Normalisasi Teks Bahasa Indonesia Berbasis Kamus Slang Studi Kasus: *Tweet* Produk *Gadget* Pada Twitter

Riri Riyaddulloh[1], Ade Romadhony[2]

[1,2] Fakultas Informatika, Universitas Telkom, Bandung
riririyaddulloh@student.telkomuniversity.ac.id[1], aderomadhony@telkomuniversity.ac.id[2]

Abstrak

Sosial media adalah alat bantu untuk memperkaya informasi tentang *gadget*, informasi yang diperoleh dapat berupa atribut produk *gadget*, hingga harga dari suatu *gadget*. Twitter merupakan salah satu dari sosial media yang berperan sebagai alat bantu untuk memperkaya berbagai informasi, mulai dari informasi tentang *gadget* hingga menjadi sumber berita keluhan seseorang. Normalisasi teks adalah istilah yang digunakan untuk menyampaikan gagasan dengan mengubah format teks untuk memenuhi tujuan tertentu. Terkadang dalam sebuah *tweets* terdapat unggahan kata yang berisi kata-kata non baku atau dapat disebut kata *slang*, kata *slang* adalah ragam bahasa tidak resmi dan tidak baku yang sifatnya musiman, dipakai oleh kaum remaja atau kelompok sosial tertentu untuk komunikasi intern. Kata *slang* tersebut perlu dilakukan normalisasi yang mana langkah awalnya dengan cara me-reduksi setiap kata yang memiliki imbuhan menjadi kata yang seragam, yang bertujuan agar dapat diproses pada pemrosesan selanjutnya. Pada Tugas Akhir ini, penulis membangun sistem untuk menormalisasi kata *slang* dari tweets produk *gadget*. Proses normalisasi teks menggunakan model *word2vec* untuk mencari kata formal dengan *similarity* tertinggi terhadap sebuah kata *slang*. Hasil normalisasi dievaluasi pada sebuah task klasifikasi yang akan mengelompokkan sentiment *tweets* ke dalam 3 kelas, yaitu: Positif, Negatif, dan Netral. Hasil pengujian menunjukkan bahwa terdapat peningkatan akurasi klasifikasi pada data yang sudah dinormalisasi, dengan nilai akurasi sebesar 91%, dibandingkan dengan dataset tanpa normalisasi, dengan nilai akurasi sebesar 88%.

Kata kunci: *gadget*, kata *Slang*, *Slang List*, normalisasi teks, korpus, *word2vec*

Abstract

Social media is a tool to enrich information about gadgets, the information obtained can be in the form of gadget product attributes, to the price of a gadget. Twitter is one of the social media that acts as a tool to enrich various information, ranging from information about gadgets to being a source of news for someone's complaints Text normalization is a term used to convey ideas by changing the format of the text to fulfill a specific purpose. Sometimes in a tweet there are uploads of words that contain non-standard words or can be called slang words, slang words are a variety of informal and non-standard languages that are seasonal in nature, used by teenagers or certain social groups for internal communication. in which the *slang* word needs to be normalized, the initial step is to reduce every word that has an affix to a assorted word, which aims to be processed in the next processing. In this final project, the author builds a system to normalize slang words from tweets of gadget products. The text normalization process uses the word2vec model to find the formal word with the highest similarity to a slang word. The results of normalization are evaluated in a classification task that will classify sentiment tweets into 3 classes, namely: Positive, Negative, and Neutral. The test results show that there is an increase in classification accuracy in normalized data, with an accuracy value of 91%, compared to datasets without normalization, with an accuracy value of 88%.

Keywords: *gadget*, *slang words*, *Slang List*, text normalization, corpus, *word2vec*

1. Pendahuluan

1.1. Latar Belakang

Penggunaan sosial media di internet telah merambah kepada semua golongan masyarakat Indonesia khususnya dan seluruh dunia pada umumnya, mulai dari usia dini maupun yang sudah dewasa. Sering kali ditemukan kalimat yang sulit dimengerti di berbagai artikel yang tersedia di internet. Terkadang di beberapa sosial media terkenal seperti *Twitter, Facebook*, dan sebagainya terdapat unggahan kata yang berisi kata-kata singkatan ataupun kata-kata non baku dan dapat disebut pula sebagai kata *slang*. Kata *slang* memiliki makna yang sukar dipahami dan perlu ada solusi untuk memahami kata-kata tersebut agar pembaca dapat mengerti maksud dari suatu artikel/ bacaan yang disajikan, karena setiap pembaca memiliki taraf pemahaman yang berbeda antar individu. Terkait masalah tersebut, maka normalisasi teks adalah solusinya. Normalisasi teks merupakan istilah yang digunakan untuk menyampaikan gagasan untuk mengubah format teks untuk memenuhi persyaratan tertentu [1].

Sosial media merupakan alat bantu untuk memperkaya informasi tentang *gadget*, sehingga pengguna dapat yakin dalam situasi pembelian *gadget* yang diinginkan. Dalam menentukan sebuah produk yang akan dibeli oleh konsumen, maka konsumen perlu memperhatikan atribut produk untuk mempertimbangkan keputusan dalam melakukan pembelian. Atribut produk memiliki hubungan yang positif terhadap keputusan konsumen. Melalui atribut produk, suatu produk dapat dikomunikasikan dan disampaikan kepada konsumen agar dapat dilakukan penilaian dan keputusan dalam pembelian. Selanjutnya atribut produk tersebut akan mempengaruhi sikap konsumen dan perilaku konsumen sebelum melakukan pembelian [6].

Twitter merupakan salah satu media sosial yang menjadi sumber berita keluhan seseorang hingga menjadi tempat berbisnis, dan merupakan salah satu media sosial yang terpopuler di Indonesia, menurut Direktur Jenderal Kominfo Budi Setiawan yang dipublikasikan dalam kominfo.go.id menyatakan perkembangan dunia teknologi berkembang sangat pesat di dunia tak terkecuali indonesia dengan tercatat 19,5 juta warga indonesia sebagai pengguna Twitter [2].

