Klasifikasi Topik Twitter menggunakan Metode Random Forest dan Fitur Ekspansi Word2Vec

Rafly Ghazali Ramli[1], Yuliant Sibaroni[2]
[1,2] Universitas Telkom, Bandung
raflyramli@student.telkomuniversity.ac.id[1], yuliantsibaroni@telkomuniversity.ac.id[2]

Abstrak

Pengguna social media Twitter biasanya hanya tertarik pada *tweet* yang termasuk dalam jenis topik tertentu. *Tweet* yang hanya memuat tidak lebih dari 140 karakter, membuat klasifikasi *tweet* menjadi banyak tantangan, karena *tweet* yang pendek, *noice*, dan kurang fokus pada topik. Solusi untuk menyelesaikan tantangan tersebut dalam penelitian ini menggunakan fitur ekspansi agar memperkaya teks sehingga tampak seperti dokumen teks berukuran besar. Metode yang dipilih pada fitur ekspansi adalah *Word2Vec*, untuk mengelompokkan vektor dari kata-kata yang mirip menjadi satu di dalam ruang vektor, artinya mendeteksi kemiripan secara matematis. Penulis menggunakan metode *Random Forest* untuk klasifikasi data *tweet* pada penelitian ini, karena terkenal karena menjaga ketidak seimbangan data di kelas yang berbeda, terutama kumpulan data yang sangat besar.

Kata kunci : *Tweet*, Fitur Ekspansi, Word2Vec, Random Forest

Abstract

Twitter social media users are usually only interested in *tweet*s that fall under certain types of topics. *Tweet*s that only contain no more than 140 characters, make classification of *tweet*s a lot of challenges, because *tweet*s are short, noisy, and less focused on the topic. The solution to solve these challenges in this study uses the expansion feature to enrich the text so that it looks like a large text document. The method chosen in the expansion feature is Word2Vec, to group vectors of similar words together in a vector space, meaning to detect similarity mathematically. The author uses the Random Forest method for the classification of *tweet* data in this study, because it is well known for maintaining data imbalance in different classes, especially very large data sets.

Kata kunci : *Tweet, Expansion Feature, Word2Vec, Random Forest*.

1. Pendahuluan

Latar Belakang

Sosial media Twitter dibatasi untuk memuat tidak lebih dari 140 karakter saat membuat *tweet*, sehingga tidak memberikan konteks yang cukup jelas untuk dikategorikan kedalam topik tertentu. Pengguna biasanya hanya tertarik pada *tweet* yang termasuk dalam jenis - jenis topik tertentu, sehingga pengguna sulit mengetahui informasi yang baik dari topik yang diinginkan, karena terdapat *tweet* yang tidak masuk kedalam kategori dan topik tertentu, maka saat ini mengidentifikasikan *tweet* dengan topik yang tepat sangat penting dan perlu. Mengklasifikasikan sekumpulan *tweet* menimbulkan serangkaian masalah baru, karena *tweet* yang pendek, noise, dan kurang fokus terhadap topik[1]. Manfaat dilakukannya klasifikasi topik *tweet* pada penelitian ini agar *tweet* yang tidak memiliki konteks dan tidak dapat diketahui topiknya akan diklasifikasikan ke dalam kategori umum yang sudah ada[2], sehingga kegunaan untuk pengguna agar pemahaman topik yang lebih mudah dan pencarian informasi yang lebih baik. Oleh karena itu tujuan penelitian ini untuk mengurangi masalah klasifikasi teks pendek pada topik Twitter dengan memperluas fitur dengan menggunakan *Word2Vec*, karena teknologi klasifikasi teks pendek sangat penting dalam pemeriksaan spam[3],mesin pencari[4], dan lainnya.

Fitur ekspansi adalah proses memperkaya atau memperluas teks asli dengan tambahan semantik agar tampak seperti dokumen teks berukuran besar[5]. Metode fitur ekspansi, banyak digunakan karena dapat meningkatkan ketersebaran teks, seperti model LDA (*Latent Dirichlet Alloction*), LF-LDA (*Latent Feature-LDA*) dan *Word2vec*[6]. Dibandingkan dengan metode tradisional, model *Word2vec* dapat memecahkan

masalah klasifikasi teks pendek dengan lebih baik berdasarkan informasi konteks[6]. Tujuan dan kegunaan *Word2vec* adalah untuk mengelompokkan vektor dari kata-kata yang mirip menjadi satu di dalam ruang vektor. Artinya, ia mendeteksi kemiripan secara matematis. *Word2vec* adalah model yang membuat vektor yang merepresentasi numerik dari setiap kata[7].

Banyak algoritma pembelajaran seperti, *Rocchio Classifier*, *Support Vevtor Machine*, *K-Nearest Neighbors*, *Naive Bayes*, *Maximum Entropy*, dan *Random Forest*, telah diterapkan pada banyak masalah klasifikasi teks, dan mencapai hasil yang baik[1], Klasifikasi *Random Forest* adalah salah satu algoritma *machine learning* yang paling terkenal dan efektif dalam data *mining* dan khususnya dalam klasifikasi teks. Klasifikasi mereka didasarkan pada suara terbanyak dari klasifikasi *decision tree*. Nilai masing-masing antar atribut dari kumpulan suatu data merupakan asumsi utama dan poin kunci untuk membuat suatu prediksi[8]. *Random Forest* dapat dikaitkan dengan kemampuannya menangani tugas klasifikasi secara efisien. *Random forest* menjadi klasifikasi terkenal karena menjaga ketidak seimbangan data di berbagai kelas[9],[10] terutama untuk kumpulan data yang berjumlah besar[11]. Karena arsitektur paralelnya, pengklasifikasi *random forest* lebih cepat dibandingkan dengan pengklasifikasi lainnya[12].

