# Ekspansi Fitur Pada Analisis Sentimen *Twitter* Dengan Pendekatan Metode *Word2Vec*

**Hildan Fawwaz Naufal[1], Erwin Budi Setiawan[2]**

[1,2] Universitas Telkom, Bandung
**hildanfawwazn@student.telkomuniversity.ac.id[1], erwinbudisetiawan@telkomuniversity.ac.id[2]**

**Abstrak**

**Media sosial yang kemajuannya semakin pesat memfasilitasi para pengguna media sosial untuk memberikan sebuah opini, entah itu opini positif atau negatif. Salah satu sosial media yang digunakan adalah *twitter*, dari opini yang pengguna unggah pada *twitter* tersebut akan ada kosa kata yang tidak terstruktur yang dapat di manfaatkan untuk melakukan penelitian analisis sentimen.**
**Pada penelitian ini, dilakukan percobaan untuk mengetahui pengaruh dari pembobotan TF-IDF dan pengaruh penerapan *feature expansion* dengan pendekatan *Word2Vec* pada klasifikasi SVM dan KNN. Hasil penelitian didapatkan akurasi sebesar 83.7% pada SVM dan 83% pada KNN, kemudian pengaruh penerapan *feature exspansion* dengan pendekatak *Word2Vec* menghasilkan akurasi 84% pada SVM.**

**Kata kunci : analisis sentimen, *feature expansion*, *Word2Vec*, SVM, KNN**

**Abstract**

**Social media, which is progressing rapidly, facilitates social media users to give an opinion, whether it's a positive or negative opinion. One of the social media used is twitter, from the opinions that users upload on twitter, there will be unstructured vocabulary that can be used to conduct sentiment analysis research.**
**In this study, an experiment was conducted to determine the effect of TF-IDF weighting and the effect of the application of *feature expansion* with the Word2Vec approach on SVM and KNN classification. The results of the study obtained an accuracy of 83.7% on SVM and 83% on KNN, then the effect of applying *feature expansion* with the Word2Vec approach resulted in an accuracy of 84% on SVM.**

**Keywords: sentiment analysis, *feature expansion*, *Word2Vec*, SVM, KNN.**

## 1. Pendahuluan

Kebijakan pemerintah adalah suatu keputusan yang dibuat atau dilakukan oleh pejabat pemerintah atas nama instansi yang dipimpinnya. Kebijakan pemerintah hadir dalam berbagai aspek kehidupan bermasyarakat, salah satunya adalah kebijakan publik. Kebijakan publik adalah kebijakan pemerintah yang dapat mempengaruhi setiap orang pada suatu negara atau negara bagian pada umumnya. Diharapkan dalam pembuatan kebijakan publik pemerintah bisa memberikan kebijakan yang memberi solusi dan kebaikan untuk berbagai pihak terutama masyarakat. Namun terkadang kebijakan yang di berikan mendapatkan beberapa respon yang berbeda dari masyarakat yang kemudian diutarakan di jejaring media sosial, salah satunya jejaring media sosial *twitter*. *Twitter* adalah media sosial yang bertipe *micro-blogging* yang dapat memungkinkan pengguna untuk mengirimkan dan membaca sebuah pesan berbasis teks dengan *realtime*, *twitter* jadi situs yang sering di kunjungi [1].

Dengan *twitter* menjadi sarana untuk berpendapat oleh masyarakat mengenai kebijakan publik, akan muncul opini yang berbeda – beda entah itu opini positif ataupun negatif. Oleh karena itu *twitter* dapat di manfaatkan untuk melakukan penelitian mengenai analisis sentimen. Analisis sentimen adalah bagian dari pemrosesan bahasa alami yang tugasnya mengekstraksi informasi dari dokumen atau teks, misalnya opini. Opini yang di ambil bisa berupa opini yang positif atau negatif[2], pada tahapan proses analisis sentimen terdapat ekstraksi fitur, salah satu ekstraksi fitur tersebut adalah TF-IDF. Pada penelitian ini akan digunakan ekstraksi fitur TF-IDF yang dimana TF-IDF tersebut adalah metode ekstraksi fitur yang sering di pakai [3]. TF-IDF merupakan penghitungan bobot kata yang sering digunakan saat mengambil informasi dan merupakan metode yang hasilnya efisien dan memberikan nilai yang akurat[4]. Kemudian pada tahapan proses analisis sentimen juga terdapat ekspansi fitur, salah satu ekspansi fitur tersebut menggunakan *Word2Vec* [5]. Pada penilitian ini juga akan digunakan ekspansi fitur dengan *Word2Vec* yang dimana *Word2Vec* merupakan metode *Word Embedding* yang digunakan untuk merepresentasikan sebuah kata pada bentuk vector [6], *Word2Vec* telah terbukti membawa makna semantic dan berguna dalam berbagai tugas untuk mengoptimalkan sebuah data [7], *Word2Vec* menghasilkan vector kata yang mirip sehingga memudahkan dalam pengenalan analisis sentimen [8].

