Ekspansi Fitur dengan Word2Vec pada Klasifikasi Topik dengan Metode *Naive Bayes-Support Vector Machine* di Twitter

Muhammad Nazmi Al Malisi[1], Erwin Budi Setiawan[2]
[1,2] Universitas Telkom, Bandung
mnazmialmalisi@student.telkomuniversity.ac.id[1], erwinbudisetiawan@telkomuniversity.ac.id[2]

Abstrak

Pada layanan *microblogging* seperti Twitter, pengguna mungkin kesulitan dalam memahami tulisan dan topik yang ada, karena sering sekali ditemukan penyingkatan kata dalam setiap *tweet.* Solusi yang diterapkan pada penelitian ini adalah melakukan ekspansi fitur untuk topik pada *tweet.* Ekspansi fitur adalah proses memperkaya teks asli dengan tambahan semantik agar tampak seperti dokumen teks berukuran besar. Metode yang digunakan untuk ekspansi fitur adalah dengan menggunakan Word2Vec, Metode yang digunakan dalam melakukan klasifikasi adalah *Naive Bayes-Support Vector Machine*. Hasil penelitian menunjukkan bahwa sistem klasifikasi topik dengan ekspansi fitur Word2Vec menggunakan metode klasifikasi Naive Bayes-Support Vector Machine memiliki akurasi 84%.

Kata Kunci : ***ekspansi fitur, klasifikasi, nbsvm, tweet, twitter, word2vec***

Abstract

In microblogging services such as Twitter, users may find it difficult to understand the writings and topics, because abbreviations are often found in each tweet. The solution applied in this research is to expand features for topics in tweets. Feature expansion is the process of enriching the original text with additional semantics to make it appear like a large text document. The method used for feature expansion is to use Word2Vec. The method used in classifying is Naive Bayes-Support Vector Machine. The results showed that the topic classification system with Word2Vec feature expansion using the Naive Bayes-Support Vector Machine classification method had an accuracy 84%.

Keywords : ***feature expansion, classification, nbsvm, tweet, twitter, word2vec***

1. Pendahuluan

Pada layanan *microblogging* seperti Twitter terdapat batasan 140 karakter saja yang bisa disampaikan dalam satu kali *tweet*. Karena hal tersebut sering sekali ditemukan penyingkatan kata dalam setiap *tweet*[1]. Penggunaan variasi kata dapat meningkatkan kemungkinan ketidakcocokan kosakata dan membuat *tweet* sulit dipahami tanpa bantuan suatu topik[2]. Oleh karena itu penulis melakukan ekspansi fitur pada klasifikasi topik di Twitter untuk mengatasi permasalahan tersebut.

Ekspansi fitur adalah proses memperkaya teks asli dengan tambahan semantik agar tampak seperti dokumen teks berukuran besar[3]. Penggunaan ekspansi fitur umum digunakan pada *pattern recognition*[4][5] dan dalam konteks tertentu, seperti contohnya *text retrieval*[6] , *intrusion detection*[7], *sentiment analysis*[8], *text classification*[9] dan *topic classification*[1]. Terdapat beberapa metode untuk melakukan ekspansi fitur khususnya dalam klasifikasi teks dan klasifikasi topik, diantaranya terdapat LDA (*Latent Dirichlet Alloction)*[9], LF-LDA (*Latent Feature-LDA)*[9], Word2Vec[9][1] dan WordNet[10]. Pada Word2Vec terdapat dua algoritma pembelajaran utama, yaitu *skip-gram* dan *Continous bag-of-word* (CBOW) [1]. Dibandingkan dengan metode tradisional, model Word2vec dapat menyelesaikan klasifikasi teks pendek dengan lebih baik berdasarkan informasi konteks[9].

Metode klasifikasi tradisional, seperti: k-NN (k-Nearest Neighbours) [11], NB (Naive Bayes) [12], SVM (Support Vector Machine) [13], TF IDF[14] dapat mencapai efek yang memuaskan dalam bidang klasifikasi teks[9]. Varian *Naive Bayes* (NB) dan *Support Vector Machines* (SVM) sering digunakan sebagai metode dasar untuk klasifikasi teks, tetapi kinerjanya sangat bervariasi tergantung pada varian model, fitur yang digunakan dan tugas atau dataset[15]. *Naive Bayes-Support Vector Machine*(NBSVM) berkinerja baik pada *snippets* dan dokumen yang lebih panjang untuk klasifikasi topik dan seringkali lebih baik daripada hasil yang dipublikasikan sebelumnya. Oleh karena itu, NBSVM tampaknya menjadi dasar yang tepat dan sangat kuat untuk metode canggih yang bertujuan untuk menangani sekumpulan fitur[15].

Pemilihan sosial media Twitter sebagai domain penelitian ini karena karakter yang terbatas (140 karakter) pada setiap *tweet* nya dan sering sekali ditemukan penyingkatan kata yang terjadi pada setiap *tweet* yang bisa saja mengakibatkan informasi tidak dapat tersampaikan. Oleh karena itu penulis memilih Twitter sebagai domain untuk melakukan proses Klasifikasi Topik. Penulis melakukan proses ekspansi fitur adalah karena dengan melakukan ekspansi fitur dapat membantu meningkatkan akurasi[3]. Selanjutnya untuk alasan ekspansi fitur dengan Word2Vec adalah karena dengan menggunakan Word2Vec dapat menyelesaikan klasifikasi teks singkat dengan lebih baik berdasarkan informasi konteks[9]. Lalu untuk alasan menggunakan *Naive Bayes-Support Vector*

Machine(NBSVM) karena berkinerja baik pada *snippets* dan dokumen yang lebih panjang untuk klasifikasi topik dan seringkali lebih baik daripada hasil yang dipublikasikan sebelumnya[15].

Berdasarkan penjelasan yang telah diuraikan sebelumnya, penelitian ini akan mengimplementasikan Ekspansi Fitur dengan Word2Vec pada Klasifikasi Topik dengan Metode Naive Bayes-Support Vector Machine di Twitter. Adapun Batasan masalah dari penelitian ini adalah dataset yang digunakan hanya dataset yang berbasis Bahasa Indonesia yang di dapatkan dari crawling pada sosial media Twitter.

Adapun Tujuan dari penelitian ini adalah mengetahui pengaruh penerapan *baseline* metode NBSVM dalam mengklasifikasikan topik pada twitter, pengaruh penerapan pembobotan *TF*-IDF pada metode NBSVM dalam mengklasifikasikan topik pada twitter dan pengaruh penerapan ekspansi fitur pada metode NBSVM dalam mengklasifikasikan topik pada twitter.

2. Studi Terkait

Twitter sebagian besar telah dipelajari karena karakteristik dan struktur jaringannya. Tindakan terbesar sejauh ini dilakukan oleh Kwak dkk. [16] yang mempelajari lebih dari 106 juta tweet dan 41,7 juta profil pengguna. Mereka menyelidiki karakteristik jaringan mulai dari pengikut dasar / mengikuti hubungan hingga homofili (kecenderungan orang serupa untuk mengasosiasikan satu sama lain) hingga pagerank. Mereka juga mempelajari tren dalam kaitannya dengan kemiripan dengan topik di CNN dan Google Trends, yang selanjutnya mencirikan sebagian besar tren sebagai yang aktif.

