# ANALISIS SENTIMEN VAKSIN SINOVAC MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR (KNN) DAN SUPPORT VECTOR MACHINE (SVM)

**Muh. Mradan Syamsi1, Muhammad Fadhil Rahman2, Muhammad Yusuf, S. Kom., M.Kom3**

1,2,3Prodi Teknik Informatika, Universitas Muhammadiyah Sorong

e-mail: [\*1mardanmandhom@gmail.com](mailto:*1mardanmandhom@gmail.com), [2@fadhilrahamangmail.com](mailto:2@fadhilrahamangmail.com), [3yusuf@um-sorong.ac.id](mailto:3yusuf@um-sorong.ac.id)

***Abstrak***

*Pandemi COVID-19 yang bermula di Wuhan, China kini menjadi pandemi di seluruh dunia semua negara dunia. Upaya vaksinasi dilakukan untuk menekan tingkat penularan virus COVID-19. Perbedaan efek vaksinasi menimbulkan perbedaan pendapat untuk vaksin ini. Analisis sentimen dapat digunakan untuk menganalisis opini public untuk vaksin ini. Penelitian ini menggunakan algoritma SVM dan KNN Lakukan analisis sentimen masyarakat terhadap vaksinasi. Adapun pendapat Gunakan kata kunci sinovac untuk mendapatkannya dari aplikasi twitter. Dataset yang digunakan adalah tweet dalam bahasa Inggris. Proses pelabelan teks otomatis menggunakan textblob. Hasil Penelitian menunjukkan bahwa algoritma SVM memiliki kinerja yang lebih baik dibandingkan dengan dengan algoritma KNN. Akurasi algoritma SVM adalah 0,7, sedangkan akurasi algoritma KNN 0,56.*

***Kata kunci****—* Sentiment, SVM, KNN, textblob, sinovac

# PENDAHULUAN

COVID-19 adalah penyakit yang disebabkan oleh Disebabkan oleh virus corona. wabah coronavirus dimulai Desember 2019 di Wuhan, Cina. Tapi sekarang coronavirus baru menjadi pandemi di banyak negara di dunia[1]. salah satu cara untuk mengurangi COVID-19 menyebar melalui vaksinasi. Vaksinasi adalah suntikan vaksin (antigen) digunakan untuk merangsang sistem kekebalan tubuh. Vaksinasi berpengaruh beda, beda pendapat untuk vaksin ini. Beberapa memiliki pendapat Ada yang positif, ada yang negatif, ada yang negatif opini netral[2]. Vaksinasi hangat Dibahas di berbagai media sosial, no Kecuali Twitter. Dilihat dari berbagai balasan Bisa menganalisa emosi netizen di twitter sehingga tipe opini dapat dikelompokkan menjadi positif, negatif atau netral. Ada penelitian tentang analisis sentiment Gunakan metode berikut untuk tema covid-19 Klasifikasi seperti KNN, Naive Bayesian, SVM, LSTM-RNN dan Backpropagation[1]. Menurut penelitian F.S. Akhirnya, algoritma SVM memiliki nilai akurasi yang lebih baik jika Dibandingkan dengan Algoritma Naive Bayes Dan KNN dalam analisis sentimen. Ada Penelitian ini menggunakan data tweet Indonesia. Menurut penelitian Algoritma KNN memiliki kinerja yang baik dalam aspek-aspek berikut Analisis Sentimen dalam Bahasa Indonesia. Berdasarkan uraian di atas, penulis Diusulkan untuk menggunakan algoritma SVM dan KNN Dalam analisis sentimen vaksin Kexing, dengan data tweet dalam bahasa Inggris.[3][4][5] [6]

# METODE PENELITIAN

Objek yang digunakan dalam penelitian ini adalah Tweet pengguna diambil dari twitter. twit ini Merangkak menggunakan pustaka python Namanya Tweeby. Adapun metode penelitian Seperti yang ditunjukkan di bawah ini:



Gambar 2.1 Diagram Metode Penelitian

## pengumpulan data

Pengumpulan dengan tools tweepy, menggunakan keyword: sinovac tanpa retweet. Data tweet merupakan cuitan berbahasa inggris. Dari hasil crawling didapatkan 2105 data. Semua data tweet dilakukan proses cleansing, yakni penghilangan hastag, user dan hyperlink.