Penelitian sebelumnya setiawan dkk.[7] telah melakukan perluasan fitur menggunakan pengklasifikasi *Naïve Bayes*, *SVM*, dan *Logistic Regression* yang menggunakan fitur ekspansi *word2vec*, terbukti dapat mengatasi masalah pada ketidak cocokan kosakata secara konsisten pada beberapa metode pengklasifikasian. Tujuan pada penelitian kali ini adalah bagaimana mengimpelemtasikan, mengukur nilai performansi khususnya akurasi dan *F1 – Score*, serta menganalisis hasil sistem klasifikasi topik pada Twitter menggunakan metode *Random Forest* yang digabungkan dengan fitur ekspansi *Word2Vec*. Dan penelitian ini menggunakan metode *word embedding Word2vec* yang diperuntukan dokumen berbasis teks, sebelum melakukan klasifikasi. Sebuah penelitian juga menunjukkan bahwa *word2vec* membantu secara dramatis meningkatkan kinerja klasifikasi teks dan pengelompokan suatu teks[13][14].

2. Studi Terkait

Ombabi, dkk mengatakan, faktanya, Twitter yang merupakan jejaring sosial populer, setiap hari Twitter menghasilkan lebih dari 500 juta *tweet*, sehingga dapat digunakan *tweet* ini untuk menemukan topik minat mereka[15]. Castillo, dkk meneliti kredibilitas informasi yang menyebar menggunakan *Twitter* karena *Twitter* memfasilitasi penyebaran informasi secara *real-time*[16].

Ombabi, dkk mengatakan, klasifikasi digunakan untuk menentukan topik mana yang sesuai dengan *tweet* tertentu, sehingga menemukan topik minat pengguna[15]. Menurut[17], kinerja klasifikasi *Random Forest* meningkat dengan bertambahnya jumlah pohon. Schenbly, dkk meneliti sistem klasifikasi dengan menerapkan *Random Forest* untuk menyingkirkan bot dari informasi palsu. Penggunaan *Random Forest* karena kemampuannya untuk mencegah *overfitting*. Penelitian ini hasilnya dapat mempertahankan keakuratan 90% pada data baru[18].

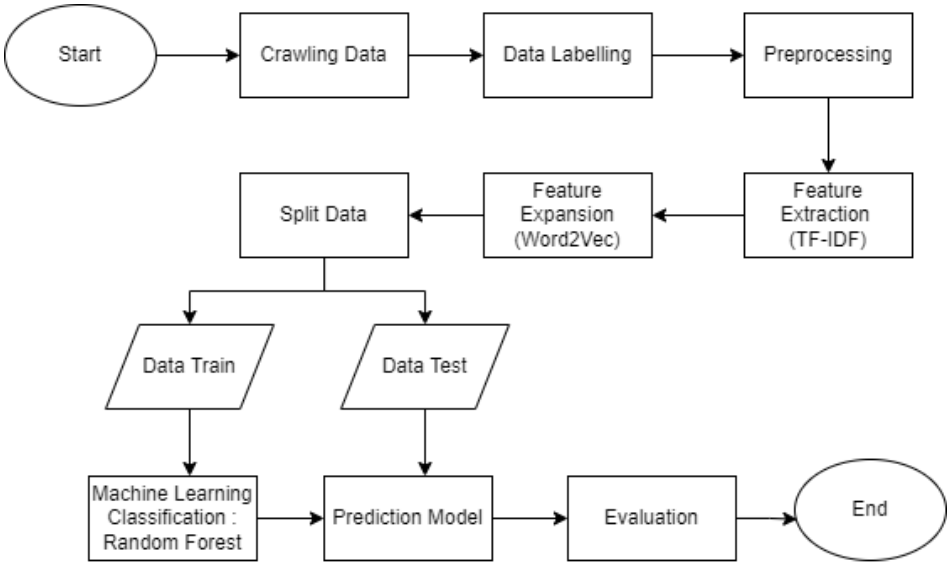
Beberapa penelitian yang membahas fitur ekspansi salah satunya, Setiawan, dkk meneliti ekspansi fitur dapat mengurangi ketidak cocokan kosakata dengan fitur "*word embeddings*", menggunakan pengklasifikasi Naïve Bayes, SVM, dan Logistic Regression yang menggunakan fitur ekspansi *word2vec*[7]. Sun, dkk, Mengatakan penggunaan fitur ekspansi dapat meningkatkan ketersebaran teks, sehingga banyak digunakan, seperti model LDA (*Latent Dirichlet Alloction*), LF-LDA (*Latent Feature-LDA*) dan *Word2vec*[6]. Giatsoglou, dkk membuktikan melatih model vektor kata dalam Bahasa Yunani dan Bahasa Inggris menggunakan metode *Word2vec*, kemudian model vektor digunakan dalam pelatihan model *classifier* yang menggunakan algoritma *Support Vector Machine* (SVM), menunjukan peningkatan dalam hal akurasi sebesar 1,6% dan 10,2%[19].

Setelah memahami beberapa penelitian, penggunaan ekspansi fitur berbasis *word embedding Word2Vec* dalam topik *twitter* dapat mengatasi masalah pada ketidak cocokan kosakata pada metode klasifikasi, dan belum ada penggunaan ekspansi fitur berbasis *Word2Vec* dalam klasifikasi topik *twitter* dengan algoritma *Random Forest.* Maka daripada itu pada penelitian ini menggabungkan fitur ekspansi *Word2Vec* dengan algoritma *Random Forest* tersebut.

3. Sistem yang Dibangun

Sistem Klasifikasi Topik yang akan dibangun pada penelitian ini dapat dilihat pada Gambar 1.