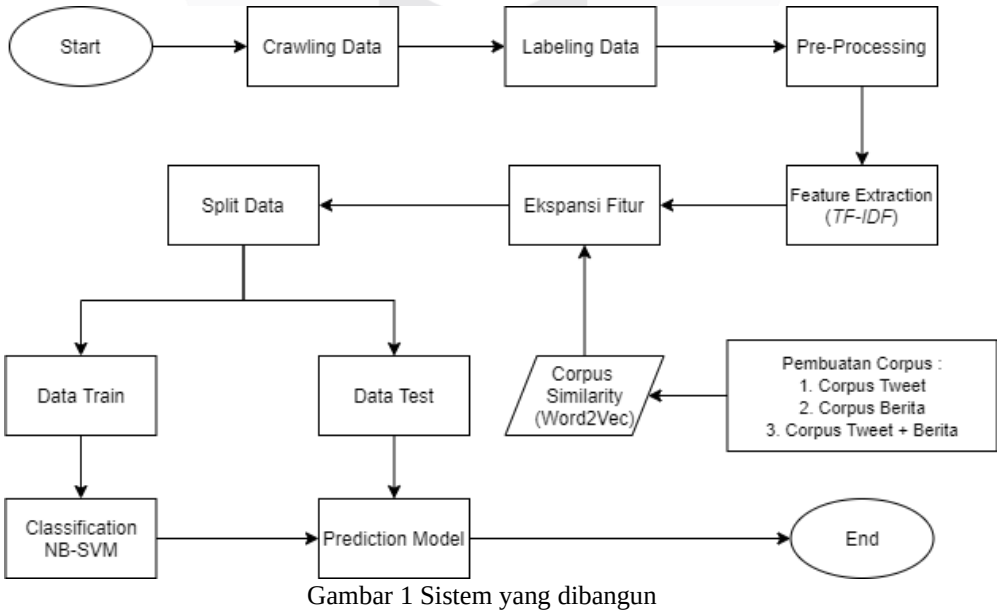
Genc dkk.[17] memperkenalkan teknik klasifikasi berbasis Wikipedia. Pada penelitian tersebut tweet diklasifikasikan dengan memetakan pesan ke halaman Wikipedia yang paling mirip, lalu menghitung jarak semantik dari pesan berdasarkan jarak antara halaman Wikipedia terdekat. Wang dkk[15] menyebutkan bahwa *Naive Bayes-Support Vector Machine* berkinerja baik pada *snippets* dan dokumen yang lebih panjang untuk klasifikasi topik dan seringkali lebih baik daripada hasil yang dipublikasikan sebelumnya.

Terdapat beberapa penelitian yang menerapkan ekspansi fitur, salah satunya adalah penelitian [3], menyimpulkan bahwa dengan menggunakan ekspansi fitur dapat meningkatkan akurasi senilai 5%. Berdasarkan korpus Wikipedia eksternal, Phan dkk. [18] mengusulkan metode untuk menemukan topik tersembunyi menggunakan LDA dan memperluas teks pendek. Chen dkk. [19]membuktikan bahwa memanfaatkan topik pada beberapa perincian dapat membuat model teks pendek dengan lebih tepat. Setiawan dkk.[1] menerapkan *word-embedding* Word2Vec untuk melakukan ekspansi fitur pada topik klasifikasi twitter dengan menggunakan kumpulan data GoogleNews dan IndoNews pada pengklasifikasi Naive Bayes, SVM, dan Regresi Logistik.

Ekspansi fitur dengan Word2Vec dapat menyelesaikan klasifikasi teks singkat dengan lebih baik berdasarkan informasi konteks[9]. Naive Bayes cukup efektif dalam berbagai tugas penambangan data, namun menunjukkan hasil yang mengecewakan dalam masalah klasifikasi[20]. Sebaliknya, SVM telah terbukti berhasil saat digunakan untuk masalah klasifikasi[21]. Oleh karena itu penulis melakukan penelitian yaitu Ekspansi Fitur dengan Word2Vec pada Klasifikasi Topik dengan Metode *Naive Bayes-Support Vector Machine* di Twitter.

3. Sistem yang Dibangun

Sistem topik klasifikasi yang akan dibangun pada penelitian ini dapat dilihat pada Gambar 1.

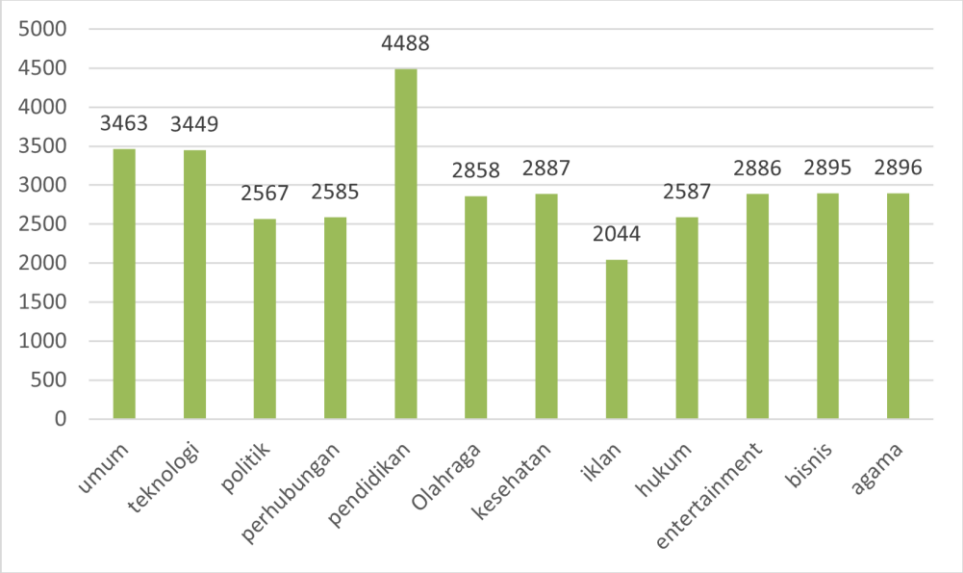


Gambar 1 Sistem yang dibangun

3.1. Crawling Data

Pengumpulan data pada penelitian ini dilakukan dengan mengumpulkan tweet yang dibuat oleh pengguna twitter setiap harinya dengan cara mengakses Application Program Interface(API) yang telah disediakan oleh Twitter. API Twitter hanya bisa diakses melalui permintaan otentikasi. Twitter memperbolehkan membuka atau mengakses (OAuth) dan permintaannya sesuai protocol via pengguna Twitter yang sah. Akses ke dalam API Twitter dibatasi. Batasan ini diterapkan pada tingkatan pengguna dan tingkatan aplikasi.

Data yang terkumpul sebanyak 35.605 yang nantinya akan digunakan sebagai data latih dan data uji, ada beberapa *keyword* yang digunakan untuk mengumpulkan data, beberapa contohnya seperti: beasiswa, dosen, guru, koding, teknologi, internet dan lain-lainnya.



