Jurnal Teknologi Informasi dan Komunikasi
Vol: 13 No 02 2022 **E-ISSN: 2477-3255**
Diterima Redaksi: 29-09-2022 | Revisi: 01-11-2022 | Diterbitkan: 26-11-2022

SVM Method with FastText Representation Feature for Classification of Twitter Sentiments Regarding the Covid-19 Vaccination Program

Mukti M Kusairi[1], Surya Agustian[2]
[1,2]Teknik Informatika, Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim,
[1,2]Jl. HR. Soebrantas RW 15, Simpang Baru, Pekanbaru, Riau, telp. (0761) 562223
e-mail: [1]11651103423@students.uin-suska.ac.id, [2]surya.agustian@uin-suska.ac.id

Abstract

Covid-19 is a virus that has a high level of spread, making the government implement a mass vaccination program throughout Indonesia. This program received a lot of responses from the public, with positive and negative opinions or comments. Currently, the public's response through social media is also an input and consideration for the government to implement a program. Therefore, this study was conducted to produce a method approach to assessing the Covid-19 vaccination program by calculating the percentage of each sentiment class. The method used is the Support Vector Machine (SVM) and the fasttext language model feature as a representation of words in the Covid-19 vaccination sentiment dataset collected from Twitter. The data used has been dataset balancing, feature selection and parameter tuning, the optimal SVM model is obtained with a composition of 2536 training data, 778 development data and testing of 400 testing data, resulting in the best value of fi-1 score of 59% with an accuracy rate of 68%. The system is quite successful in detecting sentiment in tweets compared to before.
Keywords: *sentiment classification, FastText, SVM, Covid-19 vaccine.*

Metode SVM dengan Fitur Representasi *FastText* untuk Klasifikasi Sentimen *Twitter* Mengenai Program Vaksinasi Covid-19

Abstrak

Covid-19 merupakan virus yang memiliki tingkat penyebaran yang tinggi membuat pemerintah menerapkan program vaksinasi masal diseluruh Indonesia. Program ini banyak mendapat respon dari masyarakat, dengan opini atau komentar yang bermuatan positif dan negatif. Saat ini, respon masyarakat melalui media sosial juga menjadi masukan dan pertimbangan bagi pemerintah untuk melaksanakan suatu program. Oleh sebab itu, penelitian ini dilakukan untuk menghasilkan suatu pendekatan metode untuk menilai program vaksinasi Covid-19 dengan menghitung presentase kelas sentimen masing-masing. Metode yang digunakan Support Vector Machine (SVM) dan fitur fasttext language model sebagai representasi kata-kata pada dataset sentimen vaksinasi Covid-19 yang dikumpulkan dari Twitter. Data yang digunakan sudah dilakukan balancing dataset, seleksi fitur dan tuning

https://doi.org/10.31849/digitalzone.v13i2.11531

parameter, diperoleh model SVM yang optimal dengan komposisi 2536 data training, 778 data development dan pengujian terhadap 400 data testing, menghasilkan nilai terbaik fi-1 score 59% dengan tingkat akurasi 68%. Sistem cukup berhasil mendeteksi adanya sentimen didalam tweet dibandingkan sebelumnya.

Kata kunci: *klasifikasi sentimen, FastText, SVM, vaksin Covid-19*

1. Pendahuluan

Covid-19 merupakan virus yang memiliki tingkat penyebaran tinggi dan dapat menular pada hewan dan manusia. Dikarenakan tingkat penyebaran yang tinggi, virus ini ditetapkan sebagai pandemi global. Covid-19 pertama kali teridentifikasi di Indonesia pada tanggal 02 Maret 2020 dan terus menyebar hingga saat ini terkonfirmasi jumlah kasus Covid-19 di Indonesia adalah 3.2 juta jiwa[1]. Dengan tingkat penyebaran yang tinggi membuat pemerintah menerapkan program vaksinasi masal di seluruh Indonesia berdasarkan peraturan Menteri Kesehatan Republik Indonesia No.84 Tahun 2020 dan vaksin yang akan digunakan sesuai keputusan Menteri Kesehatan No HK.01.07/Menkes/12758 yaitu *Sinovac, AstraZeneca, Sinopharm, Moderna, Novavax, Pfizer* dan merah putih. Program vaksinasi ini banyak mendapat respon dari masyarakat yang mengandung sentimen positif, negatif maupun netral. Rencana program vaksinasi harus mempertimbangkan berbagai masukan, diantaranya dengan melihat respon dan opini masyarakat[2]

Opini dan respon masyarakat terhadap vaksinasi ini berkembang cepat pada media sosial, salah satunya adalah media sosial *Twitter*. *Twitter* merupakan sebuah aplikasi yang dijadikan sebagai pilihan untuk mengutarakan respon dan opini masyarakat mengenai vaksinasi Covid-19. Opini yang disampaikan biasanya merupakan reaksi spontan dan emosional yang bisa berupa opini positif, negatif maupun netral[3]. Opini inilah yang nantinya akan dianalisa untuk mengetahui polaritasnya dengan menggunakan analisis sentimen. Sentimen merupakan pendapat dan perasaan seseorang terhadap suatu hal. Sentimen analisis mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan text mining yang bertujuan menganalisa pendapat, sentimen, evaluasi dan emosi seseorang terhadap suatu topik, produk layanan, organisasi, individu ataupun kegiatan tertentu[4]. Teknik yang sering digunakan dalam permasalahan sentimen analisis yaitu menggunakan machine learning.