Normalisasi teks adalah istilah yang digunakan untuk menyampaikan gagasan dengan mengubah format teks untuk memenuhi tujuan tertentu. Normalisasi teks penting digunakan, karena dalam sebuah sebuah teks/bacaan akan mengandung kata yang sulit dimengerti, mulai dari berbeda bahasa, berbeda penggunaan kata, hingga berbeda makna dari suatu kalimat dengan makna aslinya. Pendekatan normalisasi dibagi kepada 2 kelompok: yang pertama ada yang menerjemahkan bahasa non-standar ke dalam bahasa standar menggunakan informasi kontekstual dan yang kedua mengganti kata-kata yang berbasis leksikal dengan bentuk yang sesuai dengan bahasa standar [1]. Kata *Slang* adalah ragam bahasa tidak resmi dan tidak baku yang sifatnya musiman, dipakai oleh kaum remaja atau kelompok sosial tertentu untuk komunikasi intern [7]. Contoh kata *slang* ialah jika terdapat suatu kata misalnya "baek" dapat memiliki arti "baik, baiklah" dan sebagainya. Kata-kata tersebut perlu dinormalisasi agar pembaca dapat mengetahui maksud arti dari kata tersebut, oleh karena itu perlu dilakukan normalisasi teks agar selanjutnya dapat diproses dan mengklasifikasi teks yang telah dilabeli [11] untuk diperoleh akurasinya dan juga agar normalisasi teks ini sesuai dengan tujuan yang diinginkan.

Pada Tugas Akhir ini akan dilakukan penelitian untuk normalisasi teks yang sukar untuk dipahami dalam sebuah berita komentar *gadget*, sehingga pembaca dapat lebih mudah memahami sebuah teks. Penelitian ini melakukan normalisasi teks terhadap tweet yang mengandung kata non-baku. Proses normalisasi dibagi ke dalam beberapa tahap. Pada tahap preprocessing akan dilakukan *cleansing data* yang telah didapatkan dari proses *crawling* data dari Twitter, lalu pada tahap deteksi kata non-baku akan dilakukan proses seleksi kata-kata *slang* berdasarkan *list of basic words* dan *slang dictionary*. Selanjutnya akan dilakukan proses normalisasi teks, yaitu mengubah kata-kata *slang* dengan cara melihat frekuensi kata-kata yang memiliki kemiripan dengan kata baku yang tersedia dari korpus dengan mengecek kata target (*slang*) dapat memperoleh berapa jumlah kata-kata konteks (kata formal dari korpus) dari setiap proses pencarian kata *similarity*. Lalu dilakukan pelabelan terhadap dataset agar dataset dapat diproses pada proses klasifikasi. Pada tahap terakhir, data yang telah dihasilkan dari proses sebelumnya akan dilakukan evaluasi dengan task klasifikasi, alasan evaluasi dengan task klasifikasi karena sentiment cukup popular dan relatif sederhana pelabelannya [11], kemudian akan dihasilkan performansi dari riset yang telah dilakukan.

1.2. Identifikasi Masalah

Berdasarkan penjelasan dari latar belakang ini, permasalahan yang akan menjadi acuan pada penelitian ini adalah bagaimana melakukan normalisasi kata non baku yang terdapat pada *tweets* produk *gadget* untuk meningkatkan kinerja proses selanjutnya.

Selain itu, masalah yang menjadi acuan terhadap penelitian ini adalah bagaimana performansi dari model akhir pembangunan *slang dictionary* dari data *tweets gadget* dalam menormalisasi kata *slang*.

1.3. Tujuan

Tujuan dari penelitian yang dilakukan adalah untuk melakukan normalisasi kata *slang* dari *tweets gadget* serta melakukan analisa terhadap pengaruh normalisasi dan terhadap kinerja proses selanjutnya, yaitu klasifikasi teks.

1.4. Organisasi Tulisan

Laporan penelitian ini disusun dengan susunan sebagai berikut: bagian 1 yaitu penjelasan rumusan masalah dan metode penyelesaian terhadap penelitian ini; bagian 2 merupakan penjelasan tentang penelitian yang telah dilakukan yang berkaitan dengan normalisasi teks; bagian 3 merupakan penjelasan terhadap alur proses pembangunan model normalisasi teks; bagian 4 adalah pembangunan model normalisasi teks; dan bagian 5 merupakan penjelasan kesimpulan terhadap hasil dari penelitian ini.

2. Studi Terkait

2.1. API Twitter

Twitter menyediakan *Application Program Interface* (API) yang memungkinkan pengguna untuk mendapatkan data mereka, yang mana data-data tersebut ialah berupa status atau yang lebih dikenal dengan *tweets.* Proses pengambilan *tweets* dari API Twitter dapat dilakukan dengan menggunakan *engine web crawler* yang dapat merepresentasikan kembali data ke dalam bentuk web [3].

2.2. Preprocessing

Preprocessing adalah tahap proses awal *text mining* terhadap teks untuk mempersiapkan teks menjadi data yang dapat diolah lebih lanjut [3]. *Preprocessing* yang dilakukan, seperti: *lowercase*, *stemming*, dan lainnya agar dataset dapat diproses pada proses selanjutnya. *Preprocessing* digunakan untuk menjadikan dataset agar dapat digunakan pada proses-proses selanjutnya seperti menyeleksi kata-kata *slang,* me-normalisasi teks, dan seterusnya.

