Expansion Feature dengan *Word2Vec* untuk Analisis Sentimen pada Opini Politik di Twitter dengan Klasifikasi *Support Vector Machine*, *Naïve Bayes*, dan *Random Forest*

Muh. Dimas Lutfiyanto[1], Dr.Erwin Budi Setiawan, S.Si., M.T.[2]
[1,2]Fakultas Informatika, Universitas Telkom, Bandung
[1]muhdimaslutfiyanto@student.telkomuniversity.ac.id, [2]erwinbudisetiawan@telkomuniversity.ac.id

Abstrak

Pada saat ini pengguna media sosial sudah sangat banyak apalagi khususnya pengguna Twitter, yang mana para pengguna bisa memberikan pendapat atau memberikan informasi yang lainnya secara bebas dalam bentuk *tweet* apalagi dalam menanggapi atau memberikan informasi tentang kebijakan public, untuk *tweet*-nya sendiri dibatasi hanya 280 karakter setiap *tweet* yang di post dengan begitu akan terjadi kesalahan pada kosa kata yang di post. Penelitian ini akan diterapkan metode fitur ekspansi *Word2Vec* agar bisa mengurangi atau mengatasi terjadinya kesalahan dan ketidakcocokan kosa kata tersebut. Dan juga pada penelitian ini melakukan pengembangan dan perbandingan pada *system* Analisis Sentimen Twitter menggunakan metode fitur ekspansi *Word2Vec* dengan menggunakan algoritma klasifikasi *Support Vector Machine* (*SVM*), *Naïve Bayes*, dan *Random Forest*, dengan *system* tanpa menggunakan fitur ekspansi.

Kata kunci : *Sentiment Analysis, Word2Vec, Expansion Feature*.

1. Pendahuluan

Latar Belakang

Sentiment Analysis adalah sebuah klasifikasi teks, yang bertujuan untuk mengumpulkan sebuah teks dan juga meng-kelompokan dokumen yang sudah berisi sebuah opini yang akan dibagi menjadi positive, negative, atau netral. Dengan fitur yang akan dibagi menjadi dua pendekatan yaitu *rule based method* dan *statistical based method* dengan cara ini yang memanfaatkan *lexical resource* untuk pemrosesan sebuah fitur ekstraksi.[1]. Dan juga ini bisa melakukan proses otomatis untuk menentukan apakah segmen teks berisi konten yang objektif atau beropini, dan selanjutnya dapat menentukan polaritas teks [2]. Sedangkan *statistical method* menggunakan perhitungan yang matematis atau lebih dikenal dengan *machine learning*. Ini diartikan sebagai tugas memeriksa pendapat tentang entitas tertentu. Proses pengambilan keputusan seseorang dipengaruhi oleh opini yang dibentuk oleh pengguna. Ketika seseorang ingin membeli suatu produk secara *online* biasanya dia akan memulai dengan mencari *review* dan opini yang ditulis oleh orang lain tentang berbagai macam penawaran. Sebuah sistem yang secara otomatis menentukan sudut pandang yang akan memungkinkan pengguna untuk memahami sebagian besar opini yang diungkapkan di internet, dari *review* produk hingga sebuah kasus politik yang sedang memanas. Media jejaring sosial banyak digunakan oleh para masyarakat dari berbagai kalangan yang menyampaikan opini mereka tentang kebijakan pemerintah.

Media jejaring sosial adalah salah satu dari banyak yang digunakan untuk memberikan sebuah pendapat adalah Twitter. Twitter pada saat adanya kasus politik yang sedang memanas dapat memberikan sebuah tambahan wawasan dan gambaran kepada masyarakat tentang pendapat yang diberikan. Dengan pengklasifikasian yang bertujuan untuk mengklasifikasikan apakah *tweet* tersebut hasilnya positif dan negatif [3].

Hal itu bisa dilakukan dengan menerapkan salah satu teknik *word embedding*, ini adalah teknik untuk merubah kata menjadi sebuah vektor *real*. Teknik *word embedding* yang akan terkenal salah satunya adalah model *Word2Vec*. Metode ini berguna untuk merubah kata menjadi vektor dengan panjang N. *Multinomial Naïve Bayes* merupakan model pembelajaran mesin untuk klasifikasi yang dapat diimplementasikan terutama untuk data yang berupa dokumen teks. Berdasarkan penelitian sebelumnya membandingkan kinerja tiga metode klasifikasi, yaitu *Random Forest*, *Multinomial Naïve Bayes* (MNB), dan *Support Vector Machine* (*SVM*) dalam meng-analisis di data.

Berdasarkan beberapa penelitian sebelumnya, motivasi penelitian ini adalah melakukan eksperimen untuk meningkatkan nilai dalam membuat prediksi kepribadian menggunakan sebuah metode yang bernama *Multinomial Naïve Bayes*. *Multinomial Naïve Bayes* efektif untuk mengklasifikasikan data, dan metode *word embedding word2Vec* khususnya dokumen berbasis teks. Pada penelitian ini masalah yang dibahas yaitu apa pengaruh dari performansi *system* yang diterapkan fitur ekspansi menggunakan *Word2Vec* pada algoritma *Support Vector Machine* (*SVM*), *Naïve Bayes*, dan *Random Forest*. Dengan batasan penelitian pada penulisan ini dengan menggunakan data Bahasa Indonesia sebanyak 16.597 *tweet* yang bertopik pada kebijakan pemerintah di Indonesia, dengan proses pelabelan yang dilakukan