Kemudian proses selanjutnya yaitu proses klasifikasi, dipenelitian ini akan digunakan metode SVM yang merupakan metode klasifikasi dengan kelebihannya untuk menentukan jarak dengan menggunakan *support vector* sehingga prosesnya bisa lebih cepat [9]. Untuk metode klasifikasi kedua adalah KNN yang merupakan klasifikasi yang proses pelatihannya cepat, sederhana dan mudah di pelajari serta efektif jika data pelatihannya besar[10].

Berdasarkan beberapa penelitian sebelumnya [11] [12] [13], penulis termotivasi untuk melakukan eksperimen untuk mengetahui pengaruh performansi penerapan teknik *feature ekspansion* dengan pendekatan

*word embedding Word2Vec* pada metode klasifikasi SVM (*Suport Vector Machine*) dan KNN (*K-Nearest Neighbor*). Pada penelitian ini masalah yang dibahas adalah bagaimana pengaruh penerapan dari pembobotan TF-IDF ?, bagaimana pengaruh dan tingkat performansi sistem setelah diterapkan teknik *feature expansion* dengan pendekatan *word embedding Word2Vec* pada model klasifikasi SVM dan KNN.

Kemudian batasan penelitian dipenelitian ini, yaitu dataset sentimen Bahasa Indonesia sebanyak 16.571 *tweet* yang bertopik kebijakan publik di Indonesia yang diambil dari hasil crawling dari *twitter*, metode klasifikasi yang digunakan SVM dan KNN.

Tujuan yang ingin dicapai dari penelitian ini adalah mengetahui pengaruh dari penerapan pembobotan TF-IDF, kemudian pengaruh penggunaan teknik *feature expansion* dengan pendekatan *word embedding Word2Vec*, mengetahui kinerja dari klasifikasi SVM dan KNN.

Setelah pada bab 1 menjelaskan pendahuluan, selanjutnya pada bab 2 akan membahas teori/studi/literatur yang mendukung atau berkaitan erat dengan penelitian ini. Bab 3 membahas teori terkait penelitian dan pemodelan sistem . bab 4 membahas hasil, analisis, evaluasi model penelitian. Lalu, pada bab 5 menjelaskan kesimpulan dan saran.

## 2. Studi Terkait

Penelitian oleh Farhan Wahyu Kurniawan dan Warih Maharan [11] menghasilkan bahwa perbedaan pada arsitektur model *Word2Vec* mempengaruhi hasil klasifikasi. Dengan hasil menggunakan model skip-gram dengan dimensi 100 memberikan hasil klasifikasi terbaik dengan nilai precision 64,4%, kemudian recall 58% dan f1-score 61,1%.

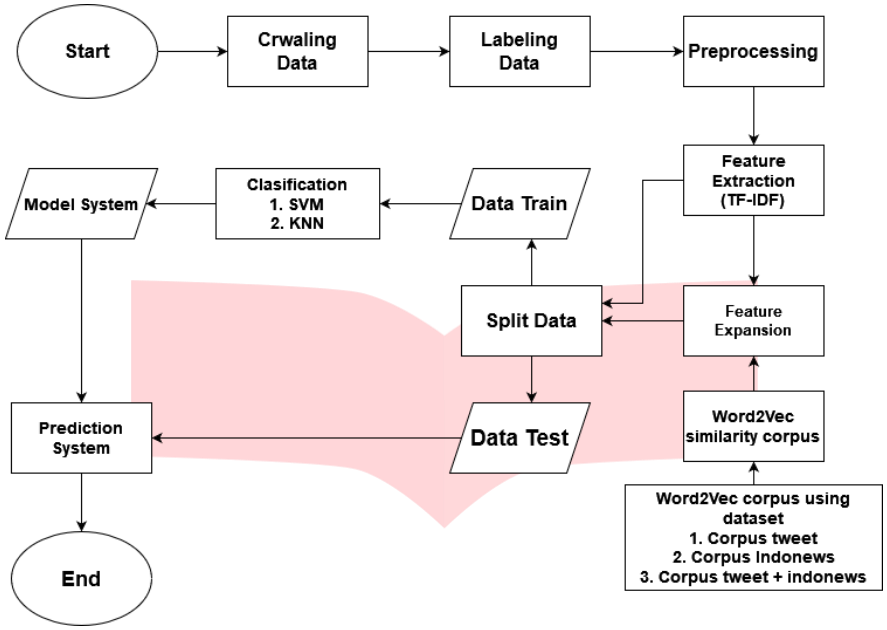
Penelitian yang di lakukan oleh Mohammad Rezwanul Huq, Ahmad Ali dan Anika Rahman berisi tentang analisis sentiment pada data *twitter* menggunakan SVM(*Support Vector Machines*) dan KNN(*K-Nearest Neighbor*) dengan hasil nilai akurasi pada KNN(*K-Nearest Neighbor*) yang hasilnya 84,32%, sedangkan untuk SVM(*Support Vector Machines*) menghasilkan nilai akurasi 77,97% [12].