2.3. Slang Dictionary

Slang Dictionary adalah kata-kata dan ungkapan informal yang pada umumnya digunakan dalam bahasa lisan maupun tulisan,yang mana slang tersebut bersifat musiman [7]. Kemudian Muhammad Okky Ibrohim, dan Indra Budi [8] meneliti tentang pendeteksian bahasa Indonesia yang bersifat menghina, yang dimana kata-kata kotor tersebut biasanya diucapkan dari aspek binatang, kondisi, dan sebagainya. Penelitian [4] mengatakan *OOV detection*, *list of basic words* dan *slang dictionary* digunakan pada tahap OOV *detection*, yang mana *slang dictionary* tersebut diisi dengan hubungan kata yang di bangun untuk membantu dalam pengambilan keputusan kata yang akan diambil pada proses *word replacement.* Muhammad Okky Ibrohim, dan Indra Budi [8] mengatakan bahwa pada proses normalisasi, pertama diperlukan penghapusan beberapa atribut yang tidak dibutuhkan seperti *username*, *retweet*, alamat URL, dan sebagainya. Setelah penghapusan, langkah selanjutnya adalah merubah kata non-baku menjadi kata baku dengan menggunakan kamus yang mereka bangun sendiri yaitu *small slang dictionary* untuk menormalisasi kata non-baku pada datasetnya. Hasil perbandingan dari beberapa algoritma yang diimplementasikan [4] perbandingannya dibagi pada 3 grup sebagai berikut: Pada versi 1, algoritma di tes dengan mengabaikan proses *cleaning* dan *slang dictionary* yang tidak diperkaya (dengan data yang dikumpulkan dan didapatkan), didapatkan akurasi sebesar 75.78%; Pada versi 2.0; diimplementasikan *cleaning* dan *indexing*,

didapatkan akurasi sebesar 75.89%. Setelah ditambahkan dengan lebih banyak fungsi untuk mengenali kata 'wkwkwk', 'hahaha', akurasi didapatkan sebesar 84%; dan versi 3; didapatkan akurasi sampai 89.83%.

Berdasarkan riset yang dilakukan Muhammad Okky Ibrahim, dan Indra Budi, dievaluasi dengan berbagai skenario [8], skenario pertama (mengklasifikasi data pada 3 label, yaitu: *non abusive language*, *abusive but no offensive,* dan *offensive language*) menggunakan *Naive Bayes* dengan kata fitur unigram + bigram memberikan hasil 70.06%, diikuti *Naive Bayes* dengan kata fitur bigram + trigram didapatkan hasil 69.64%, dan *Naive Bayes* dengan *char quadgram* didapatkan hasil 69.55%.

Pada skenario kedua (mengklasifikasi data pada 2 label; non *abusive language*, dan *abusive language*) menggunakan *Naive Bayes* dengan kata unigram memberikan hasil 86.43%, diikuti oleh *Naive Bayes* dengan char trigram + quadgram didapatkan hasil 86.17%, dan *Naive Bayes* dengan kata unigram + bigram didapatkan hasil 86.12%.

Penulis menyimpulkan [8] bahwa metode NB lebih baik dari pada metode SVM atau RFDT untuk proses klasifikasi pada eksperimen ini. Penulis mengatakan NB lebih baik karena metode SVM dengan kata unigram hanya dapat mencapai akurasi maksimum sebesar 83.94% dan metode RFDT dengan kata unigram hanya dapat mencapai akurasi maksimum sebesar 83.42%. Kedua metode tersebut mendapatkan hasil maksimum yang berada di bawah akurasi metode NB dengan kata unigram yaitu sebesar 86.43%.

2.4. Word2vec

Word2vec merupakan sekumpulan beberapa model yang saling berkaitan yang digunakan untuk menghasilkan *Word Embeddings. Word Embeddings* merupakan sebutan dari seperangkat bahasa pemodelan dan teknik pembelajaran fitur pada *Natural Language Processing* (NLP) dimana setiap kata dari kosakata memiliki vektor yang mewakili makna dari kata tersebut dan kata-kata tersebut dipetakan ke dalam bentuk vektor bilangan riil [9]. Irwan Budiman, M Reza Faisal, dan Dodon Turianto Nugrahadi [10] menjelaskan bahwa Word2vec adalah dua lapisan neural network yang memproses teks. Word2vec memiliki 2 algoritma belajar yaitu *Continuous Bag-Of-Word* (CBOW) dan *Continuous Skip-Gram*. Dengan algoritma CBOW, urutan dari kalimat di dalam riwayat tidak mempengaruhi proyeksi. Algoritma ini memprediksi kata saat ini berdasarkan konteks. Sedangkan algoritma *skip-gram* memprediksi kata-kata yang berada disekitar suatu kata.

Ekstraksi fitur yang digunakan dengan memanfaatkan vektor word2vec untuk mengontrol jumlah fitur yang dihasilkan. Dengan membandingkan beberapa model yang dihasilkan sendiri dengan jumlah fitur yang bervariasi dan model yang telah disediakan Google, hal ini dilakukan untuk mengetahui jumlah fitur yang dapat menghasilkan kinerja klasifikasi terbaik dan didapat nilai kinerja tertinggi dengan akurasi sebesar 0.877 dengan jumlah fitur adalah 300 dari model yang dihasilkan sendiri [10]. Hasil evaluasi dari [9] melakukan pengujian berupa nilai koefisien antara penilaian sistem dan penilaian manual yang dibandingkan dengan penelitian terdahulu dengan skala yang sama. Data yang digunakan sebanyak 2162 data. Hasil pengujian menunjukkan rata-rata nilai percentage error dengan menggunakan Word2vec sebesar 59.5%, dan angka tersebut menunjukkan nilai error yang tinggi. Rumus yang digunakan ialah (1).

$$V_{sentence}(w) = \frac{1}{n} \sum_{i=1}^{n} v_{wi} \qquad (1)$$

Yang mana **w** adalah dokumen atau kalimat yang berisi kata-kata, **n** adalah jumlah kata pada dokumen **w** dan $\boldsymbol{v_{wi}}$ adalah vektor dari kata $\boldsymbol{wi}$.

2.5. TF-IDF

Metode Tf-Idf merupakan perhitungan yang mendeskripsikan seberapa pentingnya sebuah kata (*term*) terhadap sebuah dokumen dengan memberikan bobot pada setiap kata. Frekuensi kata adalah ukuran seringnya kemunculan kata dalam sebuah teks dan juga pada seluruh teks dalam korpus. Tf-Idf digunakan untuk mengubah teks menjadi angka yang dapat diproses pada *Machine Learning*, lalu dihitung jumlah kemunculan katanya dalam sebuah teks [12].