Gambar 2 Persebaran Topik Tweet

Selanjutnya terdapat data yang dikumpulkan dari beberapa media berita seperti cnnindonesia, detik, kompas, liputan6, republika, sindonews dan tempo. Data yang terkumpul dari media tersebut sebanyak 142.544. Persebaran data tersebut dapat dilihat pada tabel 1. Data ini akan digunakan untuk pembuatan *Corpus Similarity*.

Table 1 Persebaran data Berita

Sumber	Jumlah
cnnindonesia.com	29349
detik.com	7974
kompas.com	15055
liputan6.com	251
republika.com	53812
sindonews.com	22401

3.2. Labeling Data

Data yang telah terkumpul pada proses *crawling* menggunakan API dari Twitter akan diberi label untuk dijadikan dataset. Pelabelan dilakukan untuk membedakan topik dari setiap *tweet* yang terdapat pada dataset. Pada penelitian ini terdapat 12 label yang berbeda untuk setiap *tweet* nya, 12 label tersebut di dapat dari penelitian yang telah dilakukan sebelumnya[1]. Pelabelan dilakukan oleh 4 orang dengan prinsip *majority votes*. Label yang digunakan seperti : Iklan, Umum, Agama, Entertainment, Olahraga, Politik, Kesehatan, Teknologi, Pendidikan, Hukum, Bisnis dan Perhubungan. Untuk contoh label pada tweet dapat dilihat pada tabel 2.

Table 2 Contoh label pada tweet

Tweet	Label

@bertanyarl Masuk IPA tapi pas kuliah di soshum, ada beberapa matkul yg dasarnya udah ada di SMA, jadi agak keteteran ngikutin matkul, soalnya gue ga dpt materi dasarnya di SMA	Pendidikan
Ibu Kota Baru bisa dikembangkan sebagai smart city dengan solusi yang diadakan oleh 5G. Jaringan generasi ke lima ini akan mendukung kerja sensor dan benda yang digerakkan dengan teknologi Internet of Things (IoT) di Ibu Kota Baru.	Teknologi

3.3. Preprocessing

Salah satu proses utama dalam text mining adalah pembersihan, reduksi dan transformasi sebelum penerapan algoritma klasifikasi. *Preprocessing* memiliki dampak yang besar dalam algoritma klasifikasi karena teks merupakan bentuk data yang tidak terstruktur dengan jumlah dimensi yang sangat besar[22]. Pada tahap *preprocessing* ini dilakukan dengan bantuan *Library NLTK* dan *Sastrawi*. Berikut beberapa tahap *preprocessing* yang dilakukan :

1. *Data Cleaning*
   Pada tahap ini dilakukan proses pembersihan pada data, yaitu dengan menghilangkan angka, *Hashtag*, tanda baca, *special character*, *url*, dan *mention*. Proses ini dilakukan dengan menggunakan *Library re* (*regular expression operations*)
2. *Case Folding*
   Pada tahap ini dilakukan perubahan kata atau frasa dalam teks tweet menjadi huruf kecil (a-z). Proses ini membantu dalam mengatasi masalah ketika kata-kata atau frasa ditulis dengan kapitalisasi yang berbeda. Proses ini dilakukan dengan menggunakan *Library re* (*regular expression operations*)
3. Normalisasi Kata
   Proses ini mengubah kata-kata yang disingkat, kata yang salah dalam penulisannya (*typo*), kata gaul dan kata alay menjadi kata yang formal dengan bantuan kamus yang terdapat di *github*.
4. *Stopwords removal*
   Proses ini menghilangkan kata-kata non topik yang dianggap tidak penting seperti: “dan”, “ini”, “itu”, “adalah”, “atau”, “yang”, dan lain-lain. Pemrosesan awal ini membantu mengurangi fitur yang tidak relevan dalam data. Proses ini dilakukan dengan menggunakan *Library NLTK*
5. *Stemming*
   Stemming adalah proses mencari akar kata dengan menghilangkan awalan, sisipan, akhiran, dan *confix* (gabungan awalan dan akhiran) pada kata turunan. Dengan stemming, kata yang memiliki akar yang sama akan dianggap sebagai token (fitur) yang sama. Dalam Pengambilan Informasi, ini membantu meningkatkan kinerja pengambilan data. Proses ini dilakukan dengan menggunakan *Library Sastrawi*.
6. *Tokenization*
   Proses ini dilakukan untuk memotong tweet masukan menjadi kata-kata yang menyusunnya. Pada prinsipnya, ini memisahkan setiap kata dalam teks tweet. Proses ini dilakukan dengan menggunakan *Library NLTK*.

Preprocessing dilakukan dengan tujuan agar dataset yang digunakan siap diproses pada sistem yang telah dibuat. Terdapat beberapa tahapan pada *preprocessing*. Semua tahapan yang dilakukan dapat dilihat pada tabel 3.

Table 3 Tahapan Preprocessing

Preprocessing	**Sebelum**	**Sesudah**
Data Cleaning	@collegemenfess Gw Maba sistem informasi di Yogyakarta ... Gila semester 1 puyeng banget kepala gw ... Udh bnyk tugas , kuis , dll. Mana gw lulusan tata boga lagi mana ngerti koding Ama pemrograman , btw gw kuliah swasta	Gw Maba sistem informasi di Yogyakarta Gila semester puyeng banget kepala gw Udh bnyk tugas kuis dll Mana gw lulusan tata boga lagi mana ngerti koding Ama pemrograman btw gw kuliah swasta
Case Folding	Gw Maba sistem informasi di Yogyakarta Gila semester puyeng banget kepala gw Udh bnyk tugas kuis dll Mana gw lulusan tata boga lagi mana ngerti koding Ama pemrograman btw gw kuliah swasta	gw maba sistem informasi di yogyakarta gila semester puyeng banget kepala gw udh bnyk tugas kuis dll mana gw lulusan tata boga lagi mana ngerti koding ama pemrograman btw gw kuliah swasta
Normalisasi	gw maba sistem informasi di yogyakarta gila semester puyeng banget kepala gw udh bnyk tugas kuis dll mana gw lulusan tata	gw maba sistem informasi di yogyakarta gila semester puyeng banget kepala gw udah banyak tugas kuis dll mana gw lulusan tata

	boga lagi mana ngerti koding ama pemrograman btw gw kuliah swasta	boga lagi mana ngerti koding sama pemrograman btw gw kuliah swasta
Stopwords	gw maba sistem informasi di yogyakarta gila semester puyeng banget kepala gw udah banyak tugas kuis dll mana gw lulusan tata boga lagi ngerti koding sama pemrograman btw gw kuliah swasta	gw maba sistem informasi yogyakarta gila semester puyeng banget kepala gw udah banyak tugas kuis dll gw lulusan tata boga lagi ngerti koding pemrograman btw gw kuliah swasta
Stemming	gw maba sistem informasi yogyakarta gila semester puyeng banget kepala gw udah banyak tugas kuis dll gw lulusan tata boga lagi ngerti koding pemrograman btw gw kuliah swasta	gw maba sistem informasi yogyakarta gila semester puyeng banget kepala gw udah banyak tugas kuis dll gw lulusan tata boga lagi ngerti koding pemrograman btw gw kuliah swasta
Tokenization	gw maba sistem informasi yogyakarta gila semester puyeng banget kepala gw udah banyak tugas kuis dll gw lulusan tata boga lagi ngerti koding pemrograman btw gw kuliah swasta	[‘gw’, ‘maba’, ‘sistem’, ’informasi’, ‘yogyakarta’, ‘gila’, ‘semester’, ‘puyeng’, ‘banget’, ‘kepala’, ‘gw’, ‘udah’, ‘banyak’, ‘tugas’, ‘kuis’, ‘dll’, ‘gw’, ‘lulusan’, ‘tata’, ‘boga’, ‘lagi’, ‘ngerti’, ‘koding’, ‘pemrograman’, ‘btw’, ‘gw’, ‘kuliah’, ‘swasta’]