Penelitian Joshua Acosta, N. Lamaute, M. Luo, E. Finkelstein, dan A. Cotoranu meliputi analisis sentimen pesan *Twitter* menggunakan metode *Word2Vec*.Penelitian ini mencoba untuk mengetahui apakah penggunaan algoritma *Word2Vec* untuk membuat *word embedding* dapat digunakan untuk mengklasifikasikan perasaan. Dengan menggunakan penyisipan kata, peneliti menghindari keharusan membuat fitur berdasarkan stilometri secara manual untuk mengklasifikasikannya dengan benar. *Tweet* tentang pengalaman maskapai dan peringkat pengguna. Karena ketidakseimbangan antara kelas sentimen, kelas kurang terwakili dalam sampel dataset kami sampai kami memiliki jumlah sampel pelatihan yang sama untuk setiap kelas. Algoritma yang digunakan termasuk *Naive Bayes*, *Logistics Regression* dan *Support Vector Machine* untuk mengklasifikasikan lebih dari 4000 *tweet* setelah melatih algoritma *Word2Vec* pada lebih dari 10.000 *tweet*. Presisi tertinggi yang dihasilkan *classifier* adalah 72% ketika *Support Vector Classifier* dan SG digunakan sebagai model pelatihan kata. Langkah kami selanjutnya adalah menyempurnakan parameter pengklasifikasi untuk meningkatkan akurasi dan menguji berbagai vektor.yang berbeda[13].

## 3. Sistem yang dibangun

Perancangan sistem proses penelitian tugas akhir ini berbentuk *flowchart*, untuk mengetahui tahapan yang di bangun dapat dilihat pada Gambar 1.

### 3.1 Gambaran Sistem




**Gambar 1 sistem analisis sentimen menggunakan *Feature expansion Word2Vec***

### 3.2 *Crawling* Data

Pertama-tama data dikumpulkan dahulu dengan cara *crawling. Crawling* data adalah pengambilan atau pengunduhan data dari database, pada penelitian ini penulis mengumpulkan data dari sever *twitter*, *Twitter* menyediakan API agar developer bisa mengakses data dari *twitter*. Yang pertama kali di lakukan adalah melakukan pendaftaran ke https://dev.*twitter*.com sebagai developer *twitter*, setelah daftar kemudian akan dilakukan otentifikasi untuk aplikasi yang akan dibuat. Otentifikasi bertujuan sebagai hak akses pengembang untuk mengunggah infoemasi dari *twitter*[14]. Data yang dikumpulkan dari hasil *crawling* sebanyak 16.571 *tweet* dengan topik kebijakan publik di Indonesia, data yang terkumpul dari hasil *crawling* terdapat beberapa *keyword* tentang kebijakan publik di Indonesia seperti #OmnibusLaw, covid-19, psbb, #uuciptakerja.

### 3.3 Pelabelan Data

Setelah data berhasil di kumpulkan kemudian dilakukan labeling dengan cara manual bersama 6 orang mahasiswa lainnya dan kemudian di tentukan sentimen pada data *tweet*, pelebelan terbagi menjadi dua yaitu *positive* ( 1 ) dan *negative* ( -1 ),, berikut contohnya :

**Tabel 1 Pelabelan Data**

| ***Tweet*** | **Label** |
|---|---|
| **UU cipta kerja banyak manfaatnya bagi pekerja dan mengakselerasi transformasi ekonomi UU cipta kerja menjawab tantangan ketenagakerjaan dan besarnya kebutuhan akan penciptaan lapangan kerja.** | 1 |
| **Duh nasib, presiden terburuk dalam sejarah.** | -1 |

### 3.4 *Preprocessing* Data

Setelah pelabelan, langkah selanjutnya adalah *preprocessing* data. *Preprocessing* data penting untuk kata atau kalimat informal dan tidak terstruktur. *Preprocessing* atau *preprocess* data adalah proses awal untuk persiapan data, *preprocessing* menghilangkan data yang kurang sesuai dan merubah data menjadi bentuk yang mudah untuk diproses. [15]. Ini beberapa Langkah penting untuk melakukan *preprocessing*, diantaranya :

1. *Filtering* : membersihkan atau menghilangkan data yang bersifat *noise* dan atasi ketidak konsistenan.
2. *Case folding* : mengubah setiap huruf besar ke kecil.
3. *Tikenization* : pemecahan kamilat menjadi kata – kata yang dipisahkan spasi.
4. *Stopword removal* adalah hapus kata kurang penting atau kurang berhubungan.
5. *Stemming* adalah proses pengubahan kata – kata yang ada menjadi kata mendasar.