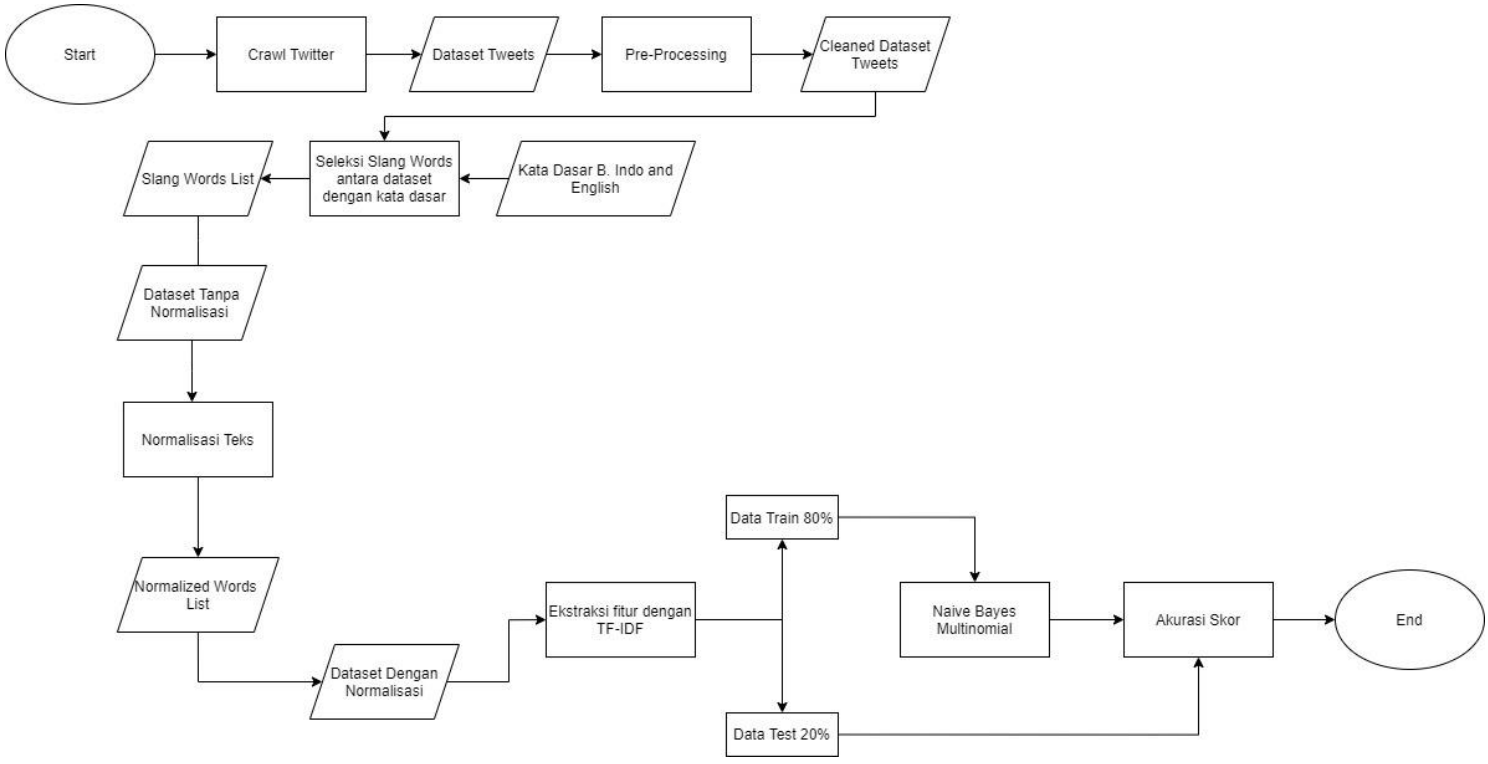
2.6. Naïve Bayes

Naïve Bayes adalah pengklasifikasi sederhana berdasarkan teorema Bayes. Yang mana ialah pengklasifikasi statistik yang melakukan prediksi probabilistik. pengklasifikasi bekerja dengan mengasumsikan bahwa distribusi kata dalam dokumen dihasilkan oleh model parametrik tertentu [11].
Ada beberapa macam model *Naive Bayes*. Pada penelitian ini menggunakan *Naïve Bayes Multinomial*, dengan rumus sebagai berikut (2).

$$P(c\ |d) = \frac{P(c)\sum_{i=1}^{n} P(wi\ |c)fi}{P(d)} \quad (2)$$

Keuntungan menggunakan *Naïve Bayes Multinomial* ialah mudah diimplementasikan dan sering memberikan kinerja prediksi yang lebih baik dan juga efisien dengan mempelajari data yang telah dilabeli [11]. Pada penelitian ini digunakan metode Naive Bayes untuk mengklasifikasikan kelas-kelas di dalam dataset yang dilabeli dengan kelas *Positif, Negatif,* dan *Netral.* Setelah proses pengklasifikasian dataset, dataset tersebut akan dilatih dan diuji untuk keperluan mendapatkan evaluasi dari model yang telah dibuat. Selanjutnya dilakukan task klasifikasi untuk menentukan hasil evaluasi dari model yang telah dibangun sebelumnya.

3. Sistem yang Dibangun



Gambar 1- Alur Eksperimen Klasifikasi Sentimen dengan Normalisasi Teks

Pada Tugas Akhir ini, untuk dapat membuat artikel dari sosial media yang bertujuan agar kinerja pada task lain yang menggunakan input hasil normalisasi, yaitu klasifikasi, memperoleh hasil yang lebih baik, maka perlu dilakukan proses normalisasi teks. Pada proses ini,hal pertama yang dilakukan adalah mendapatkan data dari Twitter dengan cara *crawling* data dari Twitter dengan konteks produk *gadget.* Lalu data tersebut dilakukan proses *cleansing* untuk menghilangkan komponen kata yang tidak diperlukan, agar dataset dapat di ekstrak pada proses selanjutnya. Setelah dilakukan proses *cleansing*, menginput kata dasar bahasa indonesia dan inggris ke dalam list

dataset untuk mendapatkan list kata-kata *slang*. Sebelum dilakukan proses seleksi kata *slang*, maka perlu dilakukan *stemming* untuk merubah kata-kata yang memiliki imbuhan agar menjadi kata dasar, lalu dilakukan seleksi kata *slang* yang terdapat pada dataset yang telah diperkaya dengan list kata-kata dasar dari proses sebelumnya.

