PERBANDINGAN METODE KLASIFIKASI DENGAN MENERAPKAN *ADABOOST* DALAM ANALISIS SENTIMEN PENGGUNA *TWITTER* X TERHADAP PENERAPAN KURIKULUM MERDEKA

Angga Septiana, Gifthera Dwilestari, Agus Bahtiar
Teknik Informatika, STMIK IKMI Cirebon
Sistem Informasi, STMIK IKMI Cirebon
Jl. Perjuangan No. 10 B Majasem Kec. Kesambi Kota Cirebon
anggaseptiana817@gmail.com

ABSTRAK

Kurikulum Merdeka merupakan kurikulum terbaru pengganti kurikulum 2013. Kurikulum merdeka dalam peluncurannya memiliki berbagai opini pro dan kontra dari masyarakat, saat ini kurikulum merdeka masih hangat dibicarakan dan menjadi kontroversi diberbagai platform digital salah satunya pada media sosial *twitter*. Oleh karena itu, perlu adanya evaluasi secara berkala dan analisis mengenai persepsi masyarakat terhadap kurikulum merdeka. Penelitian ini bertujuan untuk menganalisis sentimen yang muncul di *twitter* dengan menggunakan kata kunci "kurikulum merdeka". Data yang digunakan dalam penelitian ini diperoleh dari hasil *crawling* di *Twitter* yang terdiri dari 4717 data kemudian dimasukkan ke dalam label positif dan negatif. Metode analisis data yang digunakan adalah *Knowledge Discovery in Database* (*KDD*) yang meliputi *Data Selection*, *Preprocessing*, *Transformation*, *Data Mining*, dan *Evaluation*. Metode klasifikasi yang digunakan adalah *Multinominal Naïve Bayes* dan *Support Vector Machine* dengan menerapkan *AdaBoost* untuk meningkatkan kinerja algoritma. Penelitian ini menghasilkan nilai akurasi dan *AUC Support Vector Machine* lebih unggul dibandingkan *Multinomial Naïve Bayes* serta penerapan *AdaBoost* terbukti dapat meningkatkan kinerja algoritma. Metode *Support Vector Machine* sebelum menerapkan *AdaBoost* menghasilkan akurasi sebesar 88,68% dan nilai *AUC* sebesar 0,957, sementara algoritma *Support Vector Machine* setelah menerapkan *AdaBoost* menghasilkan akurasi sebesar 88,85% dan nilai *AUC* 0,933. Algoritma *Multinomial Naïve Bayes* sebelum menerapkan *AdaBoost* menghasilkan nilai akurasi sebesar 79,19% dan nilai *AUC* sebesar 0,777, sementara algoritma *Multinomial Naïve Bayes* setelah menerapkan *AdaBoost* menghasilkan akurasi sebesar 83,57%, dan nilai *AUC* 0,899.

Kata kunci: *Analisis Sentimen, Kurikulum Merdeka, Twitter, Klasifikasi, AdaBoost*

1. PENDAHULUAN

Perkembangan teknologi yang berkembang pesat telah menghadirkan berbagai kemudahan di berbagai bidang salah satunya di bidang pendidikan. Hal ini dibuktikan dengan diluncurkannya kebijakan merdeka belajar oleh Kementrian Pendidikan dan Kebudayaan (Kemendikbud) pada tahun 2020. Kurikulum merdeka merupakan kurikulum yang diluncurkan pada tahun ajaran 2022/2023 pengganti kurikulum 2013. Menurut Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek), saat ini merupakan momentum yang tepat untuk mengevaluasi tingkat kesiapan lembaga pendidikan dalam melaksanakan kurikulum merdeka. Sistem kurikulum merdeka ini diterapkan di sekolah dasar dan sekolah menengah, serta instansi pendidikan setara lainnya yang beroperasi di wilayah Indonesia [1]. Dengan adanya kurikulum baru ini membuat perbincangan hangat saat ini baik di kehidupan nyata ataupun pada media sosial.