### 3.5 TF-IDF

Setelah tahap *preprocessing*, data kemudian diberikan skor TF-IDF. TFIDF adalah metode penghitungan bobot kata yang sering digunakan saat mengambil informasi. Metode ini merupakan metode yang hasilnya efisien dan memberikan hasil yang akurat, metode ini menghitung nilai *Term Frequency* dan *Reverse Document Frequency* (IDF) untuk setiap kata pada dokumen korpus. [4] Berapa kali kata muncul pada dokumen menunjukkan pentingnya kata itu dalam dokumen tersebut.. Perhitungan TF-IDF didefinisikan sebagai berikut:

$$W_{ij} = tf_{ij} \times Idf_{ij}$$
$$Idf_{ij} = \left(\log\left(\frac{N}{df}\right)\right) \quad (1)$$

### 3.6 *Word2Vec*

*Word2Vec* adalah metode *Word Embedding* yang digunakan untuk merepresentasikan sebuah kata pada bentuk vektor[6]. *Word2Vec* telah terbukti membawa makna semantic dan berguna dalam berbagai tugas untuk mengoptimalkan sebuah data, *Word2Vec* juga termasuk model lanjutan *bag-of-word* dan *skipgram* serta pengoptimalan lanjutan[7]. Model *Word2Vec* menghasilkan vector untuk setiap kata dalam ruang dimensi tinggi. Arsitektur skipgram *Word2Vec* digunakan untuk melatih korpus *tweet* yang cukup lumayan besar[16]. Pada penelitian ini digunakan teknik skipgram untuk menentukan kedekatan kata secara semantik dan dalam memaksimalkan kemungkinan prediksi kata konteks atau kata sekitarnya.

### 3.7 *Feature expansion*

Untuk tahap *feature expansion* ini dilakukan setelah data *tweet* diterapkan metode *Word Embedding* yang dimana metode tersebut untuk mengatasi masalah ketidakcocokan kosakata. Disini ide yang digunakan adalah mengidentifikasi kata – kata yang hilang kemudian memformulasikan kembali dengan menambahkan kata baru yang terkait secara semantik yang telah didapatkan dari penggunaan *Word Embedding Word2Vec*. Kegunaan *Word2Vec* disini adalah untuk mengelompokan vektor dari kata – kata yang mirip menjadi satu dalam ruang vektor, dan kemudian hasil dari pengelompokan vektor dari kata – kata yang mirip tersebut berupa kamus kata yang berisi kumpulan similaritas kata. Dibutuhkan sebuah kata masukan yang kemudian akan menghasilkan satu set katak – kata yang terkait atau mirip.

Kemudian pada penelitian ini akan dilakukan perluasan fitur(*feature expansion)* pada dokumen yang telah diberi bobot nilai menggunakan TF-IDF dengan mengganti kata yang nilai bobotnya nol dengan persamaan kata yang baru yang terdapat pada daftar *Word2Vec* yang muncul pada data *tweet.* Proses ini sama dengan pada penelitian[5].

### *3.8 Support Vector Machine*

SVM adalah algoritma klasifikasi yang bekerja dengan mencari garis hyperline atau garis pemisah antar kelas yang memiliki margin atau jarak antara hyperlane dengan data terdekat di setiap kelas terbesar. Pada dasarnya SVM itu adalah upaya pencari nilai *hyperline* yang terbaik antara dua buah class[17]. Metode SVM sendiri dapat digunakan untuk mengurutkan data opini sesuai dengan atribut *rating* yang dimilikinya sehingga dapat memisahkan apakah opini tersebut termasuk dalam kelas sentimen positif atau negatif. [18].

### *3.9 K-Nearest Neighbor*

*K-Nearest Neighbor* (KNN) adalah algoritma yang mengklasifikasikan objek berdasarkan data pembelajaran yang menyerupai kemiripan paling dekat dengan objek[19]. Tujuan algoritma ini adalah mengklasifikasi objek yang baru berdasarkan atribut dan sample dari data train.Algoritma *K-Nearest Neighbor* menggunakan *Neighbothood Classification* untuk nilai prediksi dari nilai *instance* baru.

### *3.10 Confusion Matrix*

*Confusion matrix* bertujuan untuk mengukur sistem yang telah dibangun, alat visulisasi *supervised learning*. Kolom pada matriks adalah pemisalan dari kelas prediksi, beberapa baris mewakilkan kejadian pada kelas yang sebetulnya (Goronescu, 2011)

**Tabel 2 *Confusion Matrix***

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

Dari tabel tersebut terdapat :

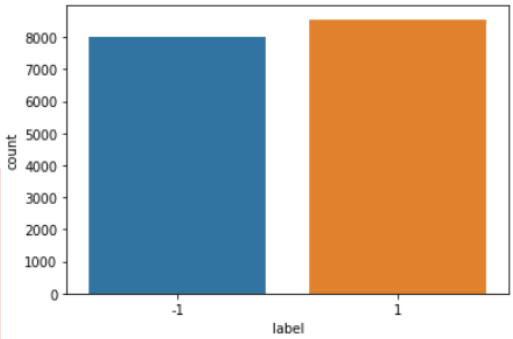
- TP = hasil dari data merupakan positif dan di prediksi dengan benar
- TN = hasil dari data merupakan negatif dan di prediksi dengan benar
- FP = hasil dari data merupakan negatif tetapi diprediksi sebagai data positif.
- FN = hasil dari data merupakan positif tetapi diprediksi sebagai data negatif

## 4. Evaluasi

Pada bagian ini dijelaskan bagaimana hasil uji dari sistem yang telah diproses.