Setelah proses *crawling* hingga proses seleksi kata *slang*, maka akan dilanjutkan pada proses pembangunan model hingga proses uji tes model. Data yang telah didapatkan dari seleksi kata-kata *slang* akan diubah menjadi kata baku dengan cara melihat frekuensi kata-kata yang mirip dengan kata baku yang tersedia dari korpus yang digunakan, dengan menggunakan *library Gensim* dan juga metode *word2vec*, maka akan menghasilkan kata-kata yang mendekati kata baku berdasarkan korpus yang digunakan. Lalu akan dilakukan pembangunan model dengan melakukan komparasi antara korpus dengan list slang dengan menggunakan model word2vec untuk mendapatkan terjemahan kata slang. Model yang telah dibangun akan dievaluasi performansinya menggunakan Ekstraksi Fitur dengan *Tf-Idf* untuk menjadikan data teks yang telah dihasilkan sebelumnya menjadi angka agar bisa diproses pada proses klasifikasi. Lalu akan dilakukan pemilihan skenario pada variabel *classifier Naïve Bayes* untuk mendapatkan skenario terbaik dari model yang dibuat.

3.1. *Crawling* Data

Pertama adalah *pre-processing* diawali dari *crawling* data dari twitter dengan konteks produk *gadget*. Proses crawling dilakukan untuk mendapatkan dataset yang akan digunakan untuk membuat model yang akan dibangun, dibutuhkan API *Twitter* untuk dapat melakukan *crawling* data twitter yang perlu diajukan kepada pihak Twitter itu sendiri. Crawling hanya bisa didapatkan maksimal 200 tweet per akun, dengan cara *crawling* data tweet pada setiap akun yang bertema kan seputar *gadget*, dan data *crawling* yang didapat untuk keperluan dataset pada tanggal 10 Oktober - 11 Oktober didapatkan total 1000 tweet.

Akun	Jumlah Tweet
@dgadgetin	200 tweet
@sobathape	200 tweet
@oppoindonesia	200 tweet
@xiaomiindonesia	200 tweet
@realmeindonesia	200 tweet
Total	1000 tweet

Tabel 1- Jumlah Tweet yang di Crawl dari Akun Twitter

3.2. Pre-Processing

Data yang telah didapatkan dari proses *crawling* sebelumnya akan dilakukan *cleansing* untuk menghilangkan komponen kata yang tidak diperlukan. Data *tweets* yang telah didapatkan dari proses *crawl* masih diperlukan *preprocessing* untuk mempermudah penggunaan data dalam pembangunan model normalisasi teks. *Preprocessing* adalah proses awal untuk mengolah data dari mulai data non-struktur hingga menjadi data terstruktur,yang bertujuan mendapatkan tingkat nilai relevan antara data dengan dokumen atau kata dengan kategori. Sehingga setelah dilakukan *Preprocessing* maka jumlah tweet dari dataset yang sebelumnya berjumlah 1000 tweet menjadi 989 tweet, dikarenakan terdapat beberapa tweet yang perlu di *drop* karena beberapa tweet tersebut berisi *null*. *Preprocessing* yang dilakukan pada Tugas Akhir ini dengan menggunakan *cleansing*. *Cleansing* merupakan proses membersihkan kata-kata yang tidak diperlukan untuk mengurangi noise [5]. Tahapan pada *cleansing* adalah :

- menghilangkan URL, simbol, penomoran;
- pengecekan pengejaan;
- menghilangkan tanda baca;
- *lowercase*;
- tokenisasi;
- menghilangkan *stopword*; dan
- *stemming*

Berikut contoh beberapa *tweets* dari sebelum proses *cleansing* dan setelah proses *cleansing.*

Username	Before Clean Tweet	Clean Tweet
@dgadgetin	Rp8 Jutaan! Unboxing Oneplus 6 Indonesia!: https://t.co/zcDm7ehjc8 via @YouTube	rp juta unboxing oneplus indonesia via
@dgadgetin	@SobatHAPE Selamatttttt!!!! 😍	selamatt
@sobathape	Sebuah ide bodoh https://t.co/OAOiOWUoFv	buah ide bodoh
@sobathape	@morningmelody Aku sih di banner Venti. Karena klo dapetnya venti lagi juga matan, buat naikkin konstelasi	aku sih di banner venti karena klo dapetnya venti lagi juga matan buat naikkin konstelasi
@sobathape	@dwikaputra Pas dibaca lagi... iya juga 😂	pas baca lagi iya juga

Tabel 2- Hasil Proses Cleansing Data

Proses stemming dengan menggunakan library *sastrawi* dengan tujuan untuk menghasilkan bentuk kata yang seragam. Contoh kata "menyesal" setelah dilakukan stemming dengan sastrawi akan berubah menjadi kata dasarnya yaitu "sesal". Berikut beberapa tahapan yang dilakukan oleh library satrawi ialah dengan memeriksa apakah merupakan akar kata atau bukan, menghilangkan *prefix, suffix,* dan *implix.* Apabila tahapan tersebut tidak dapat terselesaikan atau gagal diubah, maka kata tersebut akan dikembalikan kepada kata aslinya.

Sebelum Stemming	Setelah Stemming
Apakah boleh charge handphone sambil ditinggal tidur semalaman Ini penjelasannya	apakah boleh charge handphone sambil tinggal tidur malam ini jelas
dicatat ya kak	catat ya kak
selamat bergabung kakak	selamat gabung kakak
Perjalanan glow upku	jalan glow upku a thread
Besi ini fungsinya apaan sih Mau cari di Google ga tau namanya	besi ini fungsi apa sih mau cari di google ga tau nama

Tabel 3- Hasil Stemming Sastrawi

Pada tabel (3) beberapa tweet dalam dataset yang telah berubah ke bentuk yang seragamnya, total 989 tweet dalam dataset, terdapat 565 tweet yang diubah ke bentuk seragamnya dan 424 tweet yang tidak berubah, sehingga dapat disimpulkan bahwa tweet yang telah berubah ke bentuk seragamnya lebih banyak dari pada tweet yang belum tidak berubah kata seragamnya.