3.4. *Term Frequency – Invers Document Frequency* (*TF-IDF*)

Pada kasus klasifikasi teks, dokumen teks mungkin cocok dengan sebagian atau banyak kategori. Oleh karena itu, perlu menentukan kategori yang paling cocok untuk dokumen teks tersebut. Pendekatan *term* (*word*) frequency / inverse document frequency(tf-idf) biasa digunakan untuk menimbang setiap kata dalam dokumen teks menurut keunikannya. Dengan kata lain, pendekatan TF-IDF menangkap relevansi(keterkaitan) antar kata, dokumen teks dan kategori tertentu[23].

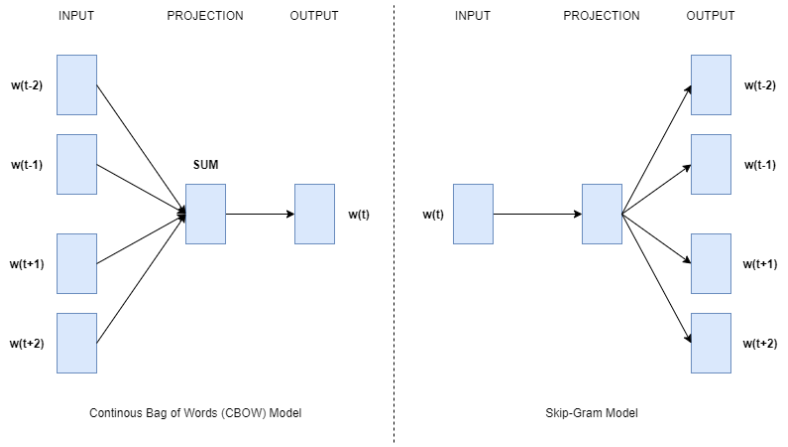
Term-Frequency (*TF*) digunakan untuk mengukur berapa kali suatu kata muncul dalam suatu dokumen. Misalkan, terdapat dokumen "T1" yang berisi 5000 kata, lalu terdapat kata "Alpha" dalam dokumen tersebut sebanyak 10 kali. Jadi dalam hal ini frekuensi dari "Alpha" dalam dokumen "T1" adalah TF= 10/5000 = 0.002 [24].

Ketika *term frequency* dokumen dihitung, dapat diamati bahwa algoritma memperlakukan semua kata secara setara, tidak masalah jika itu adalah sebuah *stopword* seperti "*of*". Semua kata memiliki kepentingan yang berbeda. Misal, *stopword* "*of*" muncul dalam dokumen 2000 kali tetapi tidak ada gunanya atau memiliki arti yang sangat kurang, itulah gunanya *IDF*. *Inverse Document Frequency* memberikan bobot yang lebih rendah untuk kata-kata yang sering dan memberikan bobot yang lebih besar untuk kata-kata yang jarang. Misalnya, terdapat 10 dokumen dan istilah "*technology*" muncul dalam 5 dokumen tersebut, sehingga perhitungan *IDF* nya adalah IDF = log_e(10/5) = 0,3010 [24].

3.5. Ekspansi Fitur dengan Word2Vec

Ekspansi fitur adalah proses memperkaya teks asli dengan tambahan semantik agar tampak seperti dokumen teks berukuran besar[3]. Pada penelitian ini penulis menggunakan Word2Vec untuk melakukan ekspansi Fitur. Word2vec mengacu pada sekelompok model yang dikembangkan oleh Mikolov dkk[25]. Word2Vec digunakan untuk membuat dan melatih ruang vektor semantik, seringkali terdiri dari beberapa ratus dimensi, berdasarkan kumpulan teks [25]. Dalam ruang vektor ini, setiap kata dari korpus direpresentasikan sebagai vektor. Kata-kata yang berbagi konteks secara geografis berdekatan di ruang vektor tersebut[26].

Model word2vec oleh Mikolov dkk[25]. Penelitian tersebut telah menerima banyak perhatian dalam beberapa tahun terakhir. Representasi vektor dari kata-kata yang diperiksa dalam model word2vec memiliki makna semantik dan berguna untuk berbagai tugas NLP.[27]. Pada Word2Vec terdapat dua algoritma pembelajaran utama, yaitu continuous bag-of-words (CBOW) dan skip-gram[1]. Untuk ilustrasi dari kedua algoritma tersebut ditunjukkan pada Gambar 3. Kedua model tersebut memiliki kompleksitas komputasi yang rendah, sehingga dapat dengan mudah menangani korpus dengan ukuran yang berkisar di miliaran kata dalam hitungan jam[26]. Sementara model CBOW lebih cepat, telah diamati bahwa skip-gram berkinerja lebih baik pada dataset kecil[25].



Gambar 3 Model Arsitektur Word2Vec[1]

Dengan menggunakan continuous bag-of-words (CBOW), urutan kata-kata tidak memengaruhi proyeksi. Model ini memprediksi kata saat ini berdasarkan konteksnya. Skip-gram memprediksi kata-kata di sekitarnya berdasarkan kata saat ini. Berbeda dengan model bag-of-words standar, continuous bag-of-words menggunakan representasi konteks yang terdistribusi. Penting juga untuk menyatakan bahwa matriks bobot antara input dan projection layer digunakan bersama untuk semua posisi kata[28].

Corpus yang dibuat menggunakan *word embedding Word2Vec* model *Skip-gram*. *Corpus* merupakan kumpulan kata yang diurutkan nilai similaritasnya dari tertinggi hingga terendah. Hasil yang didapatkan seperti berikut.

1. *Corpus* data *tweet*
   Kosakata yang didapat sebanyak 36.961 kata dan berikut contoh kata yang mirip dapat dilihat pada tabel 4.

Table 4 Contoh Corpus Data Tweet

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
vaksin	dosis	imunisasi	sinovac	suntik	aman
	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10
	ptm	pfizer	jarak	tes	pjj

2. *Corpus* data media berita
   Kosakata yang didapat sebanyak 225.930 kata dan berikut contoh kata yang mirip dapat dilihat pada tabel 5.