### 4.1 Data

Data *tweet* yang diambil dari *twitter* kemudian dipakai sebanyak 16.571 *tweet* yang isi data tersebut terkait dengan kebijakan pemerintahan Indonesia, kemudian data tersebut akan dijadikan sebagai data latih dan data uji. Data tersebut terdiri dari beberapa *keyword* yang berkaitan dengan kebijakan publik seperti, #OmnibusLaw, covid-19, psbb, #uuciptakerja. Data yang memiliki label positif sebanyak 8560 sedangkan data yang memiliki label negatif sebanyak 8011. Berikut jumlah persebaran data yang sudah diberikan label:




**Gambar 2 Persebaran Data**

Lalu terdapat data pelengkap yang merupakan data yang akan digunakan untuk membuat kamus, data tersebut diambil dari beberapa media berita, jumlah datanya sebanyak 142.544. Berikut data yang digunakan untuk pembuatan kamus kata:

**Tabel 3 Data Berita**

| Nama Redaksi | Jumlah |
|---|---|
| **CNN Indonesia** | 29349 |
| **Republika** | 53812 |
| **Kompas** | 15055 |
| **Tempo** | 13702 |
| **SindoNews** | 22401 |
| **Detik.com** | 7974 |
| **Liputan6** | 251 |
| **Total** | **142544** |

### *4.2 Preprocessing*

Berikut contoh gambaran *tweet* yang telah melalui tahap *preprocessing*:

**Tabel 4 *Filtering***

| Sebelum | Sesudah |
|---|---|
| **@AdheliaCahyani8 : Pemerintah sudah melakukan dialog sosial dan dialog sosial akan dilakuakn untuk membahas peraaturan – peraturan UU Cipta Kerja. #UUCiptaKerja #OmnibusLaw** | Pemerintah sudah melakukan dialog social dan dialog social akan terus dilakukan untuk membahas peraturan turunan UU Cipta Kerja |

**Tabel 5 *Case Folding***

| Sebelum | Sesudah |
|---|---|
| **Pemerintah sudah melakukan dialog social dan dialog social akan terus dilakukan untuk membahas peraturan turunan UU Cipta Kerja** | pemerintah sudah melakukan dialog social dan dialog social akan terus dilakukan untuk membahas peraturan turunan uu cipta kerja |

**Tabel 6 *Stopword removal***

| Sebelum | Sesudah |
|---|---|
| **pemerintah sudah melakukan dialog social dan dialog social akan terus dilakukan untuk membahas peraturan turunan uu cipta kerja** | pemerintah melakukan dialog social dan akan terus dilakuakn untuk membahas peraturan uu cipta kerja |

**Tabel 7 *Stemming***

| Sebelum | Sesudah |
|---|---|
| **pemerintah sudah melakukan dialog social dan dialog social akan terus dilakukan untuk membahas peraturan turunan uu cipta kerja** | pemerintah sudah lakukan dialog social dan dialog social akan terus lakukan untuk bahas aturan turunan uu cipta kerja |

**Tabel 8 *Tokenization***

| Sebelum | Sesudah |
|---|---|
| **pemerintah sudah melakukan dialog social dan dialog social akan terus dilakukan untuk membahas peraturan turunan uu cipta kerja** | [‘pemerintah’, ‘sudah’, ‘melakukan’, ‘dialog’, ‘social’, ‘dan’, ‘dialog’ ‘social’, ‘akan’, ‘terus’, ‘dilakukan’, ‘untuk’, ‘membahas’, ‘peraturan’, ‘turunan’, ‘uu’, ‘cipta’ ,‘kerja’] |

### 4.3 Pembuatan Kamus Kata (*Corpus*)

Pembuatan kamus kata digunakan teknik *word embedding Word2Vec* model Skip Gram. Kamus kata berupa kumpulan kata yang berurutan nilai similaritasnya dari yang tertinggi sampai yang terendah. Hasil yang didapatkan dari masing-masing kamus kata dapat digambarkan seperti berikut:

1. Kamus Kata Data *Tweet*

Didapatkan hasil kosakata sebanyak 15.121 kata dan berikut contoh hasil kata-kata yang mirip:

**Tabel 9 *Corpus Tweet***

| Kata | Rank 1 | Rank 2 | Rank 3 | Rank 4 | Rank 5 |
|---|---|---|---|---|---|
| **dewan** | rampok | khianat | wakil | tipu | problematic |
| | **Rank 6** | **Rank 7** | **Rank 8** | **Rank 9** | **Rank 10** |
| | anggota | ketua | peras | lereng | dewanperwakilanrakyat |