3.3. Seleksi Kata *Slang*

List kata dasar bersumber dari data yang sudah ada, didapatkan dari github.com[1] yang merupakan file yang berisi kumpulan kata-kata dasar, List kata-kata dasar yang telah didapat berjumlah 496.886 kata.

[1] github.com/sastrawi/sastrawi/blob/master/data/kata-dasar.txt

Setelah dilakukan *cleansing*, lalu diinputkan kumpulan kata dasar ke dalam list dataset. Selanjutnya akan dilakukan proses *stemming* untuk merubah kata-kata yang berimbuhan menjadi kata dasar. Lalu dilakukan seleksi semua kata slang yang terdapat pada dataset yang telah diperkaya dengan kata-kata dasar dari proses sebelumnya.

Proses selanjutnya adalah seleksi kata *slang*. Dataset yang sudah diinputkan kata dasar dan telah dilakukan proses *stemming* akan dilakukan proses penyeleksian kata *slang*. Setiap kata yang tidak termasuk dalam list kata dasar setelah proses penyeleksian kata *slang* akan otomatis masuk ke dalam list kata *slang*, dan begitu pula apabila setiap kata yang termasuk dalam list kata dasar, tidak akan masuk ke dalam list *slang*. Pada proses ini terseleksi kata *slang* sejumlah 938 kata dari 989 baris *tweets.*

Data dari proses ini selanjutnya akan digunakan pada proses normalisasi teks untuk mendapatkan kata-kata yang memiliki kemiripan dengan kata baku berdasarkan komparasi dengan korpus yang digunakan.

List Kata Slang
kirain
yg
samp
skarang
bener

Tabel 4- Hasil Seleksi Kata Slang

3.4. Normalisasi Teks

Pada penelitian [10] dilakukan normalisasi teks terhadap teks yang mengandung kata kasar dan juga kata *slang*, akan tetapi pada Tugas Akhir ini hanya melakukan normalisasi teks dengan model *word2vec* hanya pada teks yang terdeteksi sebagai kata *slang*.

Setelah penyeleksian kata-kata *slang* yang telah dilakukan pada proses sebelumnya, maka data tersebut akan digunakan untuk proses normalisasi teks. Langkah awal untuk menormalisasi teks adalah dengan membuat model *word2vec* dengan *library gensim*, model *word2vec* yang digunakan adalah algoritma CBOW dengan memproyeksikan vektor kata-kata konteks (w_{t-1}, w_{t+1}) untuk memprediksi vektor kata target w_{t} [9]. Model CBOW lebih mudah diproses karena semua kata-kata konteks langsung diproses menjadi satu vektor sebelum akhirnya digunakan untuk memprediksi vektor kata-kata target, karena korpus yang digunakan terbilang tidak terlalu besar, maka model CBOW ini cenderung lebih optimal memproses tiap file teks pada korpus. Korpus digunakan untuk dilakukan komparasi antara *slang list* dengan korpus untuk dapat diproses selanjutnya yaitu model *word2vec.*

Dengan menggunakan data yang cukup, *word2vec* mencari kata-kata dengan makna yang memiliki kemiripan tinggi berdasarkan riwayat kemunculannya. Prediksi tersebut dapat digunakan untuk menentukan asosiasi sebuah kata dengan kata-kata lainnya yang mirip. Cara kerja *word2vec* ialah dengan menghitung rata-rata vektor dari setiap kata yang ada dalam sebuah kalimat [10].

Model *word2vec* digunakan pada Tugas Akhir ini karena *word2vec* cenderung lebih mudah *smooth* terhadap informasi distribusional karena setiap kata konteks (kata dalam korpus) langsung diproses menjadi satu vektor sebelum akhirnya digunakan untuk memprediksi vektor kata target (kata *slang*).

Tweet	Kata Slang	Jumlah Kata Slang
yg bener adsense dlu baru glow up	Yg, bener, dlu, up	4
kirain linear aja kayak zelda	kirain	1
telah drama audio error hari akhir akhirnyaa bener juga thank you	Akhirnyaa, bener	2
saya dengar kalian lagi ngomongin windows phone	ngomongin	1
twitter tempat nyari ribut kan	nyari	1

Tabel 5-Tweet yang mengandung kata slang

Berdasarkan tabel (5) untuk setiap tweet dalam dataset mengandung beberapa kata *slang*, yang mana setiap tweet tersebut dapat mengandung 1, 2, atau 4 kata *slang*, akan tetapi tidak semua tweet dalam dataset mengandung kata *slang*. Pada analisis ini diperoleh sejumlah 684 tweet yang mengandung kata slang dan 304 tweet yang tidak mengandung kata *slang* dari jumlah total 989 tweet yang berada di dataset. 684 tweet tersebut selanjutnya dilakukan proses normalisasi dengan word2vec agar diperoleh kata termiripnya. Setelah dilakukan analisis banyaknya kata *slang* untuk setiap tweet, dapat disimpulkan bahwa lebih banyak tweet yang mengandung kata *slang* dibandingkan jumlah tweet yang tidak mengandung kata *slang.*

Setiap kata *slang* yang telah dilatih dengan model *word2vec* akan menghasilkan kata-kata yang termirip terhadap setiap kata *slang* yang menjadi target. Setiap kata *slang* akan memiliki frekuensi kata terdekat berdasarkan korpus yang digunakan, pada Tugas Akhir ini setiap kata *slang* yang ditemukan beberapa kata termiripnya meskipun lebih dari 3 kata, hanya akan di-set sejumlah 3 kata termirip yang memiliki kemiripan tertinggi, maka akan terdapat kata *slang* yang hanya memiliki 1 kata termirip dan juga terdapat kata *slang* yang memiliki 2 atau 3 kata termiripnya.

Slang List	Kata Termirip
asik	“seru”
banyaak	“banyakkk”, “banyakk”, “banyaaak”
bgt	“banget”
dapetin	“dapatin”, “dapatkan”, “ngedapaetin”

Tabel 6- Hasil Kata Termirip yang diperoleh

Kata-kata *slang* yang tidak memiliki kata-kata termirip dari list *Normalized Word* akan di-drop agar kata-kata *slang* yang memiliki kata termirip yang termasuk dalam list *Normalized Word*. Pada proses pembuatan model *word2vec* menghasilkan kata *slang* yang memiliki kemiripan dengan kata baku berjumlah 443 kata.