Machine Learning adalah mesin yang dikembangkan untuk dapat belajar otomatis berdasarkan data yang ada untuk dilakukan pembelajaran. Algoritma yang populer pada Machine Learning untuk menangani sentimen analisis yaitu *Naive Bayes, K-Nearest Neighbors, Logistic Regression, Decision Tree* dan *Support Vector Machine.* Penelitian ini akan melakukan klasifikasi pada 3 kelas yaitu positif, negatif dan netral. Untuk melakukan klasifikasi pada 3 kelas tersebut dibutuhkan algoritma yang dapat mengklasifikasi kelas berdimensi tinggi. Alogirtma SVM cocok dalam menangani permasalahan mengklasifikasi 3 kelas karena SVM memiliki kelebihan dalam memecahkan masalah berdimensi tinggi dalam keterbatasan sample data yang ada[5].

Penelitian oleh [6] menggunakan metode SVM dan KNN untuk mengetahui performa algoritma terhadap analisis sentimen. Hasil akurasi yang digunakan pada perbandingan metode ini adalah SVM mendapatkan akurasi sebesar 81,90% dan KNN 65%. *Reccurent Neural Network (*RNN*)* untuk klasifikasi teks digunakan untuk mengklasifikasi sentimen layanan pengaduan perusahaan transportasi. Penelitian ini mendapatkan hasil dari model yang telah dibangun menggunakan *FastText language model* sebesar 82,2% sedangkan *weight word embeddings* menggunakan *Word2Vec* mendapatkan akurasi terbaik sebesar 87,5%[7].

Penelitian yang dilakukan oleh[8] menggunakan algoritma *Convulution Neural Network* dan *Fasttext Embeddings* untuk meneliti *stance classification* post kesehatan pada media sosial. Penelitian ini menggunakan Data post reaksi tentang kesehatan pada media sosial *facebook*, hasil penelitia ini didapatkan akurasi *f1 macro* model dengan *Word2Vec* sebesar 52,7% dan akurasi dari *FastText* sebesar 53,8%. Penelitian selanjutnya menggunakan algoritma SVM untuk

mencari fitur terbaik pada analisis sentimen keberlanjutan pembelajaran daring. Tujuan penelitian melakukan analisis sentimen dengan membandingkan dua seleksi fitur *term frequency* dan TF-IDF untuk memperoleh nilai *k-fold* pada *k-fold Cross validation* dan *confusion matrix.* Hasil yang didapatkan yaitu evaluasi tertinggi pada *8-fold cross validation* dengan *accuracy* sebesar 86%, *precision* sebesar 87% dan *recall* 85%[9].

Penelitian yang dilakukan oleh[10] membandingkan performa tiga metode klasifikasi yaitu *Naive Bayes, Support Vector Machine* dan k-NN. Data yang digunakan dalam penelitian tersebut adalah data Twitter. Teknik yang dilakukan pada dataset penelitian seperti *transform*, *tokenize*, *stemming*, *classification* dan lain-lain. Hasil perbandingan dari klasifikasi *tweet* dihasilkan nilai terbaik yaitu *Support Vector Machine* dengan *accuracy* 84.58%, *precison* 82.14% dan *recall* 85.82%.

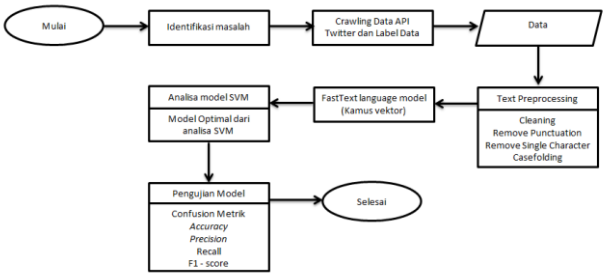
Penelitian yang terdahulu dilakukan oleh [1],[8],[9],[10] pada klasifikasi sentimen vaksin covid-19 dengan menggunakan metode *Naive Bayes*, SVM fitur TF-IDF, LSTM dan SVM fitur *Word2Vec* menghasilkan performa terbaik pada penerapan SVM fitur *Word2Vec* dengan penerapan *balancing* data, kombinasi *preprocessing* serta tuning parameter didapatkan akurasi sebesar 69% dan *f1-score* 65%. Performa setiap metode memiliki teknik dan cara masing-masing dalam penerapan *balancing* data, *preprocessing* serta tuning parameternya.