2. Kamus Kata Data Berita

Didapatkan hasil kosakata sebanyak 225.876 kata dan berikut contoh hasil kata-kata yang mirip:

**Tabel 10 *Corpus* Berita**

| Kata | Rank 1 | Rank 2 | Rank 3 | Rank 4 | Rank 5 |
|---|---|---|---|---|---|
| **kerja** | bekerja | kerjasama | undocumented | blk | plrt |
| | **Rank 6** | **Rank 7** | **Rank 8** | **Rank 9** | **Rank 10** |
| | kemnaker | tranformasi | diakhiri | binapenta | digodog |

3. Kamus Kata Data Berita+*Tweet*

Didapatkan hasil kosakata sebanyak 230.042 kata dan berikut contoh hasil kata-kata yang mirip:

**Tabel 11 *Corpus* Berita+*Tweet***

| Kata | Rank 1 | Rank 2 | Rank 3 | Rank 4 | Rank 5 |
|---|---|---|---|---|---|
| **ciptakerja** | **mardia** | subversi | indhan | rkuhp | cipker |
| | **Rank 6** | **Rank 7** | **Rank 8** | **Rank 9** | **Rank 10** |
| | ppksk | ngesahin | juntco | contitutional | bangsamoro |

### 4.4 *Feature expansion*

Sebagai contoh *feature expansion* dengan menerapkan metode TF-IDF. Diberikan sebuah illustrasi data *tweet* "....wakil rakyat turun awas", dimisalkan sebelum ataupun sesudah kata tersebut terdapat kata "dewan" yang memiliki nilai vektor 0, kemudian kata dengan representasi 0 tersebut dilakukan pengecekan pada kamus kata yang telah dibuat, seperti pada tabel 4, jika kata "dewan" ada pada kamus kata dan memiliki similarity yang sama pada data *tweet* kata "wakil" maka akan di ganti dengan nilai bobot.

### 4.5 Klasifikasi

Setelah dilakukan proses *feature expansion* kemudian dilanjutkan dengan tahap klasifikasi menggunakan SVM dan KNN, sebelum dilakukan klasifikasi untuk mendapatkan hasil performansi dilakukan terlebih dahulu penggantian rasio data latih dan data uji terlebih dahulu untuk masing – masing algoritma, penggantian rasio tersebut bertujuan untuk mencari dan mendapatkan akurasi yang optimal, dan kemudian diambil yang paling tinggi f1-score dan akurasinya. Untuk setiap sistem klasifikasi dilakukan sebanyak 3 kali pengulangan eksekusi dan kemudian diambil rata – rata f1_score dan akurasinya.

**Tabel 12 Nilai Performa Rasio *Baseline***

| Classifier | Rasio | Akurasi (%) | F1-Score |
|---|---|---|---|
| **Baseline (Support Vector Machine)** | 90:10 | **83,4** | **0,838** |
| **Baseline (K-Nearst Neighbor)** | | **79,7** | **0,775** |
| | | Akurasi (%) | F1-Score |
| **Baseline (Support Vector Machine)** | 80:20 | 81,7 | 0,818 |
| **Baseline (K-Nearst Neighbor)** | | 77,4 | 0790 |

### 4.6 Skenario dan Hasil Pengujian

Pada penelitian ini, matriks penilaian yang digunakan, yaitu nilai akurasi dan F1-Score. Oleh karena itu, sebelum dilakukan skenario pengujian untuk membandingkan performansi pengaruh *feature expansion*, perlu diketahui hasil akurasi dan F1-Score optimal untuk *baseline* masing-masing algoritma (SVM dan KNN) Hasil akurasi untuk *baseline* kedua algoritma dapat dilihat pada tabel berikut.

**Tabel 13 Hasil Performansi pada *Support Vector Machine***

| Classifier | Rasio | Akurasi (%) | F1-Score |
|---|---|---|---|
| **Baseline (Support Vector Machine)** | 90:10 | 83,4 | 0,838 |
| **Baseline (Support Vector Machine) + TF-IDF** | | 83.7 (+0,3) | 0,841 (+0,03) |

**Tabel 14 Hasil Performansi pada K-*Nearst Neighbor***

| Classifier | Rasio | Akurasi (%) | F1-Score |
|---|---|---|---|
| **Baseline (K-Nearst Neighbor)** | 90:10 | 79,7 | 0,775 |
| **Baseline (K-Nearst Neighbor) + TF-IDF** | | 83 (+3,3) | 0,823 (+0,048) |

Selanjutnya, dilakukan skenario pengujian untuk membandingkan performansi saat menggunakan *feature expansion* dengan kamus kata data *tweet*, *feature expansion* dengan kamus kata data berita, dan *feature expansion* dengan kamus kata gabungan dari data berita dan data *tweet* (data berita+*tweet*). Pada tiap skenario, pengujian juga dilakukan pada pembagian rasio data latih dan data uji sebesar 90:10. Pengujian dilakukan untuk tiap pengambilan ukuran fitur sebanyak 1, 5, 10 kata yang mempunyai nilai similaritas tertinggi dari kamus kata yang telah dibuat. Pada tiap sistem klasifikasi, juga dilakukan pengulangan eksekusi program sebanyak 3 kali untuk diambil nilai rata-rata akurasinya.