3.5. Ekstraksi fitur

Hasil dari proses komparasi korpus, yaitu dataset tanpa normalisasi dan dataset dengan normalisasi yang akan disesuaikan dengan *Tf-Idf* agar dataset dapat diproses pada naive bayes classifier. Ekstraksi Fitur adalah proses pengambilan objek yang dapat menggambarkan karakteristik dari objek tersebut. Untuk merepresentasikan hasil dengan ekstraksi fitur yaitu dengan menggunakan *Tf-Idf.* Kedua dataset yang telah dihasilkan masih berbentuk teks, maka diperlukan proses untuk merepresentasikan teks menjadi angka-angka yang dapat diolah lebih lanjut oleh *Machine Learning* dengan menggunakan *Tf-Idf*, frekuensi kemunculan kata (tf) dalam suatu dokumen akan dihitung dengan persamaan berikut [12] (3) (4).

$$tf(i) = \frac{freq_i(d_j)}{\sum_{i=1}^{k} freq_i \ \ (d_j)} \quad (3)$$

Sedangkan persamaan *Idf* dirumuskan sebagai berikut (3).

$$idf_i = \frac{\log|D|}{|\{d:t_i \in d\}|} (4)$$

Idf adalah nilai *Idf* dari setiap kata yang akan dicari, i adalah jumlah keseluruhan dokumen yang ada, lalu $d: t_i \in d$ adalah jumlah kemunculan kata pada semua dokumen. Dengan mengalikan kedua persamaan tersebut, maka akan didapatkan nilai dari *Tf-Idf* tersebut [12].

3.6. Klasifikasi

Setelah dilakukan normalisasi teks, maka proses selanjutnya adalah klasifikasi. Setiap tweet dalam dataset telah diberi label dengan 3 bentuk pelabelan, yaitu *Positif, negatif,* dan Netral. Pelabelan *Negatif* di-implementasikan pada 897 *tweets* dengan persentase 90%, dan untuk pelabelan *Positif* sejumlah 27 *tweets* dengan persentase 3%, dan untuk pelabelan Netral sejumlah 75 *tweets* dengan persentase 8% agar bisa dilakukan proses klasifikasi. Proses pengklasifikasian menggunakan *Naïve Bayes Multinomial*, yang merupakan model yang sering digunakan untuk mengklasifikasi teks.

4. Evaluasi

Mengacu pada proses di bab sebelumnya, akan dihasilkan dua dataset, dataset tanpa normalisasi dan dataset dengan normalisasi. Dua dataset ini akan diuji setelah dilakukan proses ekstraksi fitur dengan *Tf-Idf* lalu dievaluasi performansi klasifikasinya dengan metode *Naive Bayes*. Hasil yang akan dijadikan acuan pada Tugas Akhir ini adalah skor akurasi dari uji model.

4.1. Analisa Hasil Pengujian

Pada pengujian ini dilakukan pembagian train data dan test data untuk setiap dataset. Pembagian dataset tersebut dilakukan agar dapat dilakukan pada penghitungan skor tf-idf dan evaluasi performansi klasifikasi dengan *Naïve Bayes*. Pembagian dataset yang dilakukan berdasarkan pelabelan yang telah dilakukan pada proses klasifikasi dan untuk ukuran train size sebesar 80%, test size sebesar 20%.

Setelah dilakukan pembagian dataset akan dilakukan proses perhitungan skor *Tf-Idf* terhadap setiap kelas pada dataset. Untuk dataset tanpa normalisasi diperoleh skor *Tf-Idf* tertinggi sebesar 1,000000 pada label negatif, dan untuk dataset dengan normalisasi diperoleh skor *Tf-Idf* tertinggi yang sama dengan dataset tanpa normalisasi sebesar 1,000000 (lihat tabel 7 dan 8).

Label	Tweet	TF-IDF Score
Negatif	**yg bener adsense dlu baru glow up**	**1.000000**
Netral	**untuk jadwal luncur tingkat coloros**	**0.835924**
Positif	**kolom komentar yang kayak gin lebih perlu di highlight sebenernya kolom komentar jadi ruang bagi dan diskusi kal**	**0.599238**

Tabel 7- Skor Tertinggi Tf-Idf Dataset tanpa Normalisasi

Label	Tweet	TF-IDF Score
Negatif	**yuk cek ulas lengkap dari kompas tentang mi tv stick bisa kembali kamu dapat tanggal oktober pukul wib mulai dari harga rp di sini**	**1.000000**
Netral	**yang bayang david menang bukan gw trus semua percaya**	**0.938986**
Positif	**tumben ada hp vivo yang harga di bawah juta via**	**0.977854**

Tabel 8- Skor Tertinggi Tf-Idf Dataset dengan Normalisasi

4.2. Hasil eksperimen skenario

Setelah dilakukan ektraksi fitur dengan *Tf-Idf*, dilakukan evaluasi performansi klasifikasi dengan *Naïve Bayes*. Untuk evaluasi terhadap dataset tanpa normalisasi dan dataset dengan normalisasi digunakan nilai akurasi pengujian sebagai acuannya.

Accuracy	
Sebelum Normalisasi	**Setelah Normalisasi**
88%	**91%**

Tabel 9- Hasil Akurasi Dataset Sebelum & Sesudah Normalisasi

Berdasarkan hasil dari skor akurasi diatas, diperoleh hasil akurasi dataset dengan normalisasi lebih tinggi dibandingkan dataset tanpa normalisasi. Dataset sebelum normalisasi memperoleh hasil akurasi yang lebih rendah dibandingkan dataset setelah normalisasi dikarenakan dataset sebelum normalisasi masih mengandung teks yang belum dilakukan *cleansing*, yaitu dataset yang belum dapat diolah untuk proses selanjutnya, sehingga belum dapat diperoleh tingkat nilai relevan antara data dengan dokumen atau kata dengan kategori. Begitu pula masih terdapat banyak kata *slang* pada dataset yang belum di normalisasi, oleh karena itu dataset sebelum normalisasi memperoleh hasil akurasi 88% yang mana lebih rendah dari hasil akurasi dataset setelah normaisasi dengan mempreroleh hasil akurasi sebesar 91%.