Table 5 Contoh Corpus Data Media Berita

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
kuliah	mahasiswa	dosen	thinkstockphotos	lulus	yasman
	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10
	pascasarjana	ajar	kouji	raeni	mahasiswa

3. *Corpus* data tweet ditambah data media berita
   Kosakata yang didapat sebanyak 240.649 kata dan berikut contoh kata yang mirip dapat dilihat pada tabel 6.

Table 6 Contoh Corpus data tweet + data media berita

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
sekolah	sma	sd	siswa	murid	slb
	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10
	madrasah	sman	siswi	guru	tk

Sebagai contoh terdapat *tweet* "mahasiswa universitas telkom wisuda pada hari kamis" lalu dengan dilakukan ekspansi fitur dengan ukuran fitur *top similarity* 10 menggunakan *Corpus* media berita, pada representasi vektor *TF-IDF* kata "kuliah" bobot katanya adalah nol. Namun, pada *tweet* terdapat kata "mahasiswa", karena "mahasiswa" terdapat pada *top similarity* 10 dari kata "kuliah" maka kata "kuliah" bernilai bobot sama dengan nilai dari "mahasiswa", untuk *rank similarity* dari kata "kuliah" pada *Corpus* media berita dapat dilihat pada tabel 7.

Table 7 Corpus data media berita

Kata	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5
kuliah	mahasiswa	dosen	thinkstockphotos	lulus	yasman
	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10
	pascasarjana	ajar	kouji	raeni	mahasiswa

3.6. NB-SVM

Naive Bayes-Support Vector Machine adalah salah satu metode yang paling populer untuk klasifikasi teks dan telah banyak digunakan sebagai dasar untuk berbagai pendekatan representasi teks. Metode ini menggunakan fitur Naive Bayes (NB) untuk menimbang representasi bag-of-n-gram yang jarang. N-gram menangkap urutan kata dalam konteks pendek dan fitur NB memberikan bobot lebih pada kata-kata penting tersebut[29].

Varian *Naive Bayes* (NB) dan *Support Vector Machines* (SVM) sering digunakan sebagai metode dasar untuk klasifikasi teks, tetapi kinerjanya sangat bervariasi tergantung pada varian model, fitur yang digunakan dan tugas atau dataset[15]. *Naive Bayes-Support Vector Machine*(NBSVM) berkinerja baik pada *snippets* dan dokumen yang lebih panjang untuk klasifikasi topik dan seringkali lebih baik daripada hasil yang dipublikasikan sebelumnya. Oleh karena itu, NBSVM tampaknya menjadi dasar yang tepat dan sangat kuat untuk metode canggih yang bertujuan untuk menangani sekumpulan fitur[15].

Berdasarkan penelitian sebelumnya[15] dengan menggabungkan pengklasifikasi generatif dan diskriminatif, menyajikan varian model sederhana di mana SVM dibangun di atas rasio penghitungan log NB sebagai nilai fitur, dan menunjukkan kinerja yang kuat dan tangguh atas semua tugas yang disajikan.

Setelah melalui tahap preprocessing, pembobotan kata menggunakan *tf-idf*, dan proses Ekspansi Fitur, proses selanjutnya adalah klasifikasi menggunakan *Naive Bayes-Support Vector Machine* (*NBSVM*). Sebelum melakukan proses klasifikasi, dilakukan data split yaitu membagi dataset yang ada menjadi data latih dan data uji dengan perbandingan 80:20. Pada proses ini dilakukan pengulangan eksekusi program sebanyak 3 kali lalu diambil nilai rata-rata dari akurasinya.

3.7. Confusion Matrix

Confusion matrix adalah konsep *machine learning* yang berisi informasi tentang prediksi dan klasifikasi aktual dari suatu sistem klasifikasi. [30]. Matriks konfusi memiliki dua dimensi, satu dimensi diindeks oleh *actual class* dari suatu objek, yang lain diindeks oleh kelas yang diprediksi oleh pengklasifikasi [31]. Kinerja sistem biasanya dievaluasi dengan menggunakan data dalam matriks. Tabel berikut menunjukkan *confusion matrix* untuk pengklasifikasi dua kelas[32].

Table 8 Confusion Matrix

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

Arti dari entri matriks konfusi adalah sebagai berikut[32] :

- a adalah jumlah prediksi yang benar bahwa suatu instance negatif,
- b adalah jumlah prediksi yang salah bahwa sebuah instance bernilai positif,
- c adalah banyaknya prediksi yang salah pada sebuah instance negatif, dan
- d adalah jumlah prediksi yang benar bahwa suatu instance bernilai positif.

Beberapa istilah standar telah ditetapkan untuk matriks 2 kelas [32]:

a. *Accuracy*
Akurasi adalah persentase dari jumlah total prediksi yang benar, ditentukan oleh persamaan.:

$$Accuracy = \frac{a+d}{a+b+c+d}$$

b. Recall
Recall atau *true positive rate* (TP) adalah persentase kasus positif yang teridentifikasi dengan benar dan dihitung menggunakan rumus berikut:

$$Recall = \frac{d}{c+d}$$

c. *Precision*
Presisi (P) adalah proporsi kasus positif yang diprediksi yang benar, yang dihitung menggunakan persamaan:

$$Precision = \frac{d}{b+d}$$

d. F1-Measure
F1-Measure adalah perbandingan rata-rata presisi dan recall yang dibobotkan.

$$F1 - Measure = 2 \times \frac{(precision \times recall)}{(precision + recall)}$$ [33]

4. Evaluasi

Hasil pengujian dan Analisis Hasil pengujian dari sistem yang telah dibangun.

4.1. Skenario dan Hasil Pengujian

Skenario untuk mencapai tujuan dari penelitian ini yaitu yang pertama adalah penerapan *baseline* metode NBSVM dalam mengklasifikasikan topik pada twitter, pengujian kedua yaitu penerapan pembobotan *TF-IDF* pada metode NBSVM dalam mengklasifikasikan topik pada twitter dan pengujian ketiga yaitu penerapan ekspansi fitur pada metode NBSVM dalam mengklasifikasikan topik pada twitter.

4.1.1. Pengujian pertama (*Baseline NBSVM*)

Pengujian pertama yaitu menggunakan baseline NBSVM, artinya tanpa melakukan pembobotan dengan *TF-IDF*. Dalam pengujian ini rasio yang digunakan adalah 70:30 , 80:20 dan 90:10 untuk perbandingan data latih dan data ujinya. Pengujian ini dilakukan sebanyak 3 kali untuk setiap rasio nya lalu di ambil nilai rata-rata dari akurasi dan *F1-Measure* dari pengujian tersebut. Hasil dari pengujian ini dapat dilihat pada tabel 9.

Table 9 Hasil Performansi Baseline NBSVM

Rasio	Akurasi (%)	*F1-Measure*(%)
90 : 10	78,77	79,02
80 : 20	**79,31**	**79,63**
70 : 30	78,63	78,78

Dari tabel 9 dapat disimpulkan bahwa dengan menggunakan rasio 80:20 untuk perbandingan data latih dan data ujinya memiliki performansi terbaik untuk akurasi dan F1-*Measure*. Jadi, untuk pengujian selanjutnya akan menggunakan rasio dengan perebandingan 80:20 untuk data latih dan data ujinya.

