 16 yang melakukan *pre-processing* secara keseluruhan, yaitu sebesar 89,2%. Urutan tertinggi dalam tahap *full pre-processing* ini didapatkan oleh urutan yang melakukan proses normalisasi lebih dulu sebelum melakukan *stemming*, sedangkan akurasi tertinggi pada tahap *full pre-processing* dengan urutan normalisasi setelah *stemming* adalah 87,3%. Hal ini dikarenakan ketika proses normalisasi dilakukan sebelum *stemming*, maka kata non-baku sudah menjadi baku sebelum diubah ke bentuk dasar, namun apabila normalisasi dilakukan setelah *stemming*, maka akan ada beberapa kata yang tidak terdeteksi di proses *stemming* karena belum menjadi ke bentuk sebenarnya dalam kamus bahasa Indonesia, sehingga imbuhan tidak dapat dihilangkan, dan data yang dihasilkan menjadi kurang akurat serta tidak dapat dianalisis secara maksimal dalam tahap *processing*. Hasil ini juga menunjukkan bahwa urutan proses pada tahap *pre-processing* data berpengaruh pada akurasi dan kinerja model dalam analisis klasifikasi, terutama dalam mengatur tata letak dilakukannya proses normalisasi dan *stemming*.

Lima (5) urutan tertinggi berikutnya untuk tahapan *pre-processing* secara tidak lengkap, secara berurutan dijuarai oleh: skenario 9 (akurasi sebesar 85,7%) dengan kombinasi hanya melakukan normalisasi dan *stemming*; skenario 12 (akurasi sebesar 85,7%) dengan kombinasi *cleaning text,* normalisasi, dan *stemming*; skenario 14 (akurasi sebesar 85,7%) dengan kombinasi normalisasi, *stop-word removal*, dan *stemming*; skenario 11 (akurasi sebesar 85,5%) dengan kombinasi *cleaning text*, normalisasi, dan *stop-word removal*; skenario 8 (akurasi sebesar 84,1%) dengan kominasi hanya melakukan normalisasi dan *stop-word removal*. Urutan kombinasi *pre-processing* dengan akurasi terendah ada pada proses yang melakukan *cleaning text* saja, yaitu sebesar 77,1%. Berdasarkan hasil ini, kelima kombinasi *pre-processing* dengan urutan teratas diantaranya adalah sama-sama melakukan proses normalisasi, sedangkan untuk urutan dengan akurasi terendah tidak melakukan proses normalisasi. Melihat dari percobaan pada skenario 2 dengan hanya melakukan normalisasi saja, akurasi yang didapat juga sudah cukup tinggi dibandingkan dengan tanpa melakukan normalisasi, yaitu sebesar 79,6%. Berdasarkan hasil ini juga, hal terbesar yang mungkin menyebabkan tinggi dan rendahnya akurasi adalah karena adanya pengaruh dari proses normalisasi. Dalam proses normalisasi, kata-kata yang disingkat pada data, ejaan, bahasa gaul, atau kata apapun yang berbentuk non-baku, seperti kata “km”, “udah”, “utk”, “onlen”, “cpt”, “smg”, dan lain sebagainya, akan diubah menjadi bentuk yang asli yang memiliki makna sama. Hasil normalisasi akan berdampak pada data *tweet* yang menjadi mudah dipahami dan dapat diklasifikasikan dengan sesuai oleh sistem. Lain halnya dengan data yang tidak mengalami proses normalisasi, maka sistem akan kebingungan dalam mengecek kata demi kata sehingga hasil klasifikasi menyatakan *false*.

5. Kesimpulan

Kesimpulan yang bisa diperoleh dari analisa tahap *pre-processing* pada *tweet* bahasa Indonesia adalah kombinasi dan urutan proses pada tahap *pre-processing*, dalam hal ini mempengaruhi akurasi dan kinerja model yang digunakan dalam analisis klasifikasi. Berdasarkan 20 skenario *pre-processing* data dengan mengubah kombinasi dan urutan proses *cleaning text*, normalisasi, *stemming*, dan *stop-word*, didapat akurasi lebih tinggi pada *pre-processing* yang dilakukan secara keseluruhan (*full pre-processing*) dengan urutan melakukan normalisasi terlebih dahulu sebelum melakukan *stemming*, yaitu sebesar 89,2%. Pengaruh terbesar dari tingginya akurasi yang didapat, ada pada proses normalisasi sebagai kunci utamanya, karena apabila kata tidak dinormalisasikan terlebih dahulu, maka sistem akan sulit mendeteksi kata dan mengakibatkan adanya klasifikasi yang salah. Namun, hal ini tidak berarti bahwa proses *cleaning*, *stemming*, dan *stop-word* tidak berpengaruh pada kinerja analisis, meskipun akurasi yang didapat dari kombinasi tersebut lebih rendah. Ketiga proses tersebut cukup untuk menaikkan beberapa persen akurasi dari kinerja sistem. Hal ini dibuktikan dengan hasil akurasi tertinggi ada pada tahap *pre-processing* yang dilakukan dengan menggunakan

seluruh proses didalamnya, meskipun harus dengan urutan dimana normalisasi dilakukan lebih dulu daripada *stemming*.