Media sosial seringkali digunakan masyarakat untuk memberikan opini terhadap kebijakan yang diterapkan oleh pemerintah. Salah satu media sosial yang sering digunakan masyarakat adalah media sosial *twitter*. *Twitter* merupakan salah satu platform media sosial yang memiliki banyak pengguna aktif di Indonesia, dengan jumlah mencapai 191 juta individu pada bulan Januari 2022. Menurut laporan dari Katadata, jumlah pengguna *Twitter* di Indonesia per bulan April 2023 mencapai 14,8 juta [1]. Respon dari masyarakat pada media sosial twitter terhadap penerapan kurikulum merdeka ini bervariasi, di antaranya berupa ekspresi yang bersifat positif atau mendukung dan negatif atau menentang yang menciptakan suatu kontroversi. Sering munculnya salah satu pernyataan pada twitter adalah kurangnya waktu istirahat yang dimiliki oleh siswa karena banyaknya proyek yang harus dikerjakan secara mandiri atau berkelompok, serta berbagai kontroversi yang lainnya dirasa perlu dilakukan analisis yang mendalam [2]. Oleh karena itu, diperlukan sebuah analisis sentimen untuk memberikan pemahaman tentang opini masyarakat terhadap kurikulum merdeka.

Penelitian terdahulu terkait analisis sentimen dilakukan oleh [2] terkait analisis sentimen penerapan kurikulum merdeka pada pengguna twitter mengungkapkan bahwa terdapat sejumlah opini negatif yang signifikan diungkapkan masyarakat di dalam *tweet*. Penelitian tersebut menggunakan algoritma *K-Nearest Neighbor* yang dikombinasikan dengan metode *forward selection*. Berdasarkan hasil eksperimen yang telah dilakukan, diketahui bahwa metode *K-Nearest Neighbors* (*K-NN*) memiliki tingkat akurasi sebesar 73.64%. Namun, ketika metode *K-NN*

dikombinasikan dengan metode *Forward Selection* (*K-NN + FS*), tingkat akurasi meningkat menjadi 76.82%.

Penelitian selanjutnya dilakukan oleh [1] terkait Analisis sentimen kebijakan penerapan kurikulum merdeka sekolah dasar dan sekolah menengah pada media sosial *twitter* dengan menggunakan metode *word embedding* dan *Long Short-Term Memory Networks* (*LSTM*). Dalam penelitian tersebut terdapat 455 data yang dibagi ke dalam tiga kelas sentimen yaitu sentimen positif berjumlah 81, sentimen negatif berjumlah 273, dan 101 sentimen netral. Untuk model klasifikasi menggunakan *Long-Short-Term Memory* dikombinasikan dengan teknik *word embedding layers*. Didapatkan nilai *precision* 80%, nilai *recall* 81%, *f1-score* 79%, dan *accuracy* 81%.

Penelitian selanjutnya dilakukan oleh [3] yaitu mengenai analisis sentimen terhadap program kampus Merdeka menggunakan komparasi algoritma *Naïve Bayes* dan *Support Vector Machine*. Penelitian tersebut menghasilkan akurasi sebesar 86%, presisi sebesar 87% dan *recall* sebesar 80% dengan data *testing* menggunakan algoritma *Naïve Bayes*. Kemudian menggunakan algoritma *SVM kernel linear* dengan data *tesing* yang sama menghasilkan akurasi sebesar 93%, presisi sebesar 100% dan *recall* sebesar 84%.

Beberapa penelitian telah melakukan analisis sentimen tentang kebijakan kurikulum merdeka. Analisis sentimen perlu dilakukan secara berkala untuk mengetahui pandangan terbaru dari masyarakat terkait kebijakan kurikulum merdeka. Penelitian ini mengusulkan analisis sentimen menggunakan perbandingan algoritma *Multinomial Naïve Bayes* dan *Support Vector Machine* dengan penerapan *AdaBoost* sebagai perbedaan dari penelitian sebelumnya. Data yang digunakan pada penelitian ini merupakan data terbaru yaitu diambil dari *Twitter* dengan cara *crawling* data dari rentang waktu bulan April 2023 hingga bulan September 2023. Hasil penelitian ini dapat digunakan sebagai tolak ukur untuk peningkatan layanan yang ditujukan kepada pengembang kurikulum dan memberikan wawasan kepada masyarakat.