* 1. Pelabelan data

Proses pelabelan dilakukan secara otomatis menggunakan textblob. Text blob akan menghitung nilai polarity dan subjectivity. Polarity merupakan fungsi untuk melihat kecondongan sentimen sebuah teks. Sedangkan subjectivity merupakan fungsi untuk melihat value dari sebuah teks. Jenis value dari teks ini bisa berupa ebuah opini atau fakta. Semakin tinggi subjectivity sebuah teks, maka teks tersebut dapat dikatakan sebagai sebuah opini, sedangkan semakin tinggi polarity maka menandakan sebuah emosi yang positif. Dari nilai polarity, maka akan didapatkan 3 kelas, yakni positif, negatif dan netral.

* 1. Data Preprocessing

Preprocessing dilakukan dengan beberapa tahap antara lain: casefolding, tokenizing, stopword removal dan stemming. Proses casefolding dilakukan dengan mengubah teks menjadi lowercase, menghilangkan karakter angka, dan menghilangkan beberapa karakter yang tidak diperlukan seperti “:, \_,=,+” dll. Proses tokenizing dilakukan untuk mengubah sebuah kalimat menjadi kata/token/term. Proses stopword removal merupakan proses untuk menghilangkan kata sambung seperti: and, with, the, dll. Stemming merupakan proses mengubah sebuah kata menjadi kata dasar. Misal: programming diubah menjadi program.

* 1. Pembobotan

Tahap berikutnya adalah menghitung bobot dari setiap token/term berdasarkan frekuensi kemunculan term tersebut dalam document menggunakan metode TF-IDF.

* 1. Proses Klasifikasi

Proses Klasifikasi dilakukan dengan menggunakan algoritma SVM dan KNN. Sebelum proses klasifikasi, data dibagi menjadi 2 bagian, yakni: data training dan data testing, dengan komposisi pembagian data testing sebesar 20% dari total dataset yang ada. Model SVM dan KNN dibangun menggunakan data training, kemudian dievaluasi menggunakan data testing.

1. Algoritma SVM

SVM merupakan algoritma klasifikasi yang dilakukan dengan menentukan hyperplane. Hyperplane yang bagus akan berada tepat ditengahtengah kedua kelas, sehingga memiliki jarak yang paling jauh ke data-data terluar di kedua kelas[7].

1. Algoritma KNN

Algoritma KNN merupakan algoritma klasifikasi yang mengelompokkan data baru berdasarkan jarak data baru ke beberapa data atau tetangga terdekat[5].

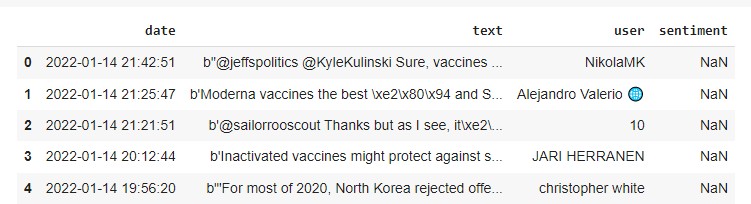
* 1. Evaluasi

Evaluasi dilakukan dengan menghitung nilai confusion matrix. Confusion matrix akan menghitung nilai accuracy, precision, recall, dan F1-Score. Hasil perhitungan akan divalidasi menggunakan K-Fold Cross Validation.