Referensi

[1] S. Bhatt, “Apa itu Twitter?,” 2022. https://www.experthoot.com/id/cara-dm-di-twitter/.

[2] L. N. Azizah, “Pengertian Data: Fungsi, Manfaat, Jenis, dan Contohnya,” *Gramedia Blog*, 2022. https://www.gramedia.com/literasi/pengertian-data/#:~:text=a.,-Sebagai Suatu Acuan&text=Manfaat dan juga fungsi data,kegiatan tertentu yang kita inginkan.

[3] Adam, “Demografi Pengguna Twitter di Indonesia Paling Banyak Pria daripada Perempuan,” *itworks.id*, 2019. https://www.itworks.id/19408/demografi-pengguna-twitter-di-indonesia-paling-banyak-pria-daripada-perempuan.html.

[4] D. Sebastian, “Implementasi Algoritma K-Nearest Neighbor untuk Melakukan Klasifikasi Produk dari beberapa E-marketplace,” *J. Tek. Inform. dan Sist. Inf.*, vol. 5, no. 1, pp. 51–61, 2019, doi: 10.28932/jutisi.v5i1.1581.

[5] P. A. Sumitro, Rasiban, D. I. Mulyana, and W. Saputro, “Analisis Sentimen Terhadap Vaksin Covid-19 di Indonesia pada Twitter Menggunakan Metode Lexicon Based,” *J-ICOM - J. Inform. dan Teknol. Komput.*, vol. 2, no. 2, pp. 50–56, 2021, doi: 10.33059/j-icom.v2i2.4009.

[6] D. Darwis, E. S. Pratiwi, and A. F. O. Pasaribu, “Penerapan Algoritma Svm Untuk Analisis Sentimen Pada Data Twitter Komisi Pemberantasan Korupsi Republik Indonesia,” *Edutic - Sci. J. Informatics Educ.*, vol. 7, no. 1, pp. 1–11, 2020, doi: 10.21107/edutic.v7i1.8779.

[7] R. Riyaddulloh and A. Romadhony, “Normalisasi Teks Bahasa Indonesia Berbasis Kamus Slang Studi Kasus: Tweet Produk Gadget Pada Twitter,” *eProceedings Eng.*, vol. 8, no. 4, pp. 4216–4228, 2021, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15246/14969.

[8] Arifin Kurniawan, Indriati Indriati, and Sigit Adinugroho, “Analisis Sentimen Opini Film Menggunakan Metode Naïve Bayes dan Lexicon Based Features,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 9, pp. 8335–8342, 2019.

[9] S. Almouzini, M. Khemakhem, and A. Alageel, “Detecting Arabic Depressed Users from Twitter Data,” *Procedia Comput. Sci.*, vol. 163, pp. 257–265, 2019, doi: 10.1016/j.procs.2019.12.107.

[10] B. Nurfadhila, “Analisis Sentimen Untuk Mengukur Tingkat Indikasi Depresi Pada Twitter Menggunakan Text Mining,” no. 1, 2018.

[11] D. Sebastian and K. A. Nugraha, “Text normalization for Indonesian abbreviated word using crowdsourcing method,” *2019 Int. Conf. Inf. Commun. Technol. ICOIACT 2019*, pp. 529–532, 2019, doi: 10.1109/ICOIACT46704.2019.8938463.

[12] D. Wahyudi, T. Susyanto, and D. Nugroho, “Implementasi Dan Analisis Algoritma Stemming Nazief & Adriani Dan Porter Pada Dokumen Berbahasa Indonesia,” *J. Ilm. SINUS*, vol. 15, no. 2, pp. 49–56, 2017, doi: 10.30646/sinus.v15i2.305.

[13] S. Khomsah and Agus Sasmito Aribowo, “Model Text-Preprocessing Komentar Youtube Dalam Bahasa Indonesia,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 4, pp. 648–654, 2020, doi: 10.13140/RG.2.2.32319.74403.

[14] R. D. Arifin, “Pengertian Twitter | Sejarah, Fitur, Manfaat,” *dianisa.com*, 2020. https://dianisa.com/pengertian-twitter/ (accessed Nov. 23, 2021).

[15] F. Z. Tala, “A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia,” *M.Sc. Thesis, Append. D*, vol. pp, pp. 39–46, 2003.