manual dan dibagi menjadi dua kategori Positif dan Negatif, nilai matriks yang digunakan ada dua akurasi dan juga *F1-Score*. Yang bertujuan untuk mengukur nilai performansi, juga menganalisis hasil dari *system* klasifikasi yang dibangun menggunakan fitur ekspansi dari *Word2Vec* pada data Bahasa Indonesia yang bertopik kebijakan pemerintah di Indonesia. Yang mana penulisan ini disusun dengan bab 2 yang membahas tentang teori dan studi literatur yang mana untuk mendukung penelitian ini, lalu bab 3 membahas teori terkain pada penelitian ini dan pemodelan *system* yang dibangun, bab 4 membahas tentang hasil, analisis, dan juga evaluasi terhadap model penelitian ini, dan terakhir pada bab 5 akan membahas hasil dari semuanya dengan kesimpulan yang diberikan.

2. Studi Terkait

Twitter adalah platform yang ideal. Yang mana proses mengklasifikasikan apakah tubuh teks menyampaikan positif, negatif, atau netral. Algoritma yang digunakan termasuk *Naïve Bayes*, Logistik Algoritma Mesin Vektor Regresi dan Dukungan dalam upaya untuk mengklasifikasikan lebih dari 4.000 *tweet* setelah melatih algoritma *Word2Vec* dengan lebih dari 10.000 *tweet*. Tingkat nilai tertinggi dihasilkan oleh pengklasifikasi sebesar 72% menggunakan Support Vector Classifier dan SG sebagai model pelatihan *Word2Vec*. Langkah peneliti selanjutnya adalah terus menyempurnakan parameter pengklasifikasi dalam upaya untuk meningkatkan tingkat nilai dan mencoba vektor yang berbeda [4].

Penelitian tidak akan menunjukan atau membahas masalah proses *stemming* yang dilakukan dan membuktikan bahwa hasil dari *pre-processing* dilakukan untuk menguji apakah hasil yang didapatkan sangat baik sehingga 93.11% adalah hasil akurasinya. Pada kasus ini bisa terjadi ketidak maksimalan karena adanya sebuah *mention* data yang terjadi [5].

Dalam makalah ini, peneliti telah menjelaskan bahwa pendekatan untuk klasifikasi topik *tweet* Indonesia. Untuk meringankan kosakata masalah ketidakcocokan, peneliti menerapkan suatu perluasan fitur menggunakan kata *embedding* berdasarkan *Word2Vec*. Peneliti menerapkan pendekatan ini menggunakan Kumpulan data *GoogleNews* dan *IndoNews* di *Naive Bayes*, *SVM*, dan pengklasifikasi Regresi Logistik. Berlawanan dengan sebelumnya temuan bahwa *SVM* selalu berada di antara yang berkinerja terbaik, peneliti eksperimen yang menerapkan perluasan fitur mengungkapkan bahwa ia cenderung menurunkan kinerja *SVM*. Menerapkan perluasan fitur dengan kumpulan data *Google IndoNews* dapat ditingkatkan secara konsisten kinerja saat menggunakan pengklasifikasi Regresi Logistik. NS penggunaan pengklasifikasi *Naive Bayes* memberikan hasil yang beragam. NS peningkatan kinerja kumpulan data menggunakan *Google News* adalah lebih baik daripada kumpulan data menggunakan *IndoNews* di *Naive Bayes*, *SVM*, dan pengklasifikasi Regresi Logistik [6].

Peneliti dalam paper ini telah memperkenalkan metode hibrida yang menggabungkan dasar fitur dan perluasan fitur untuk meningkatkan nilai dari analisis di Twitter. peneliti telah melatih *SVM*, Logit, dan NB untuk mengamati keakuratan menggunakan serangkaian perhitungan. Fitur ekspansi dapat digunakan dan terbukti meningkatkan akurasi. Dari dua ekspansi fitur, peneliti melihat peningkatan yang signifikan dalam perluasan fitur dengan fitur berbasis *tweet*, di mana nilai tertinggi 98,81% dicapai dengan menggunakan *logistic* pengklasifikasi regresi [7].