3.1. Gambaran Sistem



Gambar 1 Sistem Klasifikasi Topik Menggunakan *Feature Expansion Word2Vec*

3.2. Crawling Data

Dataset yang dibangun pada penelitian ini dilakukan dengan cara crawling data yang bersumber dari Twitter menggunakan Application Program Interface (API). API Twitter hanya bisa diakses melalui permintaan otentikasi. Twitter memperbolehkan membuka atau mengakses (OAuth) dan permintaannya sesuai protocol via pengguna Twitter yang sah. Data yang terkumpul kurang lebih 18,750 *tweet* dengan kata kunci dan topik yang berbeda, seperti Tabel 1.

Tabel 1 *Keyword* Setiap Kategori

No	Topik	Keyword
1	Agama	Doa, Islam, Hindu, Budha
2	Hiburan	Artis, Game, Film, Musik, Pesta
3	Olahraga	Basket, Berenang, Bulutangkis, Sepakbola, Lari
4	Pendidikan	Kelas, Kuliah, PTM, Kurikulum, Sekolah
5	Politik	Politik, Kampanye, Partai, Demokrasi

Dikarenakan kamus kata dataset *tweet* hanya memuat kurang lebih 6000 kosakata maka untuk melihat apakah nilai akurasi sudah lebih akurat atau tidak, maka akan digunakan data kamus kata tambahan menggunakan data yang diambil dari beberapa media berita seperti CNNIndonesia, Kompas, Detik, Liputan6, dan lainnya. Dan seperti Tabel 2 data yang terkumpul dari kamus kata tersebut berjumlah 142.544, dan untuk

pengambilannya dari tanggal 01 Mei 2016 hingga tanggal 01 Maret 2017, kamus tambahan ini akan digunakan dalam penelitian untuk pembuatan kamus *similarity* pada tahap fitur ekspansi yang nantinya akan dibandingkan dengan kamus dari dataset *tweet* yang sudah dibuat sebelumnya.

Tabel 2 Persebaran Data Kamus Berita

Sumber	Jumlah
CNN Indonesia	29349
Detik	7974
Kompas	15055
Liputan6	251
Republika	53812
SIndoNews	22401
Tempo	13702
Total	**142.544**

3.3. Data Labelling

Pada penelitian ini topik yang digunakan hanya menggunakan 5 topik yaitu, Politik, Pendidikan Hiburan, Agama dan, Olahraga, pada proses pelabelan ini dilakukan pelabelan kelas dari 18750 *tweet* data yang sudah dikumpulkan agar data tersebut memiliki label yang benar, sehingga menjadikan data yang mempunyai sebuah label kelas. Pelabelan dilakukan oleh 2 orang Mahasiswa, dan setiap mahasiswa melabeli 3000 data sesuai topik dari 18750 data *tweet.* hasil pelabelan dari kedua mahasiswa yang sama akan dijadikan dataset, sedangkan pelabelan data yang berbeda akan ditinjau ulang dan dicari *tweet* yang sesuai dengan label yang sudah ada. Dan data yang sudah dilabeli tersebut akan dimasukan kedalam format CSV. Untuk contoh pelabelan terdapat pada Tabel 4, dan untuk ketersebaran data *tweet* dapat dilihat pada Tabel 3.

Tabel 3 Kategori dan Jumlah Dataset

Label	Jumlah
Politik	600
Pendidikan	600
Hiburan	600
Agama	600
Olahraga	600
Total	**3000**

Tabel 4 Contoh Pelabelan Data

Tweet	Label
Jantung Christian Eriksen dikabarkan akan dipasang alat ICD yang berfungsi untuk mengembalikan ritme jantung dan dapatmemonitori langsung detak jantung Eriksen.	Olahraga
Ptm untuk perkuliahan akan dimulai dalam beberapa min ggu lagi	Pendidikan
Merekalah para perusak demokrasi yang memaksakan ag ama dalam demokrasi	Politik
Lagu itu enaknya dinikmati sendiri Pake earphone, tanpa ada orang lain yang interupsi selera lagu kita.	Hiburan
Doa sepertiga malamku untukmu. Bismillah ya Allah, izi nkanlah kami bersatu jika memang kau menghendaknya. Aamiin.	Agama

3.4. Preprocessing

Preprocessing data bertujuan untuk mengurangi ukuran data, menemukan hubungan antar data, menormalkan data, menghilangkan *outlier* (data yang memiliki karakteristi unik) dan mengekstrak fitur untuk data[20]. *Preprocessing* dilakukan dengan tujuan memastikan dataset siap untuk diproses. Berikut adalah tahapan dalam *preprocessing* :

1. *Cleaning*

   Pada proses ini dilakukan pembersihan dari karakter – karakter yang tidak diperlukan, tanda baca, emoticon dan angka – angka yang terdapat pada dataset. Proses ini dilakukan dengan menggunakan *Library re* (*regular expression operations*) *reguler expression*, untuk memeriksa apakah string tertentu cocok dengan *reguler expression* yang diberikan.
2. *Case Folding*

   Case Folding adalah tahap mengubah semua huruf dalam dokumen menjadi huruf kecil, hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf akan dihilangkan.
3. Normalisasi Kata

   Proses ini mengubah kata-kata yang disingkat, kata yang salah dalam penulisannya (typo), kata gaul dan kata yang tidak formal menjadi kata yang formal dengan bantuan kamus yang terdapat dari *github*.
4. *Filtering*

   Filtering adalah tahap membuang kata-kata yang sering muncul dan bersifat umum, kurang menunjukan relevansinya dengan teks. Proses ini akan menghilangkan kata-kata yang sering muncul namun tidak memiliki pengaruh apapun. Proses ini dilakukan dengan menggunakan *Library NLTK*.
5. *Stemming*

   Stemming adalah tahap mengganti suatu bentuk kata menjadi kata dasar. Hasil dari proses text mining tersebut mendeskripsikan poin poin utama dalam suatu dokumen, sehingga selanjutnya dapat digunakan untuk proses pengelompokan atau clusterization. Proses ini dilakukan dengan menggunakan *Library Sastrawi*.
6. *Tokenizing*

   Tokenizing adalah tahap pemotongan dokumen teks bedasarkan tiap kata yang menyusunnya. Potongan kata tersebut disebut dengan token atau term.