2. TINJAUAN PUSTAKA

2.1. *Text Mining*

Text mining merupakan tahap transformasi yang bertujuan untuk mengekstraksi data terstruktur dari teks yang sebelumnya tidak memiliki struktur tertentu. Setelah berhasil diekstraksi, data tersebut kemudian dianalisis lebih lanjut melalui berbagai pendekatan analisis data guna menghasilkan pola dan fakta yang memiliki signifikansi. Proses analisis teks ini dilakukan dengan memanfaatkan berbagai metode *Natural Language Processing* (*NLP*) [4]. *Text mining* bertujuan menghasilkan informasi dari satu set dokumen. *Text Mining* mampu menghasilkan informasi melalui pemrosesan, pengelompokan, dan analisis data-data tidak terstruktur dalam jumlah besar [5].

2.2. Analisis Sentimen

Analisis sentimen adalah suatu proses komputasi yang bertujuan melakukan klasifikasi terhadap data tekstual berdasarkan sentimen yang terkandung di dalamnya. Popularitas penelitian analisis sentimen semakin meningkat, tidak hanya karena ketersediaan data teks yang melimpah, tetapi juga karena adanya kebutuhan dari berbagai pihak untuk memahami pendapat publik terkait suatu topik tertentu. Proses analisis sentimen dipengaruhi secara signifikan oleh jenis dataset yang digunakan, terutama ketika dataset tersebut terdiri dari rangkaian kalimat yang cukup panjang, yang memerlukan pendekatan penanganan yang khusus [6]. Analisis sentimen dapat disebut juga dengan opinion mining, sebab keduanya menitikberatkan pada evaluasi pendapat yang menyatakan apakah bersifat positif atau negatif [5].

2.3. Klasifikasi Teks

Klasifikasi adalah suatu proses di mana dokumen atau kalimat dikelompokkan ke dalam satu atau lebih kategori yang telah ditetapkan sebelumnya atau ke dalam kelas-kelas yang serupa, umumnya melibatkan pengelompokan ke dalam kategori positif, negatif, dan netral. Klasifikasi teks menjadi semakin penting di era digitalisasi, sejalan dengan peningkatan volume produksi data teks yang pesat. Pertumbuhan yang signifikan ini terutama terjadi pada data teks yang dihasilkan oleh pengguna platform sosial media seperti *Twitter*. Oleh karena itu, analisis klasifikasi diperlukan untuk mengelompokkan data teks ini guna mendapatkan pemahaman yang terkandung dalam informasi tersebut [7].

2.4. *Multinomial Naïve Bayes*

Algoritma *Multinomial Naive Bayes* adalah metode pembelajaran probabilitas yang menggunakan teorema *Bayes* dan sering diterapkan dalam *Natural Language Processing* (*NLP*). Algoritma ini beroperasi berdasarkan konsep *term frequency*, yang mengukur berapa kali suatu kata muncul dalam suatu dokumen. Model ini menggambarkan dua informasi, yaitu keberadaan atau ketiadaan suatu kata dalam dokumen dan seberapa sering kata tersebut muncul dalam dokumen tersebut [8]. Walaupun menggunakan distribusi *multinomial*, algoritma ini dapat diterapkan pada situasi *text mining* dengan mengubah data teks menjadi format *nominal* yang dapat dihitung menggunakan nilai *integer*. Pada klasifikasi *multinomial naive bayes*, kelas dokumen tidak hanya dihitung berdasarkan kata-kata yang muncul, melainkan juga jumlah kemunculan kata-kata tersebut [9].

2.5. *Support Vector Machine*

Support Vector Machine (*SVM*) merupakan suatu metode klasifikasi yang menerapkan konsep pencarian *hyperplane* (bidang pemisah) optimal di dalam ruang

fitur untuk mengidentifikasi pemisahan antara dua kelas. *Hyperplane* yang dihasilkan oleh *SVM* memiliki karakteristik berupa jarak maksimal dari setiap titik data ke *hyperplane* tersebut. *SVM* memiliki kemampuan untuk melakukan klasifikasi pada data yang dapat dipisahkan secara linear maupun non-linear [10].