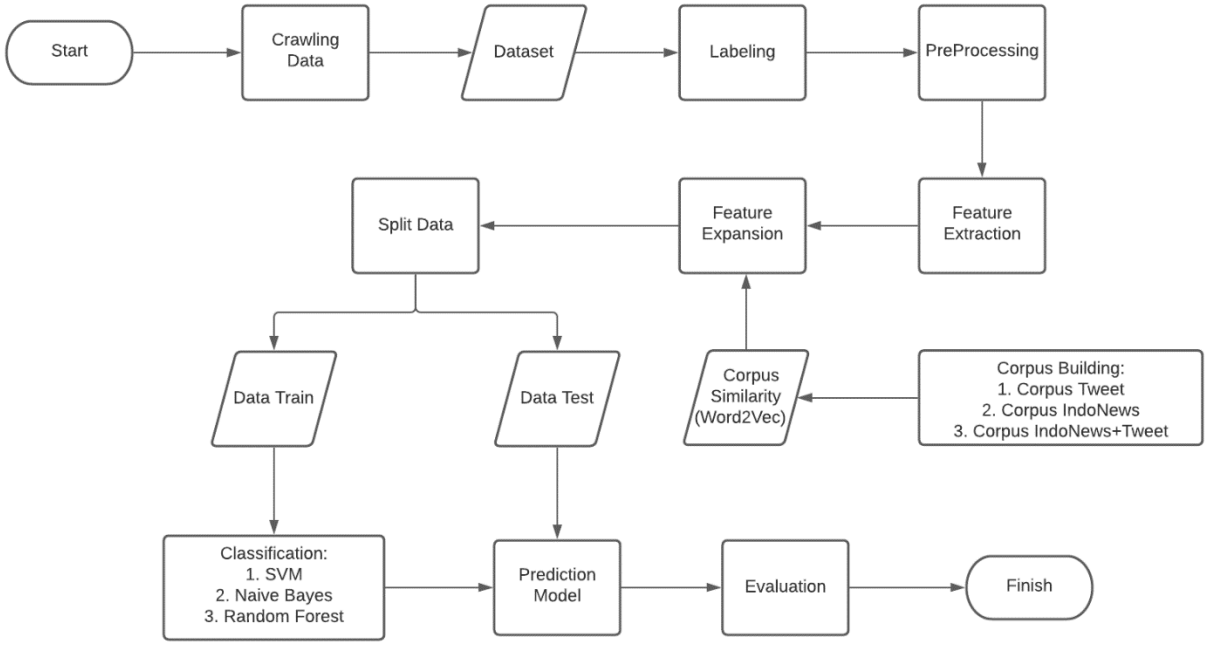
Peneliti dalam paper ini telah mempresentasikan bahwa hasil untuk analisis di Twiter. Yang menggunakan model *unigram* dari *state-of-the-art* sebagai dasar peneliti yang diusulkan dan keuntungan keseluruhan yang dilaporkan lebih dari 4% pada dua klasifikasi tugas: biner, positif melawan negatif dengan 3-arah positif melawan negatif dan melawan netral. Peneliti ini mempresentasikan bahwa serangkaian eksperimen yang komprehensif pada kedua tugas ini pada data yang dianotasi secara manual yang merupakan sampel acak dari aliran *tweet*. Pada fitur yang diberikan peneliti, analisis fiturnya mengungkapkan fitur yang paling penting yaitu yang menggabungkan polaritas kata sebelum dan tag bagian ucapannya. Peneliti secara tentatif menyimpulkan bahwa untuk data tidak jauh berbeda dengan genre lain[8].

Berdasarkan studi terkait mengenai algoritma klasifikasi, dapat disimpulkan bahwa algoritma dari klasifikasi *Logistic* Regression dan *Naïve Bayes* merupakan algoritma yang akurasinya mencapai ke posisi terbaiknya dibandingkan dengan algorima klasifikasi yang lain. Keunggulan dari metode ini sudah dibuktikan dalam penelitian [4][6]. Dengan tujuan dari penelitian ini memanfaatkan metode *Word2Vec* sebagai *Expansion Feature* dengan klasifikasi menggunakan algoritma *Support Vector Machine*, *Naïve Bayes*, dan *Random Forest*.

3. Sistem yang dibangun dengan menggunakan *Feature Expansion Word2Vec*

3.1. Alur penelitian menggunakan *Feature Expansion Word2Vec*

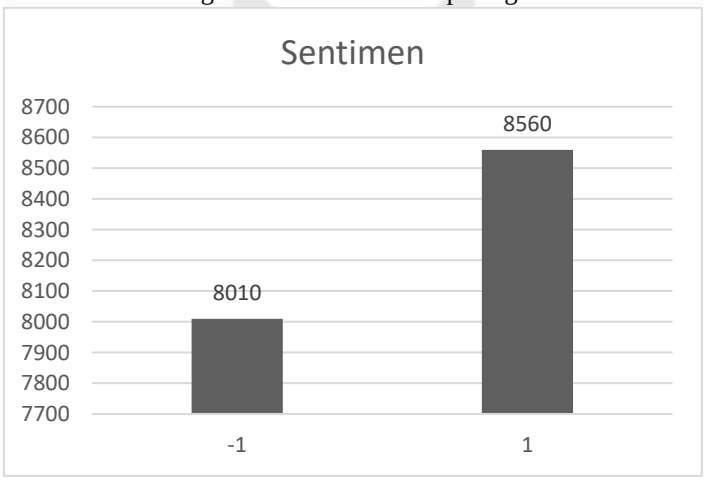
Gambar 1 disini menjelaskan tentang alur penelitian yang digunakan untuk penelitian ini yang membahas tentang *Word2Vec*.



Gambar 1 Persebaran Data *IndoNews* Sesuai Media *IndoNews*

3.2. Data Crawling

Penggunaan data *tweet* yang dikumpulkan ada sebanyak 16.597 tentang kebijakan pemerintah Indonesia, yang mana data tersebut akan dijadikan sebagai data latih dan juga data *test*, dari data yang sudah diambil tersebut ada beberapa *keyword* tentang politik yang berkaitan dengan kebijakan – kebijakan pemerintah seperti: #*omnibuslaw*, #uuciptakerja, covid-19, dan juga #ppkm, data *tweet* ini diambil dari tanggal 01 Januari 2020 hingga 01 Mei 2021. Persebaran data yang sudah diberi label sebagai berikut bisa dilihat pada gambar 2:



Gambar 2 Persebaran Data *Tweet*

Selanjutnya akan digunakan data *IndoNews* untuk pembuatan kamus, data ini diambil dari beberapa media *IndoNews* seperti Kompas, Tempo, Detik, Dan Yang Lainnya Dengan Topik Yang Berbeda Beda Seperti Agama, Bisnis, Budaya, Ekonomi, Entertainment, Hankam, Hukum, Iklan, Jurnalistik, Kesehatan, Keuangan, Motivasi, Olahraga, Pemerintahan, Pendidikan, Perhubungan, Politik, Sosial, Teknologi, Dan Umum, ada sebanyak 142.551, dan untuk pengambilannya dari tanggal 01 Mei 2016 hingga tanggal 01 Maret 2017 berikut adalah data *IndoNews* yang digunakan untuk pembuatan kamus katanya bisa dilihat pada data tabel 1.