3.5. Feature Extraction (TF-IDF)

TF-IDF adalah statistik numerik yang menunjukkan relevansi kata kunci dengan beberapa dokumen tertentu atau dapat dikatakan, menyediakan kata kunci tersebut, yang dengannya beberapa dokumen tertentu dapat diidentifikasi atau dikategorikan[21].

TF-IDF merupakan gabungan dari dua kata yang berbeda yaitu *Term Frequency* dan *Inverse Document Frequency*. Pertama, istilah "*term frequency*" digunakan untuk mengukur berapa kali suatu *term* hadir dalam suatu dokumen, misalkan memiliki dokumen yang berisi 5000 kata dan kata dan kata “alpha” muncul dalam dokumen sebanyak 10 kali maka nilai *term frequency* dapat dihitung dengan rumus.

$$TF = \text{jumlah frekuensi kata terpilih} / \text{jumlah kata}$$

$$TF = \frac{10}{5000} = 0002$$

Inverse Document Frequency berguna ketika frekuensi suatu dokumen dihitung, dapat diamati bahwa algoritma memperlakukan semua kata kunci secara setara, tidak masalah jika itu adalah kata berhenti. *IDF* memberikan bobot yang lebih rendah untuk kata-kata yang sering dan memberikan bobot yang lebih besar untuk

kata-kata yang jarang. Misalnya dalam 10 dokumen dan istilah “teknologi” mucul di 5 dokumen tersebut nilai *IDF* dapat dihitung dengan rumus [21].

$$IDF = \log(\text{jumlah dokumen}/\text{jumlah frekuensi kata terpilih}) 10$$

$$IDF = \log\left(\frac{}{5}\right) = 03010$$

Dapat dipahami bahwa, semakin besar atau tinggi kemunculan kata dalam dokumen akan memberikan frekuensi istilah yang lebih tinggi dan semakin sedikit kemunculan kata dalam dokumen akan menghasilkan tingkat kepentingan yang lebih tinggi (*IDF*) untuk kata kunci yang dicari dalam dokumen tertentu, *TF-IDF* hanya perkalian *term frequency* (*TF*) dan *inverse document frequency* (*IDF*). Untuk menghitung *TF-IDF* dapat dilakukan dengan perhitungan seperti berikut[21].

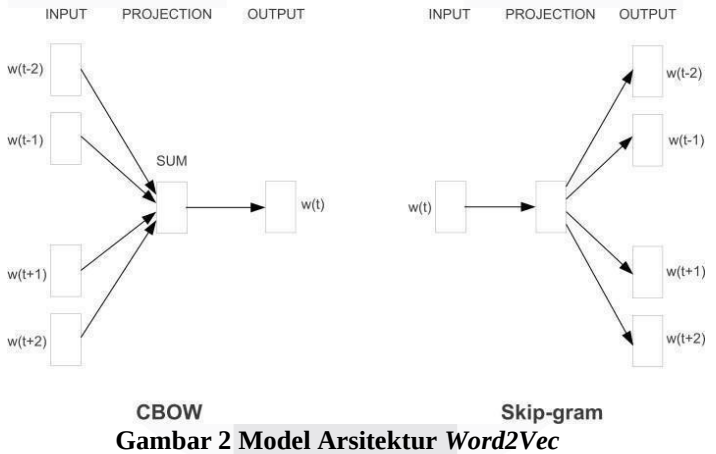
$$TF - IDF = TF \times IDF$$

$$TF - IDF = 0002 \times 03010 = 0000602$$

3.6. Feature Expansion (Word2Vec)

Word2Vec didasarkan pada ide deep learning, di mana kata direpresentasikan dalam vektor. *Word2Vec* mentrasformasikan operasi dokumen menjadi perhitungan vektor dalam ruang vektor kata. Relasi semantik pada dokumen dapat dikarakterisasi berdasarkan kesamaan kata di dalam ruang vektor. Tahap awal pada proses *word2vec* yaitu membangun kosakata dari data teks pelatihan dan kemudian mempelajari representasi vektor dari kumpulan kata. Vektor yang dihasilkan dapat digunakan sebagai fitur untuk penerapan dalam kasus *natural language processing* dan *machine learning*[22].

Word2vec adalah alat yang dirilis oleh Google pada tahun 2013. Alat ini mengadopsi dua arsitektur model utama, model bag-of-words (CBOW) berkelanjutan dan model skip-gram berkelanjutan, untuk mempelajari representasi vektor dari kata-kata[23], dua model arsitektur tersebut dapat dilihat pada Gambar 2.