Penelitian yang dilakukan oleh [14] membandingkan kinerja dari metode pembobotan *word embeddings* seperti *Word2Vec, FastText* dan *GloVe* diklasifikasikan dengan algoritma *Convolutional Neural Network.* Ketiga metode pembobotan dipilih karena dapat menangkap makna semantik, sintatik dan urutan kata. Hasilnya kinerja *FastText* ungul dibandingkan dengan *word embeddings Word2Vec* dan *GloVe.*

Berdasarkan penelitian terkait maka penelitian ini akan melakukan klasifikasi sentimen dari 3 kelas yaitu kelas positif, negatif dan netral menggunakan metode SVM dengan melakukan Tunning parammeter menggunakan Gridsearch untuk mendapatkan model optimal dari eksperimen setiap parammeter yang ada pada SVM dan fitur ekstraksi *FastText language model* sebagai kamus vektor untuk menghasilkan performa akurasi dari dataset *Twitter* yang memiliki komposisi kelas yang tidak seimbang. Dataset yang tidak seimbang akan diterapkan teknik *balancing* pada data yang memiliki jumlah kelas dominan sehingga menghasilkan data yang proporsional.Untuk mempermudah melakukan klasifikasi sentimen digunakan bahasa pemograman *python* dengan beberapa *library* yang dibutuhkan untuk proses analisis serta menggunakan *confusion matrix* untuk melakukan evaluasi dan pengujian.

2. Metode Penelitian

Metode penelitian merupakan proses penelitian yang tersusun secara sistematis dan berurutan sehingga dapat mencapai tujuan penilitian. Berikut tahapan-tahapan metode penelitian yang akan dilakukan pada Gambar 1.



Gambar 1. Metode Penelitian

2.1. Identifikasi Masalah

Identifikasi masalah pada penelitian ini adalah mengidentifikasi sentimen masyarakat mengenai vaksinasi Covid-19 pada media sosial twitter Indonesia dengan menerapkan metode *SVM* dan fitur *FastText language model* untuk mendapatkan tingkat akurasi pada analisis sentimen.

https://doi.org/10.31849/digitalzone.v13i2.11531

2.2. Pengumpulan Data

Pengumpulan data pada penelitian ini mengunakan *crawling* mengenai data *tweet* opini masyarakat pada vaksinasi Covid-19 di Indonesia dengan menggunakan bahasa pemograman *python* dan API *Developer Twitter* dengan melakukan *crwaling* data dengan kata kunci yang relevan dengan vaksin seperti "vaksin", "vaksin covid", "vaksin haram", "vaksin import", "efek vaksin", "vaksin indonesia" dll. Data yang dihasilkan dari *crawling* dari rentang bulan Maret-April 2021 didapatkan 12000 data *tweet.* Kemudian data tersebut dilakukan pemberian label untuk setiap *tweet* yang mengandung sentimen. Pada tahapan pelabelan data *tweet* dilakukan dengan menggunakan metode *crowdsourcing* dimana pelabelan dikerjakan oleh 12 orang dibagi menjadi 4 kelompok. Untuk memastikan label yang dihasilkan sudah tepat diperlukan metode tambahan seperti *majority voting*, yaitu dengan melakukan pelabelan pada data yang sama, kemudian keputusan akhir diambil dari jumlah suara terbanyak pada label tersebut[15]. Data valid yang akan digunakan berjumlah 9178 dan dibagi kedalam 3 kelompok data yaitu data latih (Data Train) 8000 *tweet*, data validasi (Data Dev) 778 *tweet* dan data uji (Data Test) 400 *tweet*.

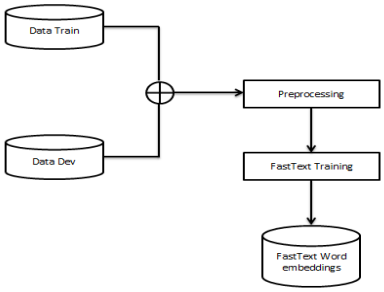
2.3. Text Preprocessing

Tahapan analisis kombinasi *text preprocessing* adalah proses seleksi data *text* untuk merubah data *text* menjadi terstruktur dengan melalui serangkaian proses. Proses *text preprocessing* yaitu menghilangkan karakter-karakter tertentu yang terkandung dalam dokumen serta mengubah huruf kapital menjadi huruf kecil[16]. Penentuan kombinasi yang akan digunakan dilakukan secara manual untuk menentukan kombinasi *text preprocessing* yang terbaik untuk menghasilkan akurasi yang terbaik. Pada penelitian ini akan dilakukan 4 jenis *text preprocessing* sebagai berikut :

1. *Cleaning*
   Pada tahapan ini menghilangkan *noise* yang berupa *emoticon* dan karakter yang yang tidak memiliki makna dalam *text* seperti : *hashtag* (*#*), *username* (*@username*), *retweet* (RT), *link* URL, dan alamat *website.*
2. *Remove Punctuation*
   Pada tahapan ini akan dilakukan penghapusan tanda baca yang ada pada Data Train. Hal ini dilakukan untuk mengurangi kata dan mempercepat proses analisa.
3. *Remove Single Character*
   Pada tahapan ini dilakukan proses untuk menghapus huruf yang berdiri sendiri atau *singgle*.
4. *Casefolding*
   Pada tahapan ini mengubahsetiap kata yang berhuruf besar menjadi kata yang berhuruf kecil, supaya nantinya tidak terjadi ambigiutas.