Tabel 1 Persebaran Data *IndoNews* Sesuai Media *IndoNews*

Nama Redaksi	Jumlah
CNN Indonesia	29350
Detik	7975
Kompas	15056
Liputan6	252
Republika	53813
SIndoNews	22402
Tempo	13703
Total	**142.551**

3.3. Pelabelan Data

Data yang dibuat dikumpulkan menjadi sebuah dataset lalu akan dilakukan sebuah pengklasifikasian untuk menjadikan data yang dikumpulkan tadi agar mempunyai sebuah label kelas. Klasifikasi diperlukan agar bisa membedakan sebuah kelas data yang tadi telah dikumpulkan. Ada 2 Kelas yang digunakan pada saat ini yaitu.

1. Kelas positif yang bernilai 1 untuk yang sifatnya ujaran pendukung atau positif kepada 4 *keyword* diatas.
2. Kelas negative yang bernilai -1 untuk yang sifatnya berkebalikan dengan nilai 1 yang mana sifatnya ujaran kebencian atau tidak mendukung pada 4 *keyword* diatas.

Untuk tabel pelabelan data ada pada tabel 2.

Tabel 2 Pelabelan Data

Tweet	Kelas
Masalah hilirisasi menjadi salah satu masalah penting dalam bidang pertambangan. Dengan adanya UU Cipta Kerja diharapkan mampu mengatasinya.	1
Waktu blm ada RUU Cipta Kerja aja 10 juta lapangan kerja,,, masa sekarang udh ada RUU Cipta Kerja cuma 2 JT lapangan kerja... Turun drastis dong klo gt..???	-1

3.4. *Preprocessing* Data

Pre-processing atau praproses adalah suatu tahap data yang salah satu prosesnya akan disiapkan sebuah data mentah yang belum dilakukan apapun proses yang lainnya. Data praproses akan dilakukan dengan cara dihapusnya data yang tidak sesuai lalu datanya diubah kebentuk yang mudah untuk diproses oleh system yang akan dibangun. Praproses sangat penting dalam dalam hal ini karena akan dilakukan untuk menganalisis, terutama pada kasus media sosial yang mana sebagian besar kata-kata atau kalimat yang tidak formal dan tidak terstruktur memiliki *noise* yang besar. Pada penelitian ini dilakukan data ekstraksi menjadi sebuah data yang nantinya akan siap untuk digunakan pada suatu teknik yang bernama *mining*. Ekstraksi data yang dilakukan tersebutlah yang dipanggil dengan praproses [5]. Dan berikut adalah tahap-tahap pada *preprocessing*:

1. *Special Text Removal (Cleaning)*
   Teknik menghilangkan semua angka, tanda baca, alamat situs/URL (“http://”, “www…com”), *hashtag* (“#”), dan *tag username* (*@username*).
2. *Lower Case*

Teknik mengubah kalimat menjadi huruf kecil.

3. Normalisasi Kata
   Teknik mengubah kata singkatan, salah penulisan (*typo*), kata alay (informal), dan kata gaul menjadi sebuah kata yang lebih formal dengan bantuan kata kamus manual yang telah dibangun.
4. *Stopwords Removal*
   Teknik menghapus kata-kata yang umum digunakan dan kata yang tidak memiliki arti khusus seperti kata ganti, preposisi, dan konjungsi.
5. *Stemming*
   Teknik mengembalikan semua kata ke dalam bentuk kata dasarnya dengan membuang kata imbuhan awal maupun akhir. Untuk proses ini diperlukan *library* sastrawi.
6. *Tokenization*
   Teknik pemisahan kalimat menjadi kata per kata.

3.5. *Term Frequency - Inverse Document Frequency* (*TF-IDF*)

TF-IDF adalah biasa menghitung setiap bobot katanya dengan menggunakan *information retrieval* dan ini adalah salah satu metode yang sering digunakan, karena hal ini menjadikannya efisien, mudah diterapkan dan hasilnya pun juga akurat [9]. Cara menghitungnya dari nilai pada setiap token (kata) diberikan vector yaitu *Term Frequency* (*TF*) dan *Inverse Document Frequency* (*IDF* pada dokumen korpus [10].

3.6. *Word2Vec*

Salah satu metode *embedding word* yang berguna adalah *Word2Vec* yang bisa merepresentasikan kata menjadi sebuah vektor dengan panjang N. Misalnya sebuah kata “Omnibus” di vektor dengan panjang 5 yang direpresentasikan menjadi: [0.2, 0.4, -0.8, 0.9, -0.5]. Tidak hanya kata yang secara sintaktik tapi vector itu juga bisa merepresentasikan secara semantik atau secara makna. Yang mana Rong [9], mengatakan bahwa model ini dan aplikasinya serta penggunaan dan teknik untuk mengoptimalkan analisis data. Hal ini mencakup dua model: model *bag-of-words* (CBOW) dan model skip-gram (SG) [9].