Gambar 2 Model Arsitektur *Word2Vec*

Word2Vec menyediakan dua langkah untuk mendapatkan kata-kata yang mirip. Langkah pertama menggunakan kata-kata tetangga untuk memprediksi target kata (metode yang dikenal sebagai *continuous bag of words*, atau *CBOW*), dan langkah kedua menggunakan kata untuk memprediksi kata-kata tetangga dalam sebuah kalimat, yang disebut *skip-gram*[24]. Arsitektur *CBOW* memprediksi kata saat ini berdasarkan konteksnya, dan *skip- gram* memprediksi kata-kata di sekitarnya berdasarkan arus kata[24].*Word2Vec* dapat mempelajari representasi vektor dari kata-kata dalam ruang vektor berdimensi tinggi dan menghitung cosinus jarak antar kata. Artinya, alat tersebut dapat menemukan hubungan semantik antara kata-kata dalam dokumen[23]. *Word2Vec* juga dapat bekerja secara efektif meskipun inputnya adalah kata individual. Dengan alat ini, prediksi yang sangat akurat tentang arti sebuah kata dapat diperoleh dan hubungan semantik antar kata dapat dengan mudah dievaluasi[15].

3.7. Random Forest

Metode Random Forest merupakan algoritma Ensemble Learning yang menggunakan dan membangun struktur Tree dalam tahapannya. Dalam penggunaannya, dibangun Decision Tree dengan memilih atau mengambil data secara acak. Untuk menentukan kelas suatu data, dalam Random Forest menggunakan sistem voting dari hasil berdasarkan Decision Tree tersebut[25].

Random Forest (RF) merupakan metode yang dapat meningkatkan hasil akurasi, karena dalam mengembangkan simpul untuk setiap node yang dilakukan secara acak. Metode ini digunakan untuk membangun Decision Tree yang terdiri dari root node, internal node, dan leaf node dengan mengambil atribut dan data secara acak sesuai ketentuan yang diberlakukan[25].

3.8. Prediction Model (Confusion Matrix)

Pengukuran performansi sistem dari model yang akan dibuat yaitu menggunakan *Confusion Matrix*. *Confusion matrix* adalah konsep dari machine learning, yang berisi informasi tentang klasifikasi aktual dan prediksi yang dilakukan oleh sistem klasifikasi. *Confusion matrix* memiliki dua dimensi, satu dimensi diindeks oleh kelas sebenarnya dari suatu objek, yang lain diindeks oleh kelas yang diprediksi oleh pengklasifikasi[26], dapat dilihat pada Tabel 5 bentuk *confusion matrix*.

Tabel 5 *Confusion Matrix*

		Predicted	
		Negative	Positive
Actual	Negative	TN (True Negative)	FP (False Positive)
	Positive	FN (False Negative)	TP (True Positive)

Sejumlah ukuran kinerja klasifikasi dapat didefinisikan berdasarkan *Confusion matrix*. Beberapa ukuran umum yang pasti diberikan sebagai berikut[26].

Keterangan:

TP = Jumlah data positif dan diprediksi benar.

TN = Jumlah data negatif yang diprediksi benar

FP = Jumlah data negatif namun diprediksi data positif.

FN = Jumlah data positif namun diprediksi data negatif

a. Accuracy

Akurasi adalah proporsi dari jumlah total prediksi yang benar :

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

b. Precision

Presisi adalah ukuran akurasi yang dimana kelas tertentu telah diprediksi terlebih dahulu. Ini didefinisikan oleh :

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

c. Recall

Recall adalah ukuran kemampuan model prediksi untuk memilih contoh kelas tertentu dari kumpulan data. Didefinisikan oleh rumus :

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

d. F1-Measure

F1-Measure adalah perbandingan rata-rata presisi dan recall. Didefinisikan oleh :

$$F1 - Measure = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$

4. Evaluasi

Bagian ini menjelaskan hasil yang telah dilakukan pada *system* yang sudah dibangun.

Tabel 6 Tahap *Preprocessing*

Preprocessing	**Sebelum**	**Sesudah**
Cleaning	Terutama media sekarang, banyak yang bungkam tentang kebobrokan Demokrasi sekarang salut buat yang sudah mau bersuara.	Terutama media sekarang banyak yang bungkam tentang kebobrokan Demokrasi sekarang salut buat yang sudah mau bersuara
Case Folding	Terutama media sekarang banyak yang bungkam tentang kebobrokan Demokrasi sekarang salut buat yang sudah mau bersuara	terutama media sekarang banyak yang bungkam tentang kebobrokan demokrasi sekarang salut buat yang sudah mau bersuara
Normalisasi	terutama media sekarang banyak yang bungkam tentang kebobrokan demokrasi sekarang salut buat yang sudah mau bersuara	terutama media sekarang banyak yang bungkam tentang kebobrokan demokrasi sekarang salut buat yang sudah mau bersuara
Filtering	terutama media sekarang banyak yang bungkam tentang kebobrokan demokrasi sekarang salut buat yang sudah mau bersuara	media bungkam kebobrokan demokrasi salut bersuara
Stemming	media bungkam kebobrokan demokrasi salut bersuara	media bungkam kebobrokan demokrasi salut bersuara
Tokenizing	media bungkam kebobrokan demokrasi salut bersuara	['media', 'bungkam', 'bobrok', 'demokrasi', 'salut', 'suara']

4.1. Preprocessing Data

Dapat dilihat pada Tabel 6, *preprocessing* dilakukan dengan tujuan memastikan dataset siap untuk diproses.

4.2. Pembuatan Kamus Kata *Word2Vec (Corpus)*

Pada pembuatan *corpus* kamus kata digunakan teknik *word embedding Word2Vec* model *skip-gram. Corpus* sendiri merupakan kumpulan kata yang diurutkan melalui kata yang nilai similaritasnya tertinggi hingga terendah, nilai similaritas digunakan untuk mempermudah melihat nilai nilai kedekatan kata. Sehingga didapatkan hasil yang ditunjukan dari tabel berikut.

1. *Corpus* data *Tweet*
   Pada *Corpus* data *Tweet* didapatkan hasil kosakata sebanyak 6.010 dan pada Tabel 7 berikut dapat dilihat contoh nilai dengan besar urutan similaritasnya, sedangkan kolom Rank 1 sampai dengan Rank 10 menyatakan derajat kedekatan dengan "partai".