2.4. FastText Language Model

Untuk representasi kalimat pada *tweet*, digunakan model *word embeddings FastText* sebagaimana proses pada Gambar 2. Model ini akan mengolah kata-kata menjadi vektor berukuran 156 (atau dapat diset untuk mendapatkan model Bahasa yang optimal) melalui *library Gensim*. Sumber kalimat yang digunakan adalah berasal dari gabungan *tweet* pada Data Train dan Data Dev. Data tersebut telah melalui proses proses *preprocessing* sebagaimana dijelaskan pada bagian 2.3.

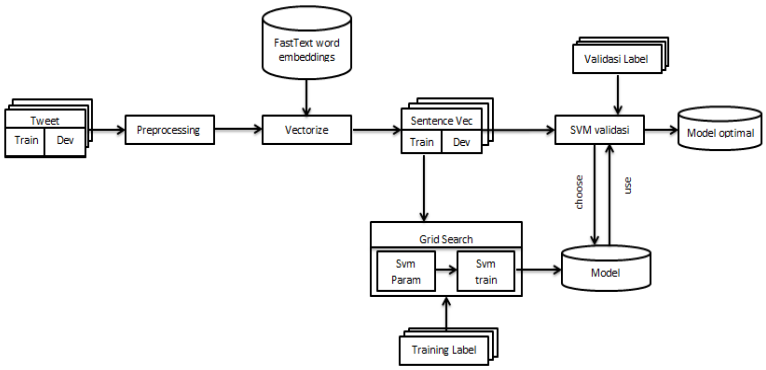


Gambar 2. Pembuatan *Word Embeddings* dengan *FastText*

2.5. Support Vector Machine

Pada tahapan ini dilakukan proses *training* untuk menguji tingkat akurasi yang dihasilkan metode SVM seperti terlihat pada Gambar 3 yang merupakan modifikasi dari penelitian[13]. Akurasi yang dihasilkan pada *training* kemudian akan di validasi untuk mendapatkan model optimal kemudian dilakukan pengujian menggunakan Data Test. Adapun tahapan pada metode *SVM* untuk penelitian yang akan dilakukan yaitu :

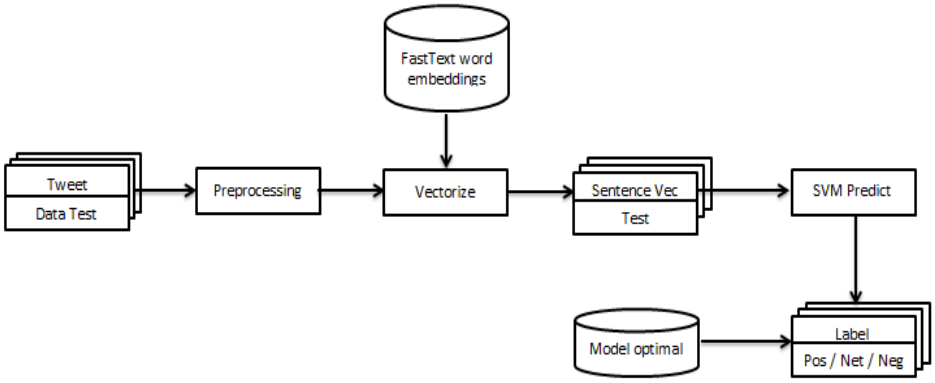
1. Tweet pada Data Train dan Data Dev diubah menjadi vektor kalimat melalui modul *vectorizer* dengan menggunakan model *FastText*.
2. Data Train digunakan untuk pelatihan model SVM. Tuning parameter SVM dilakukan dengan metode *GridsearchCV* untuk mengoptimasi kernel, nilai C dan *gamma*. Kernel yang digunakan diantaranya adalah {*RBF, Sigmoid, Polynomial*} dengan nilai C = {0.1, 1, 10} serta nilai *gamma* = {10, 1, 0.1, 0.01}.
3. Memilih 3 model terbaik dari hasil tuning parameter, yaitu berdasarkan nilai *f1-score* tertinggi yang diukur dari nilai presisi hasil prediksi terhadap *Train label (gold standard)*.
4. Model tersebut digunakan untuk proses validasi. Proses SVM *predict* dijalankan untuk ketiga model terbaik terhadap Data Dev untuk menetapkan model optimal berdasarkan *f1-score* tertinggi terhadap *validation label.*
5. Model optimal digunakan untuk mengklasifikasi Data Test yang tidak pernah terlihat selama proses *training* SVM maupun pemodelan *word embeddings* dengan *FastText*.