4.1.2. Pengujian kedua (Pembobotan dengan *TF-IDF*)

Pengujian kedua yaitu pembobotan dengan *TF-IDF*. Pada pengujian ini rasio yang digunakan adalah 80:20 untuk perbandingan data latih dan data ujinya karena pada pengujian sebelumnya telah terbukti rasio ini terbaik dibandingkan dengan rasio lainnya yaitu 90:10 dan 70:30. Pengujian ini dilakukan sebanyak 3 kali lalu diambil nilai rata-rata dari akurasi dan *F1-Measure*.

Table 10 Perbandingan Hasi Performansi NBSVM

Classifier	Akurasi (%)	*F1-Measure*(%)
Baseline NBSVM	79,31	79,63
Baseline NBSVM + TF-IDF	**84,17 (+4,86)**	**84,24 (+4,61)**

Berdasarkan pengujian sebelumnya, di dapatkan akurasi sebesar 79,31% dan *F1-Measure* sebesar 79,63% untuk baseline NBSVM. Lalu, pada pengujian ini yaitu baseline NBSVM dengan pembobotan *TF-IDF* di dapatkan akurasi sebesar 84,17% dan *F1-Measure* sebesar 84,24%. Akurasi mengalami kenaikan sebesar 4,86% dan *F1-Measure* mengalami kenaikan sebesar 4,61% ketika menambahkan pembobotan *TF-IDF* pada baseline NBSVM.

4.1.3. Pengujian Ketiga (Penerapan Ekspansi Fitur)

Pengujian ketiga yaitu menerapkan ekspansi fitur setelah dilakukan pembobotan *TF-IDF* pada baseline NBSVM. Ekspansi fitur dilakukan dengan menggunakan 3 *Corpus Word2Vec*, yaitu *Corpus* data tweet, *Corpus* data berita dan *Corpus* kata data tweet + data berita. Pengujian dilakukan masih sama seperti sebelumnya, yaitu dengan perbandingan 80:20 untuk data latih dan data ujinya. Untuk Ekspansi Fitur nya yaitu menggunakan nilai similaritas top 1, 5 dan 10 dari *Corpus* yang telah dibuat sebelumnya. Pengujian ini dilakukan sebanyak 3 kali untuk setiap *Corpus*, lalu diambil nilai rata-rata dari akurasinya.

Nilai akurasi dan *F1-Measure* dari pengujian Ekspansi Fitur menggunakan algoritma NBSVM bisa dilihat pada tabel 11. Kolom Baseline menunjukkan hasil tanpa menggunakan pembobotan TF-IDF dan tanpa ekspansi fitur. Kolom *Corpus* Tweet, *Corpus* Berita dan *Corput* Tweet + Berita menunjukkan hasil dengan pembobotan *TF-IDF* yang kemudian dilakukan fitur ekspansi sesuai dengan *Corpus*nya masing-masing. Nilai akurasi dan *F1-Measure* dari hasil pengujian tersebut mengalami peningkatan dibandingkan nilai akurasi dan *F1-Measure* pada pengujian *baseline* NBSVM. Nilai akurasi tertingginya ada pada *top similarity* 1 dengan menggunakan *corpus* tweet + berita sebesar 84,75% yang sebelumnya 79,31% pada pengujian baseline dan untuk *F1-Measure* tertingginya ada pada *top similarity* 1 dengan menggunakan *corpus* berita dengan nilai 84,90% yang sebelumnya 79,63% pada pengujian *baseline* NBSVM.

Table 11 Perbandingan Baseline dengan Ekspansi Fitur pada NBSVM

Top Similarity	**Akurasi (%)**				***F1-Measure (%)***			
	Baseline	*Corpus* Tweet	*Corpus* Berita	*Corpus* Tweet + Berita	Baseline	*Corpus* Tweet	*Corpus* Berita	*Corpus* Tweet + Berita
1	79,31	84,73 (+5,42)	84,74 (+5,43)	**84,75 (+5,44)**	79,63	84,77 (+5,14)	**84,90 (+5,27)**	84,88 (+5,25)
5	79,31	84,45 (+5,14)	84,19 (+4,88)	84,32 (+5,01)	79,63	84,55 (+4,92)	84,61 (+4,98)	84,5 (+4,87)
10	79,31	84,6 (+5,29)	84,02 (4,71)	84,36 (+5,05)	79,63	84,73 (+5,1)	84,14 (+4,51)	84,52 (+4,89)

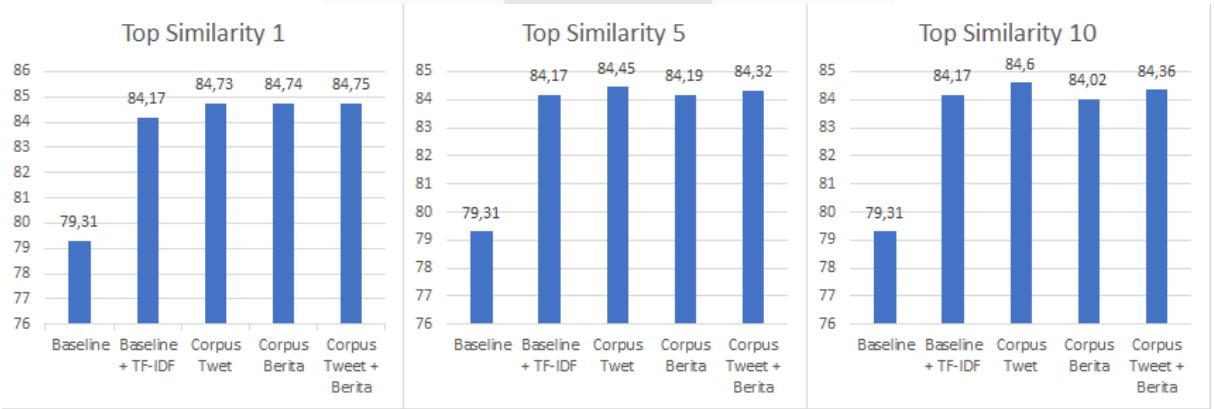
Nilai akurasi dan *F1-Measure* dari pengujian Ekspansi Fitur menggunakan algoritma NBSVM bisa dilihat pada tabel 12. Kolom Baseline + TF-IDF menunjukkan hasil menggunakan pembobotan TF-IDF dan tanpa ekspansi fitur. Kolom *Corpus* Tweet, *Corpus* Berita dan *Corput* Tweet + Berita menunjukkan hasil dengan pembobotan *TF-IDF* yang kemudian dilakukan fitur ekspansi sesuai dengan *Corpus*nya masing-masing. Nilai akurasi dan *F1-Measure* dari hasil pengujian tersebut ada yang mengalami penururan. Namun, sebagian besar mengalami peningkatan dibandingkan nilai akurasi dan *F1-Measure* pada pengujian *baseline* NBSVM ditambah dengan pembobotan *TF-IDF* tanpa ekspansi fitur. Nilai akurasi tertingginya ada pada *top similarity* 1 dengan menggunakan *corpus* tweet + berita sebesar 84,75% yang sebelumnya 84,17% pada pengujian baseline dengan pembobotan *TF-IDF* dan untuk *F1-Measure* tertingginya ada pada *top similarity* 1 dengan menggunakan *corpus* berita dengan nilai 84,90% yang sebelumnya 84,24% pada pengujian *baseline* NBSVM dengan pembobotan *TF-IDF*.