3.7. *Support Vector Machine* (*SVM*)

Klasifikasi ini menggunakan *machine learning* yang memprediksi sebuah kelas berdasarkan model dari hasil proses *training*. Klasifikasinya memisahkan satu kelas dengan kelas yang lain (*decision boundary*) atau *hyperplane* yang akan melakukan pencarian garis pembatas. Menggunakan *support vector* dan nilai margin untuk mencarikan nilai *hyperplane* [9]. Tujuan memberi label pada *tweet* yaitu inputan data yang dilakukan dengan proses pembobotan makanya vektor tersebut bisa direpresentasikan. *Training* yang dilakukan menghasilkan sebuah nilai pada klasifikasi yang digunakan pada proses *testing*[11].

3.8. *Naïve Bayes*

Karena sangat mudahnya diimplementasikan, waktu yang cepat, mudah diimplementasikan dengan strukturnya yang cukup sederhana dan untuk tingkat efektifitasnya cukup tinggi ini menjadi salah satu metode yang cukup populer untuk data *mining*. Bukan msalah yang besar untuk klasifikasi ini menggunakan dataset yang jumlahnya besar karena hasil yang diberikan sangat cepat dan tingginya nilai[12].

3.9. *Random Forest*

Algoritma ini adalah pembelajaran mesin yang diawasi berdasarkan pembelajaran *ensemble*. Pembelajaran *ensemble* adalah jenis pembelajaran yang menggabungkan berbagai jenis algoritma atau algoritma yang sama beberapa kali untuk membentuk model prediksi yang akurat. Algoritma ini menggabungkan beberapa algoritma dengan jenis yang sama, yaitu beberapa *tree* keputusan, menghasilkan beberapa *tree*. Algoritma ini dapat digunakan untuk tugas regresi dan klasifikasi. Pengklasifikasi *RF* dapat dijelaskan sebagai kumpulan pohon pengklasifikasi terstruktur. Ini adalah versi lanjutan dari *Bagging* sehingga keacakan yang akan ditambahkan ke dalamnya. Alih-alih memisahkan

setiap *node* menggunakan pemisahan terbaik di antara semua variabel, membagi setiap *node* terbaik di antara subset prediktor yang dipilih secara acak pada node tersebut [13].

3.10. *Confusion Matrix*

Visualisasi data yang biasa digunakan dengan salah satu caranya dari untuk *supervised learning*. Contoh suatu kelas prediksi yaitu dengan kolom yang pada matriks. Hasil yang diberikan yaitu *recall, precision, accuracy,* dan *error rate* dari proses *Confusion matrix*. Contohnya ada pada tabel 3.

Tabel 3 *Confusion Matrix*

		Prediksi	
Actual	Negatif	A	C
	Positif	B	D

Keterangan:
A = Hasilnya tepat dengan memiliki sifat negatif prediksinya.
B = Hasilnya salah dengan memiliki sifat positif prediksinya.
C = Hasilnya salah dengan memiliki sifat negatif prediksinya.
D = Hasilnya tepat dengan memiliki sifat positif prediksinya

Beberapa persyaratan yang telah didefinisikan untuk matrik klasifikasi diantaranya sebagai berikut:

1. *Accuracy* adalah jumlah proporsi yang prediksi benar. Dengan rumus yang digunakan adalah: AC = (A +D) / A + B + C + D
2. *Recall* adalah kasus positifnya teridentifikasi dengan proporsi yang benar, persamaan yang bisa dihitung dengan: TP = D / C +D Tingkat positif salah (FP) adalah kasus yang negatifnya salah proporsi dan diklasifikasikan sebagai positif, dihitung dengan persamaan yang menggunakan: FP = B / A+B Tingkat negatif sejati (TN) kasus negatif lah hasilnya sebagai proporsi yang peng-klasifikasiannya adalah benar, dan dihitung persamaannya dengan menggunakan: TN = A / A+B Tingkat negatif palsu (FN) ini adalah kasus positif dengan diklasifikasikan sebagai negative dan proporsi yang salah, persamaan yang dihitung dengan menggunakan: FN = C / C +D
3. *Precision* (P) adalah prediksi yang benar dengan kasus positif, dan persamaan yang dihitung dengan menggunakan: P = D / B + D
4. *F-Score* atau *F1-Score* merupakan rata-rata yang nantinya akan dibobotkan (*harmonic mean*) dengan *precision* dan *recall*.

4. Evaluasi

Pengujian yang hasilnya telah dilakukan pada *system* yang telah dibangun ada pada bagian ini penjelasannya.

4.1. *Preprocessing* Data

Elemen-elemen pengganggu pada data dibersihkan, seperti tanda baca, simbol-simbol, tautan halaman, angka, *hashtag, mention,* dan *web*. Pembersihan data ini dilakukan untuk meningkatkan kualitas data saat digunakan untuk pelatihan model [14]. Tahap – tahap yang perlu dilakukan pada *preprocessing* data bisa dilihat pada tabel 4, 5, 6, 7, 8, dan 9.