Tabel 7 Contoh *Corpus Tweet*

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
partai	orang	islam	indonesia	film	politik
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	ya	renang	agama	kampanye	sepakbola

2. *Corpus* Data Berita
   Pada *Corpus* data *Tweet* didapatkan hasil kosakata sebanyak 225.932 dan pada Tabel 8 berikut dapat dilihat contoh nilai dengan besar urutan similaritasnya, sedangkan kolom Rank 1 sampai dengan Rank 10 menyatakan derajat kedekatan dengan "pilkada".

Tabel 8 Contoh *Corpus* Berita

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
pilkada	pemilukada	pilgub	milu	pilkades	serentak
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	pilada	pilpres	pikada	pileg	kontestasi

3. *Corpus* Data Berita dan *Tweet*
   Pada *Corpus* data *Tweet* didapatkan hasil kosakata sebanyak 226.837 dan pada Tabel 9 berikut dapat dilihat contoh nilai dengan besar urutan similaritasnya, sedangkan kolom Rank 1 sampai dengan Rank 10 menyatakan derajat kedekatan dengan "basket".

Tabel 9 Contoh *Corpus* Berita dan *Tweet*

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
basket	sepak	sepakbola	voli	antarmedia	futsal
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	rugbi	rugby	baseball	invistasi	bisbol

4.3. Feature Expansion

Seperti pada Tabel 10 output dari pembuatan corpus menggunakan Word2Vec berupa kata-kata yang mirip, untuk rank 1 sampai dengan rank 10 merupakan kata yang mirip beserta nilai similaritas dengan kata

"basket", dan nilai similaritas ini hanya mewakili kata terdekat dari suatu kata, sedangkan pembobotan suatu kata ditambahkan pada proses TF-IDF dan *feature expansion*. Sebagai ilustrasi untuk eksperimen *feature expansion* dengan top *similarity* 10 menggunakan *Corpus* kamus kata gabungan, semisal diberikan contoh *tweet* :"…orang itu sepak sebuah bola". Pada representasi vektor fitur TF-IDF kata "basket" memiliki bobot kata nol. Namun, pada dokumen, *tweet* tersebut memiliki kata "sepak", dikarenakan "sepak" ada pada fitur top *similarity* 10 kata "basket" maka kamus kata "basket" tersebut bernilai bobot sama seperti kata "sepak".

Tabel 10 *Corpus* data Gabungan dan Nilai Similaritas

Kata	**Rank 1**	**Rank 2**	**Rank 3**	**Rank 4**	**Rank 5**
	sepak	sepakbola	voli	antarmedia	futsal
basket	0.7961	0.7824	0.7593	0.7532	0.7512
	Rank 6	**Rank 7**	**Rank 8**	**Rank 9**	**Rank 10**
	rugbi	rugby	baseball	invistasi	bisbol
	0.7414	0.7409	0.7398	0.7317	0.7281

Pada penelitian ini keterhubungan proses TF-IDF, dan fitur ekspansi adalah, TF-IDF digunakan untuk mengukur seberapa penting sebuah kata dalam korpus dokumen, sehingga pada proses TF-IDF memberikan bobot awal pada dokumen dalam bentuk vektor, yang nantinya pada tahap fitur ekspansi akan memberikan nilai dari bentuk vektor TF-IDF yang memiliki bobot 0, seperti pada contoh ilustrasi Tabel 11 di bawah fitur kata "basket" memiliki nilai bobot 0 dan fitur kata "sepak" memiliki nilai bobot 0.65, maka nilai bobot "basket" tersebut akan digantikan dengan nilai bobot dari kata "sepak" sehingga nilai fitur kata "basket" menjadi 0.65, kedua proses ini dilakukan pada semua dataset yang digunakan, yang nantinya hasil proses pembobotan ini akan dibagi menjadi data train dan data test yang digunakan pada tahap klasifikasi.

Tabel 11 Nilai Pembobotan *TF-IDF*

Kata					
	sepak	sepakbola	voli	antarmedia	futsal
basket	0.65	0.58	0.38	0.53	0.61
0	rugbi	rugby	baseball	invistasi	bisbol
	0.51	0.29	0.64	0.25	0.44

Bobot yang digunakan pada tahap fitur ekspansi ini sendiri merupakan hasil dari pemberian nilai pada proses TF-IDF untuk fitur yang bernilai 0, yang artinya pada tahap TF-IDF fitur tersebut tidak terlalu berkaitan dengan topik atau tidak terlalu penting pada dokumen, sedangkan bobot yang bernilai lebih dari 0 maka fitur tersebut penting dalam suatu dokumen, ilustrasi perhitungan TF-IDF dapat dilihat pada tabel 12.

Tabel 12 Ilustrasi Pembobotan *TF-IDF*

	TF			TFxIDF	
Kata	A	B	IDF	A	B
bola	1/7	1/7	Log(2/2) = 0	0	0
voli	1/7	0	Log(2/1) = 0.3	0.043	0
sepak	0	1/7	Log(2/1) – 0.3	0	0.043

4.4. Klasifikasi

Proses klasifikasi ini dilakukan seletah melalui tahap preprocessing, pembobotan kata, dan proses Feature Expansion, kemudian proses dilanjutkan ke tahap klasifikasi menggunakan klasifikasi Random Forest. Sebelum dilakukan proses klasifikasi system, akan dilakukan penggantian rasio data latih dan data uji terlebih dahulu agar hasil yang diberikan lebih optimal. Pada 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

Pengujian ini dilakukan sebelum melakukan perbandingan dengan hasil penggabungan fitur ekspansi, pada proses ini akan memperlihatkan hasi dari *baseline Random Forest* yang dibandingkan dengan pengujian menggunakan *Random Forest* yang sudah menggunakan pembobotan *tf-idf* dengan menggunakan ukuran fitur 1000, ukuran fitur 1000 sendiri merupakan hasil pembobotan fitur yang nilai *term-frequency* nya tertinggi dan sering muncul pada proses *tf-idf*, penggunaan 1000 fitur ini akan digunakan pada tahap pengujian penelitian ini, yang masing-masing sistem dilakukan pengulangan sebanyak 3 kali percobaan untuk mengambil nilai rata-rata akurasinya dengan rasio data latih dan data uji sebesar 80:20. Untuk hasil nya dapat dilihat pada Tabel 13.