3. HASIL DAN PEMBAHASAN

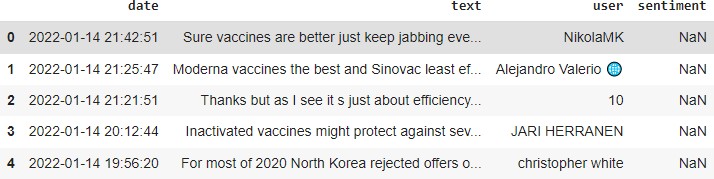
Implementasi sistem dan Analisa hasil dilakukan dengan menggunakan bahasa pemrograman *python*. Adapun implementasi setiap tahapannya adalah sebagai berikut:

* 1. Pengumpulan Data

Untuk dapat menggunakan *tweepy*, maka kita perlu memiliki *API Key* dan *Access Token* terlebih dahulu. Data yang telah didapatkan kemudian disimpan dalam format.csv. Data mentah hasil *crawling* ditunjukkan oleh gambar 3.1 berikut ini:

Gambar 3.1 Data Mentah

Setelah itu dilakukan proses *cleansing* dengan menghilangakan *hastag*, *user* dan *hyperlink*. Hasil proses *cleansing* ditunjukkan oleh gambarl 3.2 berikut ini:



Gambarl 3.2 Hasil *Cleansing*

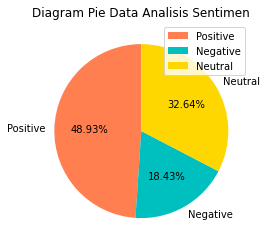
* 1. Pelabelan data

Pelabelan data dihitung dengan menggunakan *textblob* untuk mendapatkan nilai *polarity* dan subjectifity. Nilai *polarity* <0, akan dilabeli sebagai sentiment *negative*, untuk *polarity* =0 akan dilabeli sebagai sentimen netral dan untuk *polarity*>0 akan dilabeli sebagai sentimen positif. Adapun hasil pelabelan ditunjukkan oleh gambar 3.3 berikut ini.



Gambarl 3.3 Hasil Pelabelan

Komposisi hasil pelabelan, didapatkan data sebagai berikut:



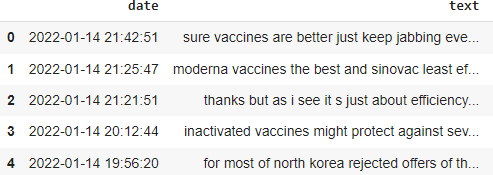
Gambar 3.4 Diagram Pie Hasil pelabelan

* 1. *Data Preprocessing*

*Data preprocessing* dilakukan dengan beberapa tahap antara lain:

1. *Casefolding*

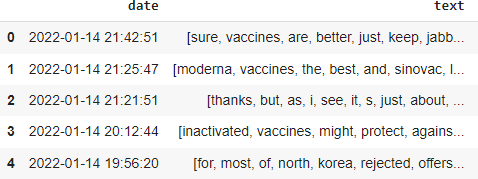
Hasil proses *casefolding* ditunjukkan oleh tabel berikut ini gambar 3.5 berikut ini:



Gambar 3.5 Hasil *Casefolding*

1. *Tokenizing*

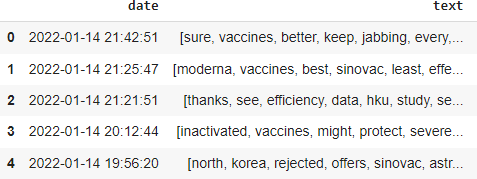
Proses *tokenizing* digunakan untuk membagi kalimat kedalam *token*/kata/*term*. Adapun hasil proses *tokenizing* ditunjukkan oleh gambar 3.6 berikut ini:



Gambar 3.6 Hasil *Tokenizing*

1. *Stopword Removal*

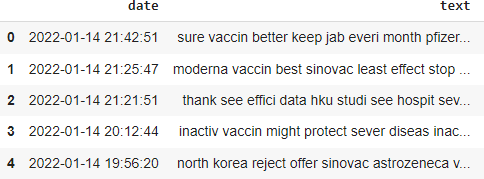
Proses s*topword removal* merupakan proses penghapusan kata hubung seperti *the, for, are, of* dll. Hasil proses *stopword removal* ditunjukkan oleh gambar 3.7 berikut ini.