2.6. *Adaptive Boosting* (*AdaBoost*)

Algoritma *Adaboost* (*Adaptive Boosting*) adalah suatu algoritma *ensemble learning* yang dimanfaatkan untuk meningkatkan kinerja model prediktif dengan mengintegrasikan beberapa model prediksi yang relatif kurang efektif menjadi satu model yang lebih kuat. Tujuan utama dari *Adaboost* adalah mengurangi tingkat bias dan *varians* dari model prediksi yang lemah dengan memberikan bobot yang berbeda pada masing-masing model tersebut [10].

2.7. *Cross Validation*

Cross Validation merupakan suatu teknik dalam *machine learning* yang mengimplikasikan pemisahan dataset menjadi subset-subset, yang kemudian diikuti oleh proses pelatihan dan pengujian model secara berulang pada masing-masing subset. Pendekatan ini dirancang untuk menilai kinerja model, mencegah terjadinya *overfitting*, dan memastikan bahwa model mampu melakukan generalisasi yang optimal pada data yang belum pernah ditemui sebelumnya [11].

2.8. *Evaluation*

Penilaian dalam konteks penelitian ini dilakukan dengan tujuan untuk mengidentifikasi kinerja atau performa model yang diajukan. Metode evaluasi yang diadopsi dalam penelitian ini menggunakan *confusion matrix*. *Confusion matrix* adalah suatu struktur tabel yang memberikan perbandingan antara hasil klasifikasi yang diperoleh dari sistem (prediksi) dengan hasil klasifikasi yang sebenarnya. Tabel pada *confusion matrix* memvisualisasikan jumlah data uji yang terklasifikasi dengan benar dan jumlah data uji yang salah diklasifikasikan [12].

Perhitungan *confusion matrix* adalah sebagai berikut [13]:

$$\text{Akurasi} = \left(\frac{TP+TN}{TP+FP+FN+TN}\right) \times 100$$

$$\text{Presisi} = \left(\frac{TP}{TP+FP}\right) \times 100$$

$$\textit{Recall} = \left(\frac{TP}{TP+FN}\right) \times 100$$

$$\textit{F-Measure} = 2 \times \left(\frac{Presisi \times Recall}{Presisi + Recall}\right) \times 100$$

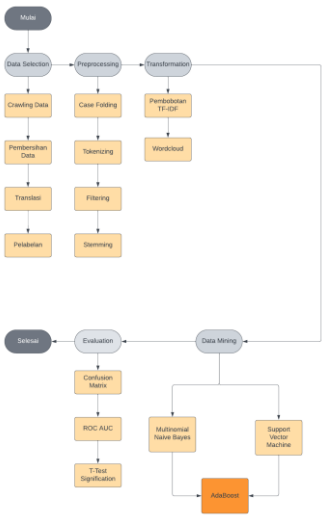
Keterangan:
TP = *True Positive*
FP = *False Positive*
TN = *True Negative*
FN = *False Negative*

2.9. Uji *T-Test*

Uji *T-Test* adalah teknik yang dapat digunakan untuk membandingkan nilai rata-rata antara dua atau lebih kelompok data yang berbeda adalah uji t (*T-Test*). Uji t dapat membantu dalam mengevaluasi perbedaan nilai rata-rata antara dua kelompok data dan menentukan apakah perbedaan tersebut memiliki signifikansi statistik. Dalam konteks analisis sentimen, uji t dapat diterapkan untuk membandingkan nilai sentimen antara dua kelompok [14]. Uji *T-Test* digunakan untuk mengetahui signifikan dari dua atau lebih algoritma yang digunakan sebagai komparasi dalam penelitian.

3. METODE PENELITIAN

Penelitian ini menggunakan metode pendekatan *Knowledge Discovery in Database* (*KDD*). yang melibatkan sejumlah tahapan, mulai dari pemilihan data (*Data Selection*), pra-pemrosesan data (*Preprocessing*), transformasi data (*Transfomation*), penggalian data (*Data Mining*), dan *Evaluation*. Gambar 1. akan digambarkan tahapan dari *KDD*.