Tabel 13 Hasil Performansi *Random Forest*

Classifier	**Akurasi (%)**	***F1-Score***
Baseline (*Random Forest*)	98.44	0.9842
Baseline (*Random Forest*) + *TF-IDF*	**98.60 (+0.16)**	**0.9859 (+0.0017)**

Dari Tabel tersebut dapat disimpulkan bahwa dengan rasio 20% data uji nilai akurasi rata-rata dari hasil klasifikasi memiliki nilai yang lebih baik saat menggunakan *feature extraction TF-IDF* dibandingkan dengan klasifikasi baseline yang hanya menggunakan *Random Forest.*

Pada pengujian berikutnya yaitu membandingkan hasil ketika menggunakan fitur ekspansi *Word2Vec* dengan kamus kata data *tweet*, kamus kata data berita, dan gabungan dari *tweet* dan kamus data berita. Pengujian kali ini menggunakan data latih dan data uji dengan perbandingan ratio 80:20. Seperti pada penelitain sebelumnya Erwin dkk[7], untuk melakukan pengujian fitur ekspansi dilakukan pengujian menggunakan *top similarity*, untuk mengetahui hasil *top similarity* dilakukan seperti penelitain sebelumnya tiap pengambilan fitur top 1, 5, 10[7], yang mempunyai similaritas tertinggi atau *similarity* dari kamus kata yang telah dibuat. Penggunaan top 1, 5, 10 pada penelitian ini untuk melihat model *top similarity* berapa yang lebih baik. Jika menggunakan top 1 maka kata yang digunakan hanya menggunakan kata dari kolom rank 1 atau kata paling mirip pertama, jika menggunakan top 5 maka kata yang digunakan menggunakan kata kolom rank 1 – 5, begitu juga dengan top 10 maka kata yang digunakan menggunakan kata kolom rank 1 – 10 untuk pencarian

bobot pada saat fitur ekspansi, dan jika ada kata yang nilai pembobotan sama maka tetap akan digunakan kata yang sebelumnya. Pengujian dilakukan sebanyak 3 kali untuk mengambil nilai rata-rata dari hasil akurasinya.

Hasil pengujian perbandingan menggunakan fitur ekspansi dengan klasifiaksi *Random Forest* dapat dilihat pada Tabel 14 dan Tabel 15.

Tabel 14 Hasil Akurasi *Feature Expansion* Pada *Random Forest*

Top Similarity	Akurasi (%)			
	Baseline	*Corpus Tweet*	*Corpus* Berita	*Corpus* Berita + *Tweet*
Top 1	98.44	99,16 (+0,72)	99,22 (+0,78)	99.27 (+0,83)
Top 5	98.44	99,38 (+0,94)	99,44 (+1,00)	**99,49 (+1,05)**
Top 10	98.44	99,05 (+0,61)	99,11 (+0,67)	98,82 (+0,38)

Tabel 15 Hasil *F1-Score Feature Expansion* Pada *Random Forest*

Top Similarity	*F1-Score*			
	Baseline	*Corpus Tweet*	*Corpus* Berita	*Corpus* Berita + *Tweet*
Top 1	0.9842	0,9913 (+0,0071)	0,9921 (+0,0079)	0,9927 (+0,0085)
Top 5	0.9842	0.9938 (+0,0096)	0.9944 (+0,0102)	**0,9949 (+0,0107)**
Top 10	0.9842	0,9905 (+0,0063)	0.9911 (+0,0069)	0,9882 (+0,0004)

Hasil dari nilai akurasi dan *F1-Score* pada fitur ekspansi menggunakan algoritma Random Forest menunjukan bahwa hasil dari keseluruhan nilai mengalami peningkatan sehingga nilai tertingginya ada pada fitur top 5 dengan menggunakan kamus kata data berita+ *tweet* sebesar 99,49% dan untuk hasil *F1-Score* terjadi peningkatan dengan nilai tertinggi ada di fitur top 5 dengan menggunakan kamus kata data berita+ *tweet* sebesar 0,9949 dengan adanya peningkatan dengan nilai baseline yang sebelumnya 98.44% untuk akurasi dan *F1-Score* 0.9842.

4.6. Analisis Hasil Pengujian

Hasil dari pengujian percobaan menggunakan *Feature Expansion* dengan penggunaan kamus kata dan ukuran fitur yang berbeda, mendapatkan hasil yang juga berbeda-beda. 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 dibangun sistem klasifikasi topik pada *tweet* menggunakan ekspansi fitur metode *Word2Vec* dengan metode klasifikasi *Random Forest*. Ekspansi Fitur *Word2Vec* digunakan pada sistem klasifikasi ini bertujuan untuk mengurangi ketidak cocokan kosakata pada kalimat *tweet.* Ekspansi fitur dilakukan dengan menggunakan 3 *copus Word2Vec* (*tweet,* berita dan gabungan antara *tweet* dan berita) dan juga 3 variasi ekspansi fitur (Top 1, Top 5 dan, Top 10) untuk mencari model terbaik, sehingga mendapatkan model terbaik menggunakan fitur top 5 dengan nilai akurasi 99,49% dan nilai *F1-Score* yang juga meningkat pada model fitur top 5 sebesar 0,9949 menggunakan kamus kata gabungan dari data berita dan *tweet*. Pada penelitian ini, model ekspansi fitur ini berhasil meningkatkan nilai akurasi untuk metode klasifikasi *Random Forest* yang sebelumnya hanya mendapat nilai akurasi 98.44% dan nilai *F1-Score* sebesar 0.9842.