Gambar 3. Proses pencarian model optimal

2.6. Pengujian Model

Skema pengujian dalam penelitian ini diterangkan seperti Gambar 4 merupakan hasil dari modifikasi penelitian[13]. Untuk membentuk vektor kalimat (*sentence vector)* pada *tweet*, digunakan modul *vectorizer* yang membaca model *FastText* dari hasil *training* pada bagian sub bab 2.4. Input vektor ini akan diprediksi oleh SVM menggunakan model yang sudah dipilih pada sub bab 2.5 untuk menentukan apakah memiliki sentiment positif, negatif atau netral



Gambar 4. Proses pengujian model optimal

https://doi.org/10.31849/digitalzone.v13i2.11531

3. Hasil dan Pembahasan

Pada tahapan ini memiliki proses untuk mendapatkan model optimal dengan proses vektorisasi menggunakan *FastText language model*, Data *balancing* pada Data Train, text preprocessing, tuning parameter dan pengujian model.

3.1. Metrik Pengukuran

Hasil pengujian ini akan dievaluasi berdasarkan *confusion matrix* untuk melihat performa sistem. Evaluasi yang akan digunakan pada *confusion matrix* yaitu *accuracy* untuk mengambarkan keakuratan model dalam mengklasifikasikan dengan benar, *precision* untuk mengambarkan tingkat akurasi data yang diminta dengan data yang dihasilkan model, *recall* untuk mengambarkan keberhasilan model dalam menemukan informasi dan *f1-score* untuk membandingan antara *presisi* dan *recall.*

3.2. FastText Language Model

Setelah *FastText language model* dibangun menggunakan Data Train dan Data Dev didapatkan kamus vektor. Kamus vektor ini digunakan untuk vektorisasi pada *tweet* Data Train, Dev dan Test untuk menjadi inputan pada SVM dengan mengubah *tweet* menjadi vektor dari setiap kalimat yang pada dataset. Vektorisasi ini dilakukan dengan cara menghitung setiap vektor pada kamus vektor dan dilakukan penjumlahan sesuai kalimat pada data *tweet*. Berikut contoh vektorisasi pada kalimat:

Kamus *Word embeddings FastText language model* :

Kamus Vektor = [[0.8377928, -0.9451819, 1.9583086, ..., -0.46919054],
[3.7356644, -0.12420304, 2.8183975, ..., -0.01584086],
[-3.9044623, -1.4495498, 8.719044, ..., 3.0040727],
.....,
[2.8155358, -14379541, 0.30571818, ..., -0.7690656],
[1.9499773, 1.4766246, 0.6803053, ..., 0.93774927],
[-0.83477825, -0.08699539, 2.7890635, ..., 2.4560118]]

Vaksin = [0.8377928, -0.9451819, 19583086, ..., -0.46919054]

Covid = [-0.5673201, -0.57411414, 2.1817667, ..., 0.38184023]

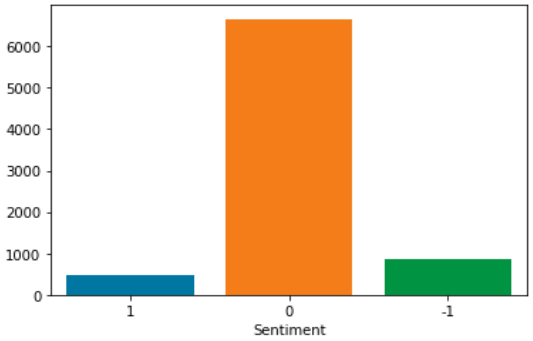
Itu = [0.75396675, 0.12176714, 0.82283074, ..., 2.2017808]

Haram = [0.14348589, -0.8766837, -0.9937711, ..., 0.74614257]

Vaksin Covid Itu Haram
= [7.25111812, -145568117, 6.38615966, ..., 2.72075117]

3.3. Data Balancing

Komposisi Data Train sebanyak 8000 tweet terdiri dari label positif, negatif dan netral dengan statistik pada Gambar 5 berikut.



Gambar 5. Grafik komposisi label Data Train

Komposisi kelas yang tidak proposional tersebut akan mempengarhui kemampuan model SVM dalam mendeteksi adanya kelas positif, negatif dan netral, karena data yang mayoritas berlabel netral (akurasi *training* bisa mencapai 80%). Sehingga, akurasi akan terlihat tinggi padahal sistem hanya mendeteksi kelas netral, dan tidak dapat mendeteksi kelas positif dan negatif. Data Train yang tidak proposional memiliki bobot kelas yaitu 6664 netral, 873 negatif dan 463 positif dan akan dilakukan proses *balancing* dengan mengurangi jumlah kelas tertinggi.

Oleh sebab itu, perlu dilakukan normalisasi Data Train dalam bentuk *balancing* antara kelas netral dan kelas yang memiliki sentimen positif atau negatif. Proses *balancing* dilakukan dengan menggunakan *baseline* dari metode SVM kernel *RBF*, nilai C = 1 dan nilai gamma 0.01. Sebelum dilakukan proses *balancing* data diseleksi dengan mengambil data yang memiliki panjang lebih besar dari 6 untuk menghasilkan data yang memiliki makna. Berikut Tabel 1 memperlihatkan hasil perbandingan komposisi kelas untuk menghasilkan data *balancing*.