1. *Special Text Removal (Cleaning)*

Tabel 4 *Preprocessing* Data *Special Text Removal*

Sebelum	Sesudah

@DimasLutfiyanto : OmnibusLaw RI Sangat Tidak Kompeten #omnibuslaw	OmnibusLaw RI Sangat Tidak Kompeten

2. Case Folding / Lower Case

Tabel 5 *Preprocessing* Data Case Folding/Lower Case

Sebelum	Sesudah
OmnibusLaw RI Sangat Tidak Kompeten	omnibuslaw ri sangat tidak kompeten

3. Normalisasi Kata

Tabel 6 *Preprocessing* Data Normalisasi Kata

Sebelum	Sesudah
omnibuslaw ri sangat tidak kompeten	omnibuslaw republik indonesia sangat tidak kompeten

4. Removal Stop Words

Tabel 7 *Preprocessing* Data Stop Words

Sebelum	Sesudah
omnibuslaw republik indonesia sangat tidak kompeten	omnibuslaw republik indonesia sangat tidak kompeten

5. *Stemming*

Tabel 8 *Preprocessing* Data *Stemming*

Sebelum	Sesudah
omnibuslaw republik indonesia sangat tidak kompeten	omnibuslaw republik indonesia sangat tidak kompeten

6. Tokenizing

Tabel 9 *Preprocessing* Data Tokenizing

Sebelum	Sesudah
omnibuslaw republik indonesia sangat tidak kompeten	[‘omnibuslaw’, ‘republik’, ‘indonesia’, ‘sangat’, ‘tidak’, ‘kompeten’]

4.2. Pembuatan Kamus Kata *Word2Vec*(Corpus)

Pemograman yang digunakan dari https://medium.com/@yunusmuhammad007/*TF-IDF*-term-frequency-inverse-document-frequency-representasi-vector-data-text-2a4eff56cda. Untuk pembuatan kamus kata digunakan

teknik *word embedding Word2Vec* model Skip Gram. Kamus kata berupa kumpulan kata yang diurutkan nilai similaritasnya dari tertinggi hingga terendah. Hingga begitu hasil yang didapatkan digambarkan seperti berikut.

1. Kamus Kata Data *Tweet*

   Didapatkan hasil kosakata sebanyak 15.948 kata dan berikut contoh hasil kata-kata yang mirip bisa dilihat pada tabel 10.

Tabel 10 Corpus *Tweet*

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
omnibus	omnimbus	ominbus	ombibus	kadalin	teriakin
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	omnisbus	batal	mk	bahas	*omnibuslaw*

2. Kamus Kata Data *IndoNews*

   Didapatkan hasil kosakata sebanyak 225.876 kata dan berikut contoh hasil kata-kata yang mirip bisa dilihat pada tabel 11.

Tabel 11 Corpus *IndoNews*

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
jakarta	dki	ahok	provinsi	apbdp	pemprov
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	gubernur	basuki	thahja	purnama	dpr

3. Kamus Kata Data *IndoNews+Tweet*

   Didapatkan hasil kosakata sebanyak 229.984 kata dan berikut contoh hasil kata-kata yang mirip bisa dilihat pada tabel 12.

Tabel 12 Corpus *Tweet*

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
yogyakarta	jogja	keraton	borobudur	candi	malioboro
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	prambanan	merapi	sultan	ugm	tugu

4.3. *Feature Expansion*

Sebagai contoh untuk eksperimen dengan ukuran fitur top 10 menggunakan kamus kata data *IndoNews*, pada representasi vektor fitur *TF-IDF* kata “omnibus” memiliki bobot kata nol. Namun, pada dokumen, *tweet* tersebut memiliki kata “ciptakerja”, dikarenakan “ciptakerja” ada pada *table* fitur top 10 “omnibus” maka kata “omnibus” tersebut bernilai bobot sepert nilai dari “ciptakerja” seperti tabel 13 berikut.

Tabel 13 Corpus *Tweet*

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
omnibus	*omnibuslaw*	law	cilaka	ciptaker	omnimbus
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	ciptakerja	ominbus	rkuhp	gembar	pertembakuan

4.4. Klasifikasi

Seletah melalui tahap *preprocessing*, pembobotan kata, dan proses *Feature Expansion*, kemudian proses dilanjutkan ke tahap klasifikasi menggunakan *SVM*, *Naïve Bayes*, dan *Random Forest*. Sebelumnya akan dilakukan penggantian rasio data latih dan data uji terlebih dahulu untuk masing-masing algoritma agar hasil yang diberikan lebih optimal. Pada tiap sistem klasifikasi dilakukan pengulangan eksekusi program sebanyak 3 kali yang diambil nilai rata-rata akurasinya dan menggunakan data uji dan data latih yang diperbandingkan dengan 20:80. Lalu, juga diambil paling tinggi akurasinya dari percobaan penggantian rasio pada data latih dan data uji.

4.5. Skenario dan Hasil Pengujian

Untuk hasil pengujian ini menggunakan nilai matriks, sebelum membandingan dengan hasil fitur ekspansi disini akan memperlihatkan hasil dari baseline dari setiap algoritma *SVM*, *Naïve Bayes* dan juga *Random Forest* dengan rasio perbandingan data *train* dan data *test* 80:20 dan juga memperlihatkan nilai matriks. Untuk hasilnya bisa dilihat pada tabel 14, 15, dan 16.