1. Hasil Akurasi

Performansi untuk nilai akurasi *feature expansion* menggunakan algoritma klasifikasi *Support Vector Machine* dapat dilihat pada tabel 15. Nilai akurasi tertinggi didapatkan pada fitur top 5 menggunakan kamus kata data beritadengan nilai akurasi sebesar 84% dengan peningkatan akurasi terhadap *baseline* sebesar 0.6%.

**Tabel 15 Hasil Akurasi *Feature expansion* pada *Support Vector Machine***

| | Akurasi (%) | | |
|---|---|---|---|
| **Feature** | ***Corpus Tweet*** | ***Corpus* Berita** | ***Corpus* Berita+*Tweet*** |
| **Top 1** | 83.1 **(+0,0)** | **83.5 (+0,1)** | 83.4 (+0,0) |
| **Top 5** | 82.9 (+0.0) | **84 (+0,6)** | **83.8 (+0,3)** |
| **Top 10** | 83.3 (+0,0) | **83.9 (+0,5)** | **83.6 (+0,2)** |

Performansi untuk nilai akurasi *feature expansion* menggunakan algoritma klasifikasi K-*Nearest Neighbor* dapat dilihat pada tabel 16. Nilai akurasi tertinggi didapatkan pada fitur top10 menggunakan kamus kata data berita dengan nilai akurasi sebesar 83.1% dengan peningkatan akurasi terhadap *baseline* sebesar 3.4%.

**Tabel 16 Hasil Akurasi *Feature expansion* pada K-*Nearest Neighbor***

| | Akurasi (%) | | |
|---|---|---|---|
| **Feature** | ***Corpus Tweet*** | ***Corpus* Berita** | ***Corpus* Berita+*Tweet*** |
| **Top 1** | 81.9 **(+2,2)** | 82.3 (+2.6) | 81.9 (+2.2) |
| **Top 5** | 82.2 (+2.3) | 82.6 (+2.9) | 81.1 (+1.4) |
| **Top 10** | 82.2 (+2.3) | **83.1 (+3.4)** | 82.3 (+2.6) |

2. F1-Score

Performansi untuk nilai F1-Score *feature expansion* menggunakan algoritma klasifikasi *Support Vector Machine* dapat dilihat pada tabel 17. Nilai f1-score tertinggi didapatkan pada fitur top 5 menggunnakan kamus kata data berita dengan nilai f1-score sebesar 0.844 dengan peningkatan f1-score terhadap *baseline* sebesar 0.006.

**Tabel 17 Hasil F1-Score *Feature expansion* pada Support Vector Machine**

| | F1-Score | | |
|---|---|---|---|
| **Feature** | ***Corpus Tweet*** | ***Corpus* Berita** | ***Corpus* Berita+*Tweet*** |
| **Top 1** | 0,835 (+0,000) | 0,839 (+0,001) | 0,838 (+0,000) |
| **Top 5** | 0,832 (+0,000) | **0,844 (+0,006)** | 0,842 (+0,004) |
| **Top 10** | 0,837 (+0,000) | 0,843 (+0,005) | 0,839 (+0,001) |

Performansi untuk nilai F1-Score *feature expansion* menggunakan algoritma klasifikasi K-*Nearest Neighbor* dapat dilihat pada tabel 18. Nilai f1-score tertinggi didapatkan pada fitur top 10 menggunnakan kamus kata data berita dengan nilai f1-score sebesar 0.825 dengan peningkatan f1-score terhadap *baseline* sebesar 0.05.

**Tabel 18 Hasil F1-Score *Feature expansion* pada K-Nearest Neighbor**

| | F1-Score | | |
|---|---|---|---|
| **Feature** | ***Corpus Tweet*** | ***Corpus* Berita** | ***Corpus* Berita+*Tweet*** |
| **Top 1** | 0,81 **(+0.035)** | 0,815 (+0.04) | 0,811 (+0.036) |
| **Top 5** | 0,813 (+0,038) | 0,819 (+0.044) | 0,801 (+0.026) |
| **Top 10** | 0,813 (+0,038) | **0,825 (+0.05)** | 0,815 (+0.04) |