Case	Accuracy	
	Sebelum Normalisasi	**Setelah Normalisasi**
Normalisasi menggunakan word2vec	**88%**	**91%**
Normalisasi tanpa word2vec	**75%**	**89%**

Tabel 10- Hasil Perbandingan dengan Riset Sebelumnya

Melihat tabel (10) dataset setelah normalisasi menggunakan *word2vec* memperoleh hasil akurasi 91% yang mana lebih tinggi dibandingkan normalisasi tanpa menggunakan *word2vec* dengan memperoleh hasil akurasi sebesar 89%. Normalisasi tanpa menggunakan *word2vec* berdasarkan dari penelitian sebelumnya [4] dan normalisasi dengan menggunakan *word2vec* ialah berdasarkan penelitian yang dilakukan pada Tugas Akhir ini. Dengan melihat tabel (10) normalisasi dengan word2vec lebih efektif dan memperoleh hasil akurasi yang lebih tinggi dibandingkan dengan normalisasi tanpa *word2vec*, meskipun pada penelitian ini terdapat beberapa kata *slang* yang tidak dapat di normalisasi, akan tetapi normalisasi dengan menggunakan word2vec tetap memperoleh hasil akurasi yang lebih baik dari pada normalisasi tanpa menggunakan word2vec. Sehingga dapat disimpulkan

bahwa dataset setelah dilakukan proses normalisasi dengan *word2vec* akan memiliki performansi klasifikasi yang lebih baik dibandingkan dengan normalisasi dataset tanpa menggunakan *word2vec* .

5. Kesimpulan

Berdasarkan analisis dan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa normalisasi teks bahasa indonesia berbasis kamus *slang* dapat digunakan untuk menormalisasi kata *slang* pada teks bahasa indonesia. Pada pengujian ini digunakan dua dataset yaitu dataset sebelum normalisasi dan dataset setelah normalisasi, yang mana dataset tanpa normalisasi ialah dataset yang belum dilakukan proses normalisasi, sedangkan dataset yang telah dilakukan normalisasi dan ektraksi fitur dengan *Tf-Idf* lalu dilakukan pengklasifikasian untuk diuji performansi klasifikasinya sehingga menjadi dataset dengan normalisasi. Pada riset ini normalisasi yang digunakan yaitu dengan *word2vec* dan diperoleh hasil akurasi yang lebih baik dari normalisasi tanpa menggunakan *word2vec*. Begitu pula pada riset ini ditemukan bahwa skor tertinggi tf-idf pada kedua dataset tersebut memiliki skor tertinggi yang sama sebesar 1,000000 pada label negatif. Hasil akurasi yang diperoleh dalam pengujian dengan menggunakan variabel Train Data Size sebesar 80%, Test Data Size sebesar 20% diperoleh hasil akurasi dataset tanpa normalisasi sebesar 88% dan diperoleh hasil akurasi dataset dengan normalisasi menggunakan *word2vec* sebesar 91%, Sehingga dapat disimpulkan bahwa dataset setelah dilakukan proses normalisasi akan memiliki performansi klasifikasi yang lebih baik terhadap dataset yang belum di normalisasi.

Referensi

[1] Magali Sanches Duran. Lucas Avanço. M. Graças Volpe Nunes. A Normalizer for UGC in Brazilian Portuguese. Proceedings of the ACL 2015 Workshop on Noisy User-generated Text, pages 38–47.

[2] Indonesia Peringkat Lima Pengguna Twitter. https://kominfo.go.id/content/detail/2366/indonesia-peringkat-limapengguna-twitter/0/sorotan_media. 02-11-2012.

[3] Bayu Rima Aditya. Penggunaan Web Crawler Untuk Menghimpun Tweets dengan Metode Pre-Processing Text Mining. Jurnal Infotel Vol. 7 No. 2 November 2015.

[4] Novita Hanafiah, Alexander Kevin, Charles Sutanto, Fiona, Yulyani Arifin, and Jaka. Text Normalization Algorithm on Twitter in Complaint Category. Novita Hanafiah et al. / Procedia Computer Science 116 (2017) 20–26.

[5] Jie Cheng, Russell Greiner. Comparing Bayesian Network Classifiers. In: UAI'99 Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. p. 101-108.

[6] Ni Luh Gede Diah Nirmala Dewi, Made Jatra. Pengaruh Atribut Produk Terhadap Keputusan Pembelian Handphone Di Kota Denpasar. E-Jurnal Manajemen Vol 2 No 2 (2013).

[7] Petrania T. Anis. Kata-Kata Slang Dalam Instagram. Jurnal Elektronik Fakultas Sastra Universitas Sam Ratulangi Vol 1,No 2(2018).

[8] Muhammad Okky Ibrohim, Indra Budi. A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media. Procedia Computer Science 135 (2018) 222–229.

[9] Faisal Rahutomo, Deasy Sandhya Elya Ikawati, Obby Auliyaur Rohman. Evaluasi Fitur Word2Vec Pada Sistem Ujian Esai Online. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika) Volume 04, Nomor 01, Juni 2019 : 36 - 45.

[10] Irwan Budiman, M Reza faisal, dan Dodon Turianto Nugrahadi. Studi Ekstraksi Fitur Berbasis Vektor Word2Vec pada Pembentukan Fitur Berdimensi Rendah. Vol 8 No. 1 , 2020 ©2020 Ilmu Komputer Unila Publishing Network All Rights Reserved

[11] Jiang Su, Jelber sayyad-Shirabad, Stan Matwin. Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes. Proceedings of the 28 th International Conference on Machine Learning, Bellevue, WA, USA, 2011.

[12] Munjiah Nur Saadah, Rigga Widar Atmagi, Dyah S. Rahayu, Agus Zainal Arifin. Sistem Temu Kembali Dokumen Teks dengan Pembobotan Tf-Idf Dan LCS. JUTI, Volume 11, Nomor 1, Januari 2013 : 17 – 20