Gambar 3.7 Hasil *Stopword Removal*

1. *Stemming*

Proses *stemming* dilakukan dengan menggunakan algoritma porter. Algoritma ini digunakan untuk mengubah *token* menjadi sebuah kata dasar. Hasil dari proses *stemming* ditunjukkan oleh Gambar 3.8 berikut ini:

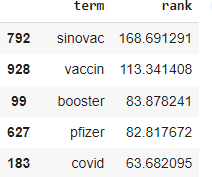


Gambar 3.8 Hasil *Stemming*

* 1. Pembobotan

Proses pembobotan kata digunakan dengan algoritma TF-IDF. Bobot sebuah kata akan dinilai berdasarkan frekuensi kemunculan kata tersebut dalam sebuah kalimat ataupun dalam sebuah dokumen.

Hasil Pembobotan TF-IDF dengan nilai bobot terbesar kata ditunjukkan oleh gambar 3.9 berikut ini:



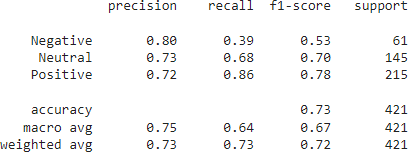
Gambar 3.9 *Top Rank-TF-IDF*

* 1. Proses Klasifikasi dan Evaluasi Hasil

Proses Klasifikasi dilakukan dengan algoritma SVM dan KNN. Pembuatan model klasifikasi dilakukan dengan library *sckitlearn*. *Data training* yang digunakan sebesar 80%, dan d*ata testing* yang digunakan untuk evaluasi sebesar 20 %.

1. Penerapan Algoritma SVM

Hasil Klasifikasi dengan menggunakan algoritma SVM ditunjukkan oleh *confusion matrix* berikut ini:



Gambar 3.10 *Confusion matrix* hasil SVM

Dari tabel tersebut, didapatkan hasil akurasi sebesar 73%. Hasil pengujian kemudian dievaluasi menggunakan *K-Fold Cross Validation*, dengan jumlah *fold* sebesar 10 *fold*. Dari hasil validasi didapatkan rata-rata akurasi sebesar: 0.70

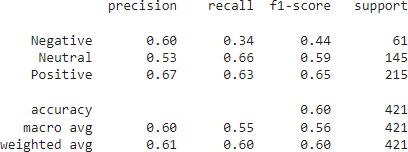
Pengujian juga dilakukan dengan memakai beberapa jenis kernel, antara lain sebagai berikut:

Tabel 3.1 Pengujian Ragam *Kernel* SVM

|  |  |  |
| --- | --- | --- |
| kernel | accuracy | Rata-rata K-fold(10 Fold) |
| Linear | 0.73 | 0.70 |
| Polynomial | 0.60 | 0.57 |
| RBF | 0.65 | 0.66 |

Dari table diatas dapat di simpulkan bahwa hasil terbaik di tunjukkan oleh model SVM yang menggunakan kaenel linear, dengan rata-rata akurasi 0.70.

1. Penerapan Algoritma KNN

Hasil Klasifikasi dengan menggunakan algoritma KNN (*n=7*) ditunjukkan oleh *confusion matrix* berikut ini:

Gambar 3.11 *Confusion Matrix* KNN

Dari tabel tersebut, dapat diketahui bahwa nilai akurasi yang didapatkan adalah sebesar 0.60. Adapun setelah divalidasi menggunakan *k-fold* validation (10 *fold*), maka didapatkan rata-rata akurasi sebesar 0.56

Tabel 3.2 Pengujian Ragam nilai *n* pada KNN

|  |  |  |
| --- | --- | --- |
| Jumlah *n* | *accuracy* | Rata-rata *K-fold*(10 *Fold*) |
| 3 | 0.60 | 0.55 |
| 5 | 0.56 | 0.55 |
| 7 | 0.60 | 0.56 |

Dari tabel 3.2 tersebut dapat disimpulkan bahwa hasil terbaik ditunjukkan oleh model KNN yang menggunakan *n* sebanyak 7, dengan rata-rata akurasi sebesar: 0.56.