Tabel 1. Hasil Validasi F1-score

Kelas	Jumlah	Precision	Recall	F1-score
Jumlah Data Netral 1000 tweet				
Negatif	873	0.31	0.65	0.42
Netral	1000	**0.92**	0.79	0.85
Positif	463	0.40	0.42	0.41
				56%
Jumlah Data Netral 1200 tweet				
Negatif	873	0.35	0.58	**0.44**
Netral	1200	0.91	0.85	**0.88**
Positif	463	0.50	0.38	**0.43**
				58%
Jumlah Data Netral 1500 tweet				
Negatif	873	0.38	0.45	0.41
Netral	1500	0.89	0.89	0.89
Positif	463	0.46	0.36	0.40
				57%
Jumlah Data Netral 2656 tweet				
Negatif	873	0.65	0.20	0.31
Netral	2656	0.86	0.98	0.92
Positif	463	0.77	0.22	0.34
				0.52
Jumlah Data Netral Awal 6621				
Negatif	873	0.00	0.00	0.00
Netral	6621	0.83	**1.00**	**0.91**
Positif	463	0.00	0.00	0.00
				0.30

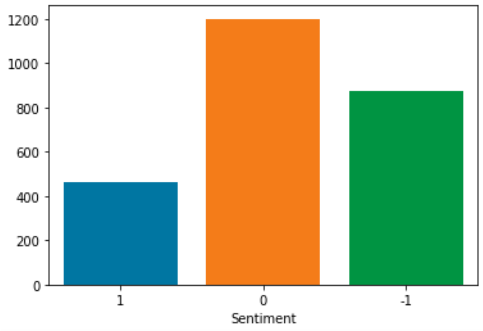
Hasil pengujian yang dihasilkan dari perbandingan komposisi kelas pada Data Train didapatkan komposisi data *balancing* terbaik yang memiliki kelas negatif 873, netral 1200 dan positif 463 terlihat pada Tabel 1 menghasilkan *f1-score* lebih baik dengan nilai 0.58%. Data yang sudah *balancing* terlihat pada Gambar 6 ini nantinya akan digunakan untuk proses pencarian kombinasi *text preprocessing.*

3.4. Text Preprocessing

Data Train yang telah dilakukan *balancing* kemudian diolah dengan menggunakan *text preprocessing.* Pengolahan *text preprocessing* pada penelitian ini akan mengacu pada pegalaman hasil optimal yang dicapai pada penelitian [1],[8],[9],[10] dengan menerapkan *cleaning, remove singgle character, remove punctuation, stopwords removal, stemming* dan *casefolding.* Pada penelitian ini mendapatkan performa terbaik dengan menggunakan kombinasi *remove punctuation*,

https://doi.org/10.31849/digitalzone.v13i2.11531

remove single character dan *casefolding*. Hasil yang didapatkan dari kombinasi terbaik *text preprocessing* ini akan dijadikan model optimal. Berikut Tabel 2 hasil pengujian kombinasi *text preprocessing* :



Gambar 6. Grafik komposisi terbaik Data Train *balancing*

Tabel 2. Hasil validasi F1-score berdasarkan kombinasi *text preprocessing*

RPUNC	RSC	CASFOLD	F1-Score
Ya	**Ya**	**Ya**	**0.56%**
Ya	Tidak	Ya	0.55%
Tidak	Ya	Ya	0.54%
Tidak	Tidak	Ya	0.54%
Tidak	Ya	Tidak	0.54%
Ya	Tidak	Tidak	0.55%

3.5. Tuning Parameter

Tuning parameter dilakukan untuk meningkatkan performa model *machine learning* yang di bangun. Peningkatan performa ini dibutuhkan untuk mengetahui parameter terbaik untuk model. Optimasi parameter pada SVM dilakukan menggunakan *GridsearchCV*. Adapun parameter SVM yang digunakan sebagai berikut :

Kernel = { *RBF, Sigmoid, Polynomial* }

C = {0.1, 1, 10}

gamma = {10, 1, 0.1, 0.01}

Pelatihan model SVM ini menggunakan Data Train dengan mencari parameter terbaik yang divalidasi menggunakan Data Dev untuk mendapatkan performa model yang optimal. Berikut hasil dari tuning parameter yang dilakukan pada SVM pada Tabel 3 berikut.