Gambar 1. Tahapan *Knowledege Discovery in Database*

Gambar 1 merupakan tahapan yang dilakukan dalam penelitian ini, untuk lebih lengkapnya adalah sebagai berikut:

3.1. *Data Selection*

Tahapan *Data Selection* meliputi *crawling* data yang dilakukan pada media sosial *twitter* dengan kata kunci "kurikulum merdeka" dengan bantuan *library Tweet-Harvest*. Data yang diambil dari rentang waktu bulan April hingga bulan September 2023 dan menghasilkan 4712 data. Data yang berhasil dikumpulkan kemudian dilakukan pembersihan data dan pembuangan data duplikat. Langkah selanjutnya adalah melakukan translasi yaitu proses menyeragamkan data *tweet* berbahasa asing menjadi bahasa Indonesia. Terakhir adalah melakukan

pelabelan yang dilakukan secara manual dengan memahami makna tiap kalimat.

3.2. *Preprocessing*

Preprocessing adalah suatu proses yang dilakukan supaya data *tweet* siap diolah dan relevan dengan penelitian [15]. Tahapan yang dilakukan dalam *preprocessing* diantaranya adalah:

3.2.1. *Case Folding*

Case Folding merupakan proses merubah atau menyeragamkan huruf-huruf di dalam kalimat sehingga menjadi huruf kecil semua (*lowercase*).

3.2.2. *Tokenizing*

Tokenizing merupakan proses memisahkan kata-kata dari kalimat sehingga mempunyai token masing-masing.

3.2.3. *Filtering*

Filtering merupakan proses untuk memfilter data yang tidak memiliki makna dan memiliki jumlah huruf yang terlalu sedikit atau terlalu banyak.

3.2.4. *Stemming*

Stemming merupakan proses menghilangkan imbuhan pada kata sehingga menjadi kata dasar.

3.3. *Transformation*

Tahap *Transformation* adalah proses pembobotan kata dengan *TF-IDF* yang dihasilkan dari proses *preprocessing*. Pembobotan *TF* akan menghitung seberapa sering kata muncul di dalam dokumen, sementara *IDF* menunjukkan relasi antara keberadaan suatu kata pada seluruh dokumen. Setelah pembobotan *TF-IDF*, maka akan di visualisasikan data berdasarkan frekuensi kemunculan paling tinggi atau banyak menggunakan *wordcloud*.

3.4. *Data Mining*

Tahap *data mining* adalah proses mengimplementasikan algoritma *Multinomial Naïve Bayes* dan *Support Vector Machine* sebelum dan setelah menerapkan metode *Adaptive Boosting* untuk peningkatan akurasi.

3.5. *Evaluation*

Pada tahap ini akan didapatkan hasil berupa *confusion matrix* yaitu hasil yang menunjukkan performa dari *machine learning* yaitu akurasi, presisi, *recall*, *F-Measure*, dan nilai *AUC*, di mana didapatkan dari pemodelan menggunakan *Multinomial Naïve Bayes* dan *Support Vector Machine* sebelum dan setelah menerapkan metode *Adaptive Boosting* (*AdaBoost*). Uji *T-Test* digunakan pada penelitian yang berfungsi untuk melihat signifikan perbandingan kedua algoritma setelah menerapkan *AdaBoost*.

4. HASIL DAN PEMBAHASAN

4.1. *Data Selection*

Proses *crawling* data akan menghasilkan data yang masih memiliki banyak karakter seperti *hashtag*, *mention*, *simbol*, dan karakter lainnya serta menghasilkan banyak atribut atau kolom, sehingga perlu dilakukan pembersihan data dan seleksi atribut. Proses pembersihan dan seleksi atribut dilakukan menggunakan *python* dengan *library string* dan *re*. Atribut yang digunakan pada penelitian ini adalah atribut tanggal dan isi konten saja.

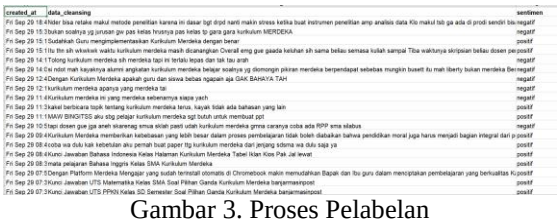
Gambar 2. *Read CSV*

Dari gambar 2, dapat dilihat data yang sudah bersih. Setelah pembersihan dan seleksi atribut maka langkah selanjutnya adalah proses translasi atau menyeragamkan bahasa asing menjadi bahasa Indonesia.