Dari penggunaan algoritma SVM ataupun KNN didapatkan hasil yang kurang memuaskan yakni kurang dari 75%. Adapun SVM memiliki performa yang lebih baik jika dibandingkan dengan algoritma KNN dalam penelitian ini.

Hasil akurasi yang cukup rendah ini diduga karna proses pelabelan yang otomatis, serta hanya menggunakan satu jenis platform saja yakni: *textblob*.

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut:

1. Opini netizenterkait dengan vaksin *sinovac* yang diambil dari aplikasi *twitter* dengan menggunakan keyword **sinovac**, menghasilkan 1030 opini(data *tweet*) bersentimen positif, 388 opini bersentimen negatif, dan 687 opini bersentimen netral.
2. Algoritma SVM dan KNN dapat digunakan untuk mengklasifikasikan data *tweet* dengan rata-rata hasil akurasi sebesar 0.7 (untuk algoritma SVM) dan 0.56 (untuk algoritma KNN)
3. Dalam penelitian ini algoritma SVM menunjukkan performa yang lebih baik jika di bandingkan dengan algoritma KNN dalam hal akurasi
4. Algoritma SVM memiliki performa yang baik jika dijlankan dengan menggunakan karnel linear.
5. Algoritma KNN akan memiliki performa yang baik Ketika dijalankan menggunakan *n=7*
6. Nilai akurasi dari SVM ataupun KNN terbilang rendah, diduga karena proses pelebelan sentiment yang otomatis menggunakan text blob, untuk itu penelitian selanjutnya dapat di coba menggunakan pelebelan manual atwpun dengan manggunkan platfrom lain seperti *valder* ataupun *spacy*.

# DAFTAR PUSTAKA

[1] S. Lestari and S. Saepudin, “Analisis Sentimen Vaksin Sinovac Pada Twitter Menggunakan Algoritma Naive Bayes,” *SISMATIK (Seminar Nas. Sist. Inf. dan Manaj. Inform.*, pp. 163–170, 2021.

[2] F. Fathonah and A. Herliana, “Penerapan Text Mining Analisis Sentimen Mengenai Vaksin Covid - 19 Menggunakan Metode Naïve Bayes,” *J. Sains dan Inform.*, vol. 7, no. 2, pp. 155–164, 2021, doi: 10.34128/jsi.v7i2.331.

[3] S. Ernawati and R. Wati, “Penerapan Algoritma K-Nearest Neighbors Pada Analisis Sentimen Review Agen Travel,” *J. Khatulistiwa Inform.*, vol. 6, no. 1, pp. 64–69, 2018.

[4] M. R. A. Nasution and M. Hayaty, “Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter,” *J. Inform.*, vol. 6, no. 2, pp. 226–235, 2019, doi: 10.31311/ji.v6i2.5129.

[5] M. N. Muttaqin and I. Kharisudin, “Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine dan K Nearest Neighbor,” *UNNES J. Math.*, vol. 10, no. 2, pp. 22–27, 2021, [Online]. Available: http://journal.unnes.ac.id/sju/index.php/ujm

[6] T. Hendrawati and C. P. Yanti, “Analysis of Twitter Users Sentiment against the Covid-19 Outbreak Using the Backpropagation Method with Adam Optimization,” *J. Electr. Electron. Informatics*, vol. 5, no. 1, p. 1, 2021, doi: 10.24843/jeei.2021.v05.i01.p01.

[7] F. Sodik and I. Kharisudin, “Analisis Sentimen dengan SVM , NAIVE BAYES dan KNN untuk Studi Tanggapan Masyarakat Indonesia Terhadap Pandemi Covid-19 pada Media Sosial Twitter,” *Prisma*, vol. 4, pp. 628–634, 2021.