Tabel 14 Hasil Performansi pada *Support Vector Machine*

Classifier	Akurasi (%)	*F1-Score*
Baseline (*SVM*)	77.60	0.777
Baseline (*SVM*) + *TF-IDF*	**78.20 (+0.77)**	**0.791 (+0.018)**

Tabel 15 Hasil Performansi pada *Naïve Bayes*

Classifier	Akurasi (%)	*F1-Score*
Baseline (*Naïve Bayes*)	78.90	0.801
Baseline (*Naïve Bayes*) + *TF-IDF*	**80.10 (+1.52)**	**0.806 (+0.006)**

Tabel 16 Hasil Performansi pada *Random Forest*

Classifier	Akurasi (%)	*F1-Score*
Baseline (*Random Forest*)	80.40	0.825
Baseline (*Random Forest*) + *TF-IDF*	**82.40 (+2.49)**	**0.85 (+0.03)**

Untuk bagian selanjutnya yaitu membandingkan hasil ketika menggunakan fitur ekspansi dengan kamus kata data *tweet*, *IndoNews*, dan gabungan dari *tweet* juga *IndoNews*. Pengujian yang dilakukan juga menggunakan ratio 80:20 untuk perbandingan data *train* dan data *test*. Untuk pengambilan fiturnya dari nilai similaritas tertinggi dari kamus kata yang dibuat dengan nilai 1, 5, dan 10. Dengan setiap klasifikasinya dilakukan pengulangan sebanyak 3 kali untuk mengambil nilai rata-rata dari hasil akurasinya.

Hasil yang telah menggunakan fitur ekspansi pada masing masing algoritma ada di bagian tabel 17, 18, dan 19 yang dapat dilihat bahwa.

1. Hasil Akurasi
   Hasil dari nilai akurasi dan *F1-Score* pada fitur ekspansi menggunakan algoritma *Support Vector Machine* bisa dilihat pada *table* 4.6.1. Hasil dari keseluruhan nilai mengalami peningkatan sehingga nilai tertingginya ada pada fitur top 1 dengan menggunakan kamus kata data *IndoNews+tweet* sebesar 78.70% dan untuk *F1-Score* pun ada peningkatan dengan nilai tertinggi ada di fitur top 5 dengan menggunakan kamus kata data *IndoNews* sebesar 0.792 dengan adanya peningkatan dengan nilai baseline yang sebelumnya 77.60% untuk akurasi dan *F1-Score* 0.777 bisa dilihat pada tabel 17.

Tabel 17 Hasil *Feature Expansion* pada *Support Vector Machine*

	Akurasi (%)			***F1-Score***		
	Corpus *Tweet*	**Corpus *IndoNews***	**Corpus *IndoNews+Tweet***	**Corpus *Tweet***	**Corpus *IndoNews***	**Corpus *IndoNews+Tweet***
Top 1	77.20 (-0.52)	77.90 (+0.39)	**78.70 (+1.42)**	0.777 (0)	0.784 (+0.009)	0.791 (0.018)
Top 5	78.20 (+0.77)	78.50 (+1.16)	77.50 (-0.13)	0.785 (0.010)	**0.792 (+0.019)**	0.775 (-0.002)
Top 10	77.50 (-0.13)	78.00 (+0.52)	77.30 (0.39)	0.78 (0.003)	0.786 (+0.011)	0.777 (0)

Hasil dari nilai akurasi dan *F1-Score* pada fitur ekspansi menggunakan algoritma *Naïve Bayes* bisa dilihat pada *table* 4.6.2. Hasil dari keseluruhan nilai mengalami peningkatan sehingga nilai tertingginya ada pada fitur top 5 dengan menggunakan kamus kata data *IndoNews* sebesar 81.90% dan untuk *F1-Score* pun ada peningkatan dengan nilai tertinggi ada di fitur top 5 dengan menggunakan kamus kata data *IndoNews* sebesar 0.821 dengan adanya peningkatan dengan nilai baseline yang sebelumnya 80.10% untuk akurasi dan *F1-Score* 0.806 bisa dilihat pada tabel 18.

Tabel 18 Hasil *Feature Expansion* pada *Naïve Bayes*

	Akurasi (%)			***F1-Score***		
	Corpus *Tweet*	**Corpus *IndoNews***	**Corpus *IndoNews+Tweet***	**Corpus *Tweet***	**Corpus *IndoNews***	**Corpus *IndoNews+Tweet***
Top 1	79.70 (+1.01)	79.40 (+0.63)	79.20 (+0.38)	0.763 (-0.047)	0.781 (-0.024)	0.778 (-0.028)
Top 5	81.40 (+3.17)	**81.90 (+3.80)**	80.90 (+2.53)	0.804 (+0.003)	**0.821 (+0.024)**	0.813 (+0.014)
Top 10	79.30 (+0.51)	80.30 (+1.77)	79.10 (0.25)	0.782 (-0.023)	0.775 (-0.032)	0.766 (-0.043)

Hasil dari nilai akurasi dan *F1-Score* pada fitur ekspansi menggunakan algoritma *Random Forest* bisa dilihat pada *table* 4.6.3. Hasil dari keseluruhan nilai mengalami peningkatan sehingga nilai tertingginya ada pada fitur top 5 dengan menggunakan kamus kata data *IndoNews+tweet* sebesar 83.80% dan untuk *F1-Score* pun ada peningkatan dengan nilai tertinggi ada di fitur top 5 dengan menggunakan kamus kata data *IndoNews+tweet* sebesar 0.861 dengan adanya peningkatan dengan nilai baseline yang sebelumnya 82.40% untuk akurasi dan *F1-Score* 0.850 bisa dilihat pada tabel 19.