REFERENSI

[1] R. F. Quanzhi Li, Sameena Shah, Xiaomo Liu, Armineh Nourbakhsh, "Tweet Topic Classification Using Distributed Language Representations," pp. 1–12, 2015, doi: 10.1109/WI.2016.22.

[2] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary, "Twitter trending topic classification," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 251–258, 2011, doi: 10.1109/ICDMW.2011.171.

[3] M. Washha, A. Qaroush, M. Mezghani, and F. Sedes, "A Topic-Based Hidden Markov Model for Real-Time Spam Tweets Filtering," *Procedia Comput. Sci.*, vol. 112, pp. 833–843, 2017, doi: 10.1016/j.procs.2017.08.075.

[4] S. Schmidt, S. Schnitzer, and C. Rensing, "Text classification based filters for a domain-specific search engine," *Comput. Ind.*, vol. 78, pp. 70–79, 2016, doi: 10.1016/j.compind.2015.10.004.

[5] M. A. Fauzi, R. F. N. Firmansyah, and T. Afirianto, "Improving sentiment analysis of short informal Indonesian product reviews using synonym based feature expansion," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 3, pp. 1345–1350, 2018, doi:10.12928/TELKOMNIKA.v16i3.7751.

[6] F. Sun and H. Chen, "Feature extension for Chinese short text classification based on LDA and Word2vec," *Proc. 13th IEEE Conf. Ind. Electron. Appl. ICIEA 2018*, no. 1, pp. 1189–1194, 2018, doi: 10.1109/ICIEA.2018.8397890.

[7] 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.

[8] A. Bouaziz, C. Dartigues-Pallez, C. Da Costa Pereira, F. Precioso, and P. Lloret, "Short Text Classification Using Semantic Random Forest," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8646 LNCS, pp. 288–299, 2014, doi: 10.1007/978-3-319-10160-6_26.

[9] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse, "An empirical study of learning from imbalanced data using random forest," *Proc. - Int. Conf. Tools with Artif. Intell. ICTAI*, vol. 2, pp. 310–317, 2007, doi: 10.1109/ICTAI.2007.46.

[10] X. Y. Liu, J. Wu, and Z. H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 39, no. 2, pp. 539–550, 2009, doi:10.1109/TSMCB.2008.2007853.

[11] T. G. Dietterich and Oregon, "Ensemble Methods in Machine Learning," *Oncogene*, vol. 12, no. 2, pp. 265–275, 1996.

[12] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved Random Forest for Classification," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4012–4024, 2018, doi: 10.1109/TIP.2018.2834830.

[13] A. Handler, "An empirical study of semantic similarity in WordNet and Word2Vec A Thesis Submitted to the

Graduate Faculty of the University of New Orleans in partial fulfillment of the requirements for the degree of Master of Science in Computer Science by Abram Handl," vol. 2007, no. December, 2014.

[14] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning Sentiment-Specific Word Embedding," *Acl*, pp. 1555–1565, 2014.

[15] A. M. A. R.-L. Ombabi, Abubakr H, Onsa Lazzez, Wael Ouarda, "Deep Learning Framework based on Word2Vec and CNN for Users Interests Classification," 2017.

[16] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," *Proc. 20th Int. Conf. Companion World Wide Web, WWW 2011*, pp. 675–684, 2011, doi: 10.1145/1963405.1963500.

[17] L. BREIMAN, "Random Forests," 2001, doi: 10.1007/978-3-030-62008-0_35.

[18] J. Schnebly and S. Sengupta, "Random forest twitter bot classifier," *2019 IEEE 9th Annu. Comput. Commun. Work. Conf. CCWC 2019*, pp. 506–512, 2019, doi: 10.1109/CCWC.2019.8666593.

[19] M. Giatsoglou, M. G. Vozalis, K. Diamantaras, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Sentiment analysis leveraging emotions and word embeddings," *Expert Syst. Appl.*, vol. 69, pp. 214–224, 2017, doi: 10.1016/j.eswa.2016.10.043.

[20] S. A. Alasadi and W. S. Bhaya, "Review of data preprocessing techniques in data mining," *J. Eng. Appl. Sci.*, vol. 12, no. 16, pp. 4102–4107, 2017, doi: 10.3923/jeasci.2017.4102.4107.

[21] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 25–29, 2018, doi: 10.5120/ijca2018917395.

[22] R. Naous, M. Al-Shedivat, and K. N. Salama, "Stochasticity modeling in memristors," *IEEE Trans. Nanotechnol.*, vol. 15, no. 1, pp. 15–28, 2016, doi: 10.1109/TNANO.2015.2493960.

[23] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1857–1863, 2015, doi: 10.1016/j.eswa.2014.09.011.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[25] A. Wiraguna, S. Al Faraby, and Adiwijaya, "Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest," *e-Proceeding Eng.*, vol. 6, no. 1, pp. 2144–2153, 2019.

[26] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci. (Ny).*, vol. 340–341, pp. 250–261, 2016, doi: 10.1016/j.ins.2016.01.033.