Table 12 Perbandingan Baseline + TF-IDF dengan Ekspansi Fitur pada NBSVM

Top Similarity	**Akurasi (%)**				***F1-Measure (%)***			
	Baseline + TF-IDF	*Corpus* Tweet	*Corpus* Berita	*Corpus* Tweet + Berita	Baseline + TF-IDF	*Corpus* Tweet	*Corpus* Berita	*Corpus* Tweet + Berita
1	84,17	84,73 (+0,56)	84,74 (+0,57)	**84,75** (+0,58)	84,24	84,77 (+0,53)	**84,90 (+0,66)**	84,88 (+0,64)
5	84,17	84,45 (+0,28)	84,19 (+0,02)	84,32 (+0,15)	84,24	84,55 (+0,31)	84,61 (+0,37)	84,5 (+0,26)
10	84,17	84,6 (+0,43)	84,02 (-0,15)	84,36 (+0,19)	84,24	84,73 (+0,49)	84,14 (-0,1)	84,52 (+0,28)

4.2. Analisis Hasil Pengujian

Data hasil pengujian yang telah dilakukan sebelumnya akan divisualisasikan melalui diagram yang dapat dilihat pada Gambar 4. Hasil tersebut dikelompokan berdasarkan variasi penggunaan fitur pada proses ekspansi fitur.



Gambar 4 Grafik Analisis Nilai Akurasi Sistem Klasifikasi Topik

Pada hasil pengujian didapatkan nilai akurasi yang berbeda-beda ketika menggunakan Ekspansi Fitur dengan corpus yang berbeda. Nilai akurasi yang didapatkan mengalami peningkatan ketika ditambahkan teknik TF-IDF untuk pembobotan kata. Lalu, ketika sistem mengimplementasikan Ekspansi Fitur, sebagian besar nilai dari akurasi juga ikut meningkat.

Pada Gambar 4 menunjukkan untuk model *top similarity* 1 menggunakan *Corpus* Tweet mendapat nilai akurasi sebesar 84,73% atau terjadi peningkatan akurasi sebesar 5,42% dari baseline dan peningkatan sebesar 0,56% dari baseline dengan pembobotan *TF-IDF*. Lalu saat menggunakan *Corpus* Berita mendapat nilai akurasi sebesar 84,74% atau terjadi peningkatan sebesar 5,43% dari baseline dan peningkatan sebesar 0,57% dari baseline dengan pembobotan *TF-IDF*. Dan saat menggunakan *Corpus* Tweet + Berita mendapat nilai akurasi sebesar 84,75% atau terjadi peningkatan sebesar 5,44% dari baseline dan peningkatan sebesar 0,58% dari baseline dengan pembobotan *TF-IDF*.

Sementara itu, pada model *top similarity* 5 menggunakan *Corpus* Tweet mendapat nilai akurasi sebesar 84,45% atau terjadi peningkatan sebesar 5,14% dari baseline dan peningkatan sebesar 0,28% dari baseline dengan pembobotan *TF-IDF*. Lalu saat menggunakan *Corpus* Berita mendapatkan nilai akurasi sebesar 84,19% atau terjadi peningkatan sebesar 4,88% dari baseline dan peningkatan sebesar 0,02% dari baseline dengan pembobotan *TF-IDF*. Dan saat menggunakan *Corpus* Tweet + Berita mendapat nilai akurasi sebesar 84,32% atau terjadi peningkatan sebesar 5,01% dari baseline dan peningkatan sebesar 0,15% dari baseline dengan pembobotan *TF-IDF*.

Selanjutnya, pada model *top similarity* 10 menggunakan *Corpus* Tweet mendapat nilai akurasi sebesar 84,6% atau terjadi peningkatan sebesar 5,29% dari baseline dan peningkatan sebesar 0,43% dari baseline dengan pembobotan *TF-IDF*. Lalu saat menggunakan *Corpus* Berita mendapatkan nilai akurasi sebesar 84,02% atau terjadi peningkatan sebesar 4,71% dari baseline dan penurunan sebesar 0,15% dari baseline dengan pembobotan *TF-IDF*. Dan saat menggunakan *Corpus* Tweet + Berita mendapat nilai akurasi sebesar 84,36% atau terjadi peningkatan sebesar 5,05% dari baseline dan peningkatan sebesar 0,19% dari baseline dengan pembobotan *TF-IDF*.

Model *top similarity* 1 dengan menggunakan *Corpus* Tweet + Berita mendapatkan peningkatan yang paling besar ketika dibandingkan dengan baseline tanpa pembobotan *TF-IDF* dan baseline dengan pembobotan *TF-IDF* yaitu kenaikan akurasi sebesar 5,44% dari baseline tanpa pembobotan *TF-IDF* dan kenaikan sebesar 0,58% dari baseline dengan pembobotan *TF-IDF*.

Perbandingan klasifikasi yang digunakan pada penelitian ini dengan penelitian sebelumnya yang menerapkan ekspansi fitur pada kasus klasifikasi topik[1], dapat dilihat pada tabel 13. Pada penelitian ini menggunakan klasifikasi Naive Bayes-Support Vector Machine (NBSVM), lalu pada penelitian sebelumnya terdapat penggunaan *Naive Bayes* (NB) dan *Support Vector Machine* (SVM) secara terpisah. Pada penelitian sebelumnya, ketika klasifikasi yang digunakan adalah *Naive Bayes* (NB), peningkatan akurasi tertinggi ketika penerapan *baseline* dan ketika penerapan ekspansi fitur adalah 0.21%. Lalu, ketika klasifikasi yang diguanakan adalah *Support Vector Machine* (SVM), peningkatan akurasi tertinggi ketika penerapan *baseline* dan ketika penerapan ekspansi fitur adalah 0.13%. Selanjutnya, pada penelitian saat ini dengan menggunakan klasifikasi *Naive Bayes-Support Vector Machine* (NBSVM), peningkatan akurasi tertinggi ketika penerapan baseline dan ketika penerapan ekspansi fitur adalah 0.58%.

Table 13 Perbandingan Peningkatan Akurasi dengan Penelitian Sebelumnya

	Penelitian Sebelumnya		**Penelitian saat ini**
	Naive Bayes (NB)	Support Vector Machine (SVM)	Naive Bayes-Support Vector Machine (NBSVM)
Peningkatan Akurasi Tertinggi	0.21%	0.13%	**0,58%**

5. Kesimpulan

Pada penelitian ini, telah dilakukan pembuatan sistem untuk klasifikasi topik pada tweet menggunakan Ekspansi Fitur Word2Vec. Naive Bayes-Support Vector Machine(NBSVM) digunakan untuk klasifikasi di penelitian ini. Ekspansi Fitur Word2Vec pada penelitian digunakan pada sistem bertujuan untuk mengurangi ketidakcocokan kosakata pada kalimat tweet. Ekspansi Fitur dilakukan dengan menggunakan 3 *corpus* similarity Word2Vec (Tweet, Berita dan Tweet + Berita) dan juga 3 jenis *top similarity* yaitu *top* 1, *top* 5 dan *top* 10 untuk mencari model yang terbaik.