Tabel 3. Hasil *tuning parameter SVM**

Kernel	C	gamma	F1-score
RBF	0.1	10	0.30%
		1	0.30%
		0.1	0.30%
		0.01	0.41%
	1	10	0.30%

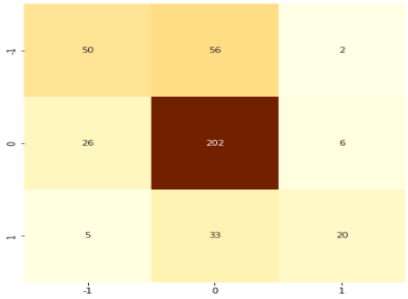
		1	0.30%
		0.1	0.30%
		0.01	**0.56%**
	10	10	0.30%
		1	0.30%
		0.1	0.30%
		0.01	0.51%

**Kernel sigmoid dan polynomial tidak ditulis dalam table, karena hasilnya lebih rendah dari RBF*

Parameter terbaik yang dihasilkan yaitu kernel *RBF* dengan nilai $C = 1$ dan nilai *gamma* = 0.01 menghasilkan *f1-score* sebesar 56% kemudian parameter tersebut digunakan sebagai model optimal yang akan digunakan sebagai pengujian model menggunakan Data Test.

3.6. Pengujian Model

Hasil pengujian ini akan dievaluasi berdasarkan *confusion matrix* untuk melihat performa sistem. Nilai *f1-score, precision, recall* dan *accuracy* akan dihitung menggunakan *library sklearn*. Model optimal yang dihasilkan dari tuning parameter dilakukan pengujian menggunakan Data Test sebanyak 400 *tweet*. Model optimal dapat mendeteksi kelas 202 netral, 50 negatif dan 20 positif pada Data Test yang belum pernah diketahui oleh model menghasilkan akurasi sebesar 0.68%, Dalam Hasil pengujian dapat dilihat pada diagram *confusion matrix* pada Gambar 7.



Gambar 7. *Confusion matrix*

Hasil yang didapatkan dari *confusion matrix* pada gambar 7 dapat dilihat pada tabel 4 berikut.

Tabel 4. Performa dan hasil dari pengujian

Metode	*Accuracy*	*Precision*	*Recall*	*F1-Score*
SVM + *FastText*	0.68	0.68	0.56	0.59

Pada tabel 4 dapat dilihat hasil pengujian dari metode SVM dengan *fitur FastText* menghasilkan nilai *accuracy* 0.68, *Precision* 0.68, *Recall* 0.56 dan *F1-score* 0.59.

3.7. Perbandingan Pengujian Metode Dahulu

Dilihat dari hasil *confusion matrix* pada penelitian sebelumnya menggunakan Dataset dengan kasus yang sama mengacu pada pengujian *accuracy, precision, recall, dan f1-score* menghasilkan performa yang tidak jauh berbeda dengan penelitian sebelumnya yang menggunakan metode *Naive Bayes*[11] , SVM dengan *fitur* TF-IDF[1] , LSTM dengan *Word2Vec*[12] dan SVM dengan *Word2Vec* [13]. Berikut hasil performa dan yang telah dicapai pada penelitian sebelumnya pada tabel 5.

Tabel 5. Perbandingan Performa Penelitian sebelumnya

Metode	***Accuracy***	***Precision***	***Recall***	***F1-Score***
Naive Bayes [11]	0.61	0.58	0.60	0.57
SVM + TF-IDF[1]	0.65	0.61	0.54	0.56
LSTM + *Word2Vec*[12]	0.66	**0.75**	0.53	0.54

https://doi.org/10.31849/digitalzone.v13i2.11531

SVM + *Word2Vec*[13]	**0.69**	0.69	**0.63**	**0.65**
SVM + *FastText* (Penelitian ini)	0.68	0.68	0.56	0.59

Performa setiap metode memiliki hasil yang berbeda, dari segi akurasi SVM dengan FastText lebih kompetitif dibandingan dengan penelitian sebelumnya yang menggunakan metode *Naive Bayes* [11], SVM dengan TF-IDF[1], dan LSTM [12] namun lebih rendah dari SVM dengan *Word2Vec.* Pada *f1-score* metode SVM dengan *fitur FastText* memiliki *f1-score* lebih baik dari kompetitornya *Naive Bayes* dengan *f1-score* 57% dan SVM *fitur* TF-IDF *f1-score* 56% dan LSTM *f1-score* 54% . Dengan demikian SVM menggunakan *fitur FastText* menghasilkan performa lebih baik pada metode *machine learning* seperti *Naive bayes,* SVM *fitur* TF-IDF dan LSTM.

4. Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan menggunakan metode SVM dengan *FastText* dapat meningkatkan performa SVM dibandingkan menggunakan TF-IDF. Proses *balancing* data pada kelas yang dominan di Data Train dapat meningkatkan performa *machine learning* serta penentuan parameter juga mempengaruhi performa model *machine learning* yang dibangun. Parameter terbaik dari tuning parameter yang dihasilkan adalah kernel *RBF* dengan nilai C = 1 dan *gamma* = 0.01 menghasilkan akurasi sebesar 68% dan *f1-score* 59%.