Tabel 19 Hasil *Feature Expansion* pada *Random Forest*

	Akurasi (%)			***F1-Score***		
	Corpus *Tweet*	**Corpus *IndoNews***	**Corpus *IndoNews+Tweet***	**Corpus *Tweet***	**Corpus *IndoNews***	**Corpus *IndoNews+Tweet***
Top 1	79.70 (+0.62)	79.40 (+1.24)	80.40 (0)	0.819 (-0.007)	0.841 (+0.019)	0.801 (+0.029)
Top 5	83.50 (+3.86)	82.20 (+2.24)	**83.80 (+4.23)**	0.841 (+0.019)	0.849 (+0.029)	**0.861 (+0.043)**
Top 10	81.10 (+0.87)	80.10 (-0.37)	81.30 (+1.12)	0.818 (-0.007)	0.814 (-0.013)	0.824 (-0.001)

4.6. Analisis Hasil Pengujian

Hasilnya, percobaan menggunakan *Feature Expansion* dengan penggunaan kamus kata dan ukuran fitur yang berbeda, mendapatkan hasil yang berbeda-beda pula. Berdasarkan hasil akurasi dan *F1-Score* yang didapatkan, dapat dilihat bahwa terjadi peningkatan pada setiap sistem yang ditambahkan teknik *TF-IDF* untuk membobotkan kata. Lalu, ketika sistem mengimplementasikan *Feature Expansion*, nilainya pun juga ikut meningkat.

5. Kesimpulan

Pada penelitian ini, telah dilakukan penelitian untuk analisis sentimen menggunakan teknik *Feature Expansion Word2Vec*. *Support Vector Machine* (*SVM*), *Naïve Bayes*, dan *Random Forest* adalah yang digunakan pada klasifikasi di penelitian ini. *Feature Expansion* dilakukan dengan menggunakan 3 kamus data (*Tweet*, *IndoNews*, dan *Tweet+IndoNews*). Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa, implementasi metode *Word2Vec* terbukti dapat meningkatkan nilai akurasi dan *F1-Score* pada sistem. Hasil terbaik ada pada klasifikasi *Random Forest* dengan accuracy sebesar 83.80% (+4.00%). Dan klasifikasi yang paling stabil yaitu *SVM*.

REFERENSI

[1] M. Lailiyah, "Sentiment Analysis Menggunakan Rule Based Method Pada Data Pengaduan Publik Berbasis Lexical Resources," 2017, [Online]. Available: http://repository.its.ac.id/42409/.

[2] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," *IEEE Access*, vol. 6, no. c, pp. 23253–23260, 2018, doi: 10.1109/ACCESS.2017.2776930.

[3] Z. Jianqiang and G. Xiaolin, "Comparison research on text pre-processing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, no. c, pp. 2870–2879, 2017, doi: 10.1109/ACCESS.2017.2672677.

[4] J. Acosta, N. Lamaute, M. Luo, E. Finkelstein, and A. Cotoranu, "Sentiment Analysis of Twitter Messages Using Word2Vec," *Proc. Student-Faculty Res. Day, CSIS, Pace Univ.*, pp. C8-1-C8-7, 2017.

[5] S. Mujilahwati, "Pre-Processing Text Mining Pada Data Twitter," *Semin. Nas. Teknol. Inf. dan Komun.*, vol. 2016, no. Sentika, pp. 2089–9815, 2016.

[6] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, "Feature expansion using word embedding for tweet topic classification," *Proceeding 2016 10th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2016 Spec. Issue Radar Technol.*, no. 2011, 2017, doi: 10.1109/TSSA.2016.7871085.

[7] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, "Feature expansion for sentiment analysis in twitter," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, vol. 2018-Octob, pp. 509–513, 2018, doi: 10.1109/EECSI.2018.8752851.

[8] R. Passonneau, "Sentiment Analysis of Twitter Data."

[9] X. Rong, "word2vec Parameter Learning Explained," pp. 1–21, 2014, [Online]. Available: http://arxiv.org/abs/1411.2738.

[10] A. A. Maarif, "Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah," *Dok. Karya Ilm. | Tugas Akhir | Progr. Stud. Tek. Inform. - S1 | Fak. Ilmu Komput. | Univ. Dian Nuswantoro Semarang*, no. 5, p. 4, 2015, [Online]. Available: mahasiswa.dinus.ac.id/docs/skripsi/jurnal/15309.pdf.

[11] A. Novantirani, M. K. Sabariah, and V. Effendy, "Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine," *e-Proceeeding Eng.*, vol. 2, no. 1, pp. 1–7, 2015.

[12] M. A. F. Prananda Antinasari, Rizal Setya Perdana, "Analisis Sentimen Tentang Opini Film Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dengan Perbaikan Kata Tidak Baku," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1733–1741, 2017, [Online]. Available: http://j-ptiik.ub.ac.id.

[13] I. T. S. Utilization, "Sentiment Analysis Using Random Forest Algorithm-," vol. 2, no. 2, pp. 29–33, 2019.

[14] R. Ferdiana, F. Jatmiko, D. D. Purwanti, A. S. T. Ayu, and W. F. Dicka, "Dataset Indonesia untuk Analisis Sentimen," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 8, no. 4, p. 334, 2019, doi: 10.22146/jnteti.v8i4.533.