Pada penelitian ini terbukti Ekspansi Fitur berhasil meningkatkan nilai akurasi untuk metode klasifikasi NBSVM. Peningkatan nilai akurasi tertinggi terdapat pada top similarity 1 dengan corpus tweet + berita dengan nilai 84,75% dengan kenaikan sebesar 5,44% dari baseline dan mengalami kenaikan sebesar 0,58% dibandingkan baseline dengan pembobotan *TF-IDF*.

Daftar Pustaka

[1] 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.*, 2017, doi: 10.1109/TSSA.2016.7871085.

[2] M. A. Zingla, L. Chiraz, Y. Slimani, and C. Berrut, "Statistical and semantic approaches for tweet contextualization," *Procedia Comput. Sci.*, vol. 60, no. 1, pp. 498–507, 2015, doi: 10.1016/j.procs.2015.08.171.

[3] M. A. Fauzi, R. F. N. Firmansyah, and T. Afirianto, "Improving sentiment analysis of short informal Indonesian product reviews using synonym based feature expansion," 2018, doi: 10.12928/TELKOMNIKA.v16i3.7751.

[4] C.-F. Tsai, W.-Y. Lin, Z.-F. Hong, and C.-Y. Hsieh, "Distance-based features in pattern classification," *EURASIP J. Adv. Signal Process.*, 2011, doi: 10.1186/1687-6180-2011-62.

[5] K. Yao, W. Lu, S. Zhang, H. Xiao, and Y. Li, "Feature expansion and feature selection for general pattern recognition problems," *Proc. 2003 Int. Conf. Neural Networks Signal Process. ICNNSP'03*, vol. 1, pp. 29–32, 2003.

[6] J. Dalton, L. Dietz, and J. Allan, "Entity query feature expansion using knowledge base links," *SIGIR 2014 - Proc. 37th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 365–374, 2014.

[7] C. Guo, Y. Zhou, Y. Ping, Z. Zhang, G. Liu, and Y. Yang, "A distance sum-based hybrid method for intrusion detection," *Appl. Intell.*, vol. 40, no. 1, pp. 178–188, 2014, doi: 10.1007/s10489-013-0452-6.

[8] J. Jotheeswaran and D. S. Koteeswaran, "A WEIGHTED SEMANTIC FEATURE EXPANSION USING HYPONYMY TREE FOR FEATURE INTEGRATION IN SENTIMENT ANALYSIS," *Int. Conf. Green Comput. Internet ofThings*, pp. 289–293, 2015.

[9] 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.

[10] F. F. Irfani, M. A. Fauzi, and Y. A. Sari, "News Classification on Twitter Using Naive Bayes and Hypernym-Hyponym Based Feature Expansion," *3rd Int. Conf. Sustain. Inf. Eng. Technol. SIET 2018 - Proc.*, pp. 317–321, 2018, doi: 10.1109/SIET.2018.8693213.

[11] W. Fenlin, Z. Yifei, and W. Cheng, "ADAPTIVE NORMALIZED WEIGHTED KNN TEXT CLASSIFICATION BASED ON PSO," pp. 109–115, 2016.

[12] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5432–5435, 2009, doi: 10.1016/j.eswa.2008.06.054.

[13] K. Gayathri and A. Marimuthu, "Text document pre-processing with the KNN for classification using the SVM," *7th Int. Conf. Intell. Syst. Control. ISCO 2013*, pp. 453–457, 2013, doi: 10.1109/ISCO.2013.6481197.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[15] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," *50th Annu. Meet. Assoc. Comput. Linguist. ACL 2012 - Proc. Conf.*, vol. 2, no. July, pp. 90–94, 2012.

[16] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a Social Network or a News Media?," *What is Twitter, a Soc. Netw. or a News Media?*, 2010.

[17] Y. Genc, Y. Sakamoto, and J. V. Nickerson, "Discovering context: Classifying tweets through a semantic transform based on wikipedia," *Proc. HCI Int.*, pp. 484–492, 2011.

[18] X. H. Phan, L. M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," *Proceeding 17th Int. Conf. World Wide Web 2008,*

WWW'08, pp. 91–99, 2008, doi: 10.1145/1367497.1367510.

[19] M. Chen, X. Jin, and D. Shen, "Short text classification improved by learning multi-granularity topics," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 1776–1781, 2011, doi: 10.5591/978-1-57735-516-8/IJCAI11-298.

[20] S. H. Myaeng, K. S. Han, and H. C. Rim, "Some effective techniques for naive bayes text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 11, pp. 1457–1466, 2006, doi: 10.1109/TKDE.2006.180.

[21] V. Jakkula, "Tutorial on Support Vector Machine (SVM)," *Sch. EECS, Washingt. State Univ.*, 2006, [Online]. Available: http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf.

[22] G. Orellana, B. Arias, M. Orellana, V. Saquicela, F. Baculima, and N. Piedra, "A study on the impact of pre-processing techniques in Spanish and english text classification over short and large text documents," *Proc. - 3rd Int. Conf. Inf. Syst. Comput. Sci. INCISCOS 2018*, vol. 2018-Decem, pp. 277–283, 2018, doi: 10.1109/INCISCOS.2018.00047.

[23] Y. T. Zhang, L. Gong, and Y. C. Wang, "Improved TF-IDF approach for text classification," *J. Zhejiang Univ. Sci.*, vol. 6 A, pp. 49–55, 2005, doi: 10.1631/jzus.2005.A0049.

[24] 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.

[25] 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.

[26] D. Herremans and C. H. Chuan, "Modeling musical context using word2vec," pp. 11–18, 2017.

[27] X. Rong, "word2vec Parameter Learning Explained," 2014, [Online]. Available: http://arxiv.org/abs/1411.2738.

[28] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and Word2vec for text classification with semantic features," *Proc. 2015 IEEE 14th Int. Conf. Cogn. Informatics Cogn. Comput. ICCI*CC 2015*, pp. 136–140, 2015, doi: 10.1109/ICCI-CC.2015.7259377.

[29] B. Li, Z. Zhao, T. Liu, P. Wang, and X. Du, "Weighted neural bag-of-n-grams model: New baselines for text classification," *COLING 2016 - 26th Int. Conf. Comput. Linguist. Proc. COLING 2016 Tech. Pap.*, pp. 1591–1600, 2016.

[30] C. Sammut and G. I. Webb, *Encyclopedia of Machine Learning*. .

[31] 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.

[32] A. K. Santra and C. J. Christy, "Genetic Algorithm and Confusion Matrix for Document Clustering," *Int. J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 322–328, 2012.

[33] C. Kim, V. Zhu, J. Obeid, and L. Lenert, "Natural language processing and machine learning algorithm to identify brain MRI reports with acute ischemic stroke," *PLoS One*, vol. 14, no. 2, pp. 1–13, 2019, doi: 10.1371/journal.pone.0212778.