Pada kasus ini SVM menggunakan *FastText* mempunyai performa yang lebih baik dibandingkan dengan *Naive bayes,* SVM TF-IDF, dan LSTM *Word2Vec* tetapi lebih rendah dari SVM dengan *Word2Vec*. Perbedaan performa dari SVM menggunakan *Word2Vec* dan *FastTex* dapat dilihat dari jumlah dimensi vektor yang digunakan, *Word2Vec* menggunakan panjang vektor 300 sedangkan *FastText* menggunakn panjang vektor 156 serta komposisi data *balancing* yang berbeda. Saran untuk penelitian selanjutnya adalah mengembangkan *training model FastText* dengan jumlah Dataset lebih besar yang memilki kelas seimbang dan melakukan *pretrain model* yang sudah dikembangkan oleh *Facebook* supaya representasi kata model lebih maksimal serta mencari panjang vektor terbaik untuk mendapatkan model Bahasa yang optimal. Selain itu dapat pula melakukan pengembangan dari *text preprocessing* lainnya.

Daftar Pustaka

[1] M. Rizky, Vaksin Covid-19 Menggunakan Metode Support Vector Machine Pada Media Sosial Twitter Covid-19 Menggunakan Metode Support Vector. 2021.

[2] F. F. Rachman and S. Pramana, "Analisis Sentimen Pro dan Kontra Masyarakat Indonesia tentang Vaksin COVID-19 pada Media Sosial Twitter," Heal. Inf. Manag. J., vol.8,no.2,pp.100–109,2020,[Online].Available: https://inohim.esaunggul.ac.id/index.php/INO/article/view/223/175.

[3] D. Inayah and F. L. Purba, "Implementasi Social Network Analysis Dalam Penyebaran Informasi Virus Corona (Covid-19) Di Twitter," Semin. Nas. Off. Stat., vol. 2020, no. 1, pp. 292–299, 2021, doi: 10.34123/semnasoffstat.v2020i1.573.

[4] N. Yunita, "Analisis Sentimen Berita Artis Dengan Menggunakan Algoritma Support Vector Machine dan Particle Swarm Optimization," J. Sist. Inf. STMIK Antar Bangsa, vol. 5, no. 2, pp. 104–112, 2016.

[5] D. H. Anto Satriyo Nugroho, Arief Budi Witarto, "Support vector machine teori dan Aplikasinya dalam Bioinformatika," Mach. Learn., pp. 1–11, 2003.

[6] T. S. Sabrila, V. R. Sari, and A. E. Minarno, "Analisis Sentimen Pada Tweet Tentang Penanganan Covid - 19 Menggunakan Word Embedding Pada Algoritma Support Vector Machine Dan K - Nearest Neighbor," vol. 6, no. 2, 2021.

[7] M. D. Rhman, A. Djunaidy, and F. Mahananto, "Penerapan Weighted Word Embedding pada Pengklasifikasian Teks Berbasis Recurrent Neural Network untuk Layanan Pengaduan Perusahaan Transportasi," vol. 10, no. 1, 2021.

[8] E. Lim, T. I. Istts, E. I. Setiawan, and T. I. Istts, "Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning," pp. 65–73, 2019.

[9] A. P. Natasuwarna, "Seleksi Fitur Support Vector Machine pada Analisis Sentimen Keberlanjutan Pembelajaran Daring," Techno.Com, vol. 19, no. 4, pp. 437–448, 2020, doi: 10.33633/tc.v19i4.4044.

[10] S. Hikmawan, A. Pardamean, and S. N. Khasanah, "Sentimen Analisis Publik Terhadap Joko Widodo terhadap wabah Covid-19 menggunakan Metode Machine Learning," J. Kaji. Ilm., vol. 20, no. 2, pp. 167–176, 2020, doi: 10.31599/jki.v20i2.117.

[11] P. (Universitas I. N. S. S. K. R. Yohana, "Analisis Sentimen Masyarakat Terhadap Kebijakan Pemerintah Indonesia Dalam Memberikan Vaksin Covid-19 Menggunakan Metode Naive Bayes Classifier." 2022.

[12] M. Ihsan et al., "LSTM (Long Short Term Memory) for Sentiment COVID-19 Vaccine Classification on Twitter 1,2,3," pp. 79–89, 2022.

[13] M. Sahbuddin and S. Agustian, "JITE (Journal of Informatics and Telecommunication Engineering) Support Vector Machine Method with Word2vec for Covid-19 Vaccine Sentiment Classification on Twitter," vol. 6, no. July, pp. 288–297, 2022.

[14] A. Nurdin, B. Anggo Seno Aji, A. Bustamin, and Z. Abidin, "Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks," J. Tekno Kompak, vol. 14, no. 2, p. 74, 2020, doi: 10.33365/jtk.v14i2.732.

[15] A. Rachmat and Y. Lukito, "Implementasi Sistem Crowdsourced Labelling Berbasis Web dengan Metode Weighted Majority Voting," J. Ultim. InfoSys, vol. 6, no. 2, pp. 76–82, 2016, doi: 10.31937/si.v6i2.223.

[16] A. Aziz, R. Saptono, and K. P. Suryajaya, "Implementasi Vector Space Model dalam Pembangkitan Frequently Asked Questions Otomatis dan Solusi yang Relevan untuk Keluhan Pelanggan," Sci. J. Informatics, vol. 2, no. 2, p. 111, 2016, doi: 10.15294/sji.v2i2.5076.