**Analisis Hasil Pengujian**

Hasil percobaan menggunakan *feature ekspansion* dengan pendekatan *Word2Vec* pada pembuatan kamus kata dan juga menggunakan fitur yang berbeda mendapatkan hasil yang bervariasi. Berdasarkan hasil f1-score dan akurasi yang didapatkan, terdapat peningkatan ketika TF-IDF sebagai pembobotan kata di terapkan, dan terjadi peningkatan pula pada f1-score dan akurasi ketika sistem mengimplementasikan *feature expansion.*

## 5. Kesimpulan

Pada penelitian ini, penerapan pembobotan dengan TF-IDF mengalami peningkatan akurasi pada masing – masing klasifikasi, SVM sebesar 83.7% (+0,3) dan KNN sebesar 83% (+3,3). Kemudian untuk implementasi metode *feature expansion* dengan pendekatan *Word2Vec* terbukti dapat meningkatkan nilai akurasi dan F1-Score pada sistem. Hasil terbaik didapat pada *corpus* berita dengan fitur top 5 dengan menggunakan klasifikasi SVM dengan nilai akurasi sebesar 84%(+0.6) dan untuk nilai f1-score, hasil terbaik didapat pada *corpus* berita dengan fitur top 5 dengan menggunakan klasifikasi SVMjuga dengan nilai f1-score sebesar 0.844(+0.006). Saran untuk penelitian selanjutnya dapat dilakukan percobaan dengan menggunakan model klasifikasi yang berbeda, dengan metode yang berbeda dan *corpus* yang digunakan bisa lebih banyak.

## REFERENSI


[1] F. Ratnawati and E. Winarko, "Sentiment Analysis of Movie Opinion in *Twitter* Using Dynamic Convolutional Neural Network Algorithm," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 1, p. 1, 2018, doi: 10.22146/ijccs.19237.

[2] M. A. Toçoğlu and A. Onan, *Satire Detection in Turkish News Articles: A Machine Learning Approach*, vol. 1054. 2019.

[3] G. Paltoglou and M. Thelwall, "A study of Information Retrieval weighting schemes for sentiment analysis," *ACL 2010 - 48th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, no. July, pp. 1386–1395, 2010.

[4] 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.

[5] 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.

[6] G. W. Aldiansyah, P. P. Adikara, and R. C. Wihandika, "Rekomendasi Lagu Cross Language Berdasarkan Lirik Menggunakan," vol. 3, no. 8, pp. 8036–8041, 2019.

[7] X. Rong, "*Word2Vec* Parameter Learning Explained," pp. 1–21, 2014, [Online]. Available: http://arxiv.org/abs/1411.2738.

[8] H. Juwiantho, E. I. Setiawan, J. Santoso, and M. H. Purnomo, "Sentiment Analysis *Twitter* Bahasa Indonesia Berbasis *Word2Vec* Menggunakan Deep Convolutional Neural Network," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, pp. 181–188, 2020, doi: 10.25126/jtiik.202071758.

[9] H. F. Chen, "In silico log p prediction for a large data set with support vector machines, radial basis neural networks and multiple linear regression," *Chem. Biol. Drug Des.*, vol. 74, no. 2, pp. 142–147, 2009, doi: 10.1111/j.1747-0285.2009.00840.x.

[10] N. Bhatia and Vandana, "Survey of Nearest Neighbor Techniques," vol. 8, no. 2, pp. 302–305, 2010, [Online]. Available: http://arxiv.org/abs/1007.0085.

[11] F. W. Kurniawan and W. Maharani, "Indonesian *Twitter* Sentiment Analysis Using *Word2Vec*," *2020 Int. Conf. Data Sci. Its Appl. ICoDSA 2020*, pp. 31–36, 2020, doi: 10.1109/ICoDSA50139.2020.9212906.

[12] M. Rezwanul, A. Ali, and A. Rahman, "Sentiment Analysis on *Twitter* Data using KNN and SVM," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 19–25, 2017, doi: 10.14569/ijacsa.2017.080603.

[13] 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.

[14] J. Eka Sembodo, E. Budi Setiawan, and Z. Abdurahman Baizal, "Data Crawling Otomatis pada *Twitter*," no. August, pp. 11–16, 2016, doi: 10.21108/indosc.2016.111.

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

[16] F. Ali, S. Ei-sappagh, L. Feng, and K. S. Kwak, "ONEMLI! - *Word2Vec* and LSTM-based Offensive Content Detection," no. January, pp. 1480–1481, 2019.

[17] O. Somantri and D. Apriliani, "Support Vector Machine Berbasis Feature Selection Untuk Sentiment Analysis Kepuasan Pelanggan Terhadap Pelayanan Warung dan Restoran Kuliner Kota Tegal," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, p. 537, 2018, doi: 10.25126/jtiik.201855867.

[18] 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.

[19] M. R. Irfan, M. A. Fauzi, T. Tibyani, and N. D. Mentari, "*Twitter* Sentiment Analysis on 2013 Curriculum Using Ensemble Features and K-Nearest Neighbor," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, p. 5409, 2018, doi: 10.11591/ijece.v8i6.pp5409-5414.