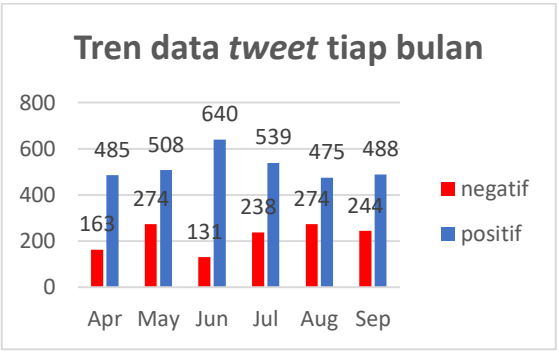
Tabel 1. Proses Translasi

Bahasa Asing atau Daerah	Hasil Translasi
kurikulum merdeka ngrepoti	kurikulum merdeka merepotkan
kurikulum merdeka *got me crying*	kurikulum merdeka membuatku menangis

Setelah semua data tweet berbahasa indonesia, langkah selanjutnya adalah proses pelabelan di mana data tweet akan dimasukkan ke dalam kelas label positif dan negatif.



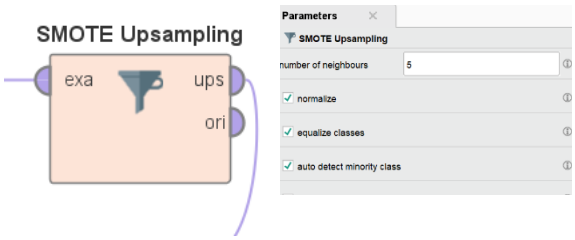
Gambar 3. Proses Pelabelan



Gambar 4. Tren data *tweet* tiap bulan

Hasil pelabelan berdasarkan Gambar 3, didapatkan jumlah sentimen positif sebanyak 3146 data, sementara sentimen negatif sebanyak 1314 data. Untuk mengetahui tren sentimen tiap bulannya, maka dibuat grafik tren data *tweet* tiap bulan pada Gambar 4.

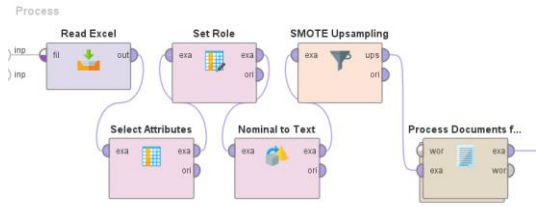
Berdasarkan gambar 4, dapat dilihat bahwa sentimen positif terbanyak ada pada bulan Juni dengan jumlah 640 data, sementara sentimen negatif terbanyak ada pada bulan Mei dan Agustus. Langkah selanjutnya adalah penerapan *SMOTE* untuk menyeimbangkan jumlah kelas dikarenakan jumlah sentimen positif dan negatif tidak seimbang.



Gambar 5. Operator *SMOTE*

4.2. *Preprocessing*

Tahap selanjutnya yaitu preprocessing menggunakan operator *Process Document from Data* pada *RapidMiner*.



Gambar 6. Operator *preprocessing*

Gambar 7. Sub operator Process *Document from Data*

Preprocessing diawali dengan melakukan *Case Folding* menggunakan operator *Transform Case*, akan didapati hasil sebagai berikut:

Tabel 2. Proses *case folding*

Sebelum	Sesudah
Sudahkah Guru mengimplementasikan Kurikulum Merdeka dengan benar	sudahkah guru mengimplementasikan kurikulum merdeka dengan benar

Langkah selanjutnya adalah memecah kata dalam kalimat menggunakan operator *Tokenize*, akan didapati hasil sebagai berikut:

Tabel 3. Proses *tokenizing*

Sebelum	Sesudah
sudahkah guru mengimplementasikan kurikulum merdeka dengan benar	[‘sudahkah’ ‘guru’ ‘mengimplementasikan’ ‘kurikulum’ ‘merdeka’ ‘dengan’ ‘benar’]

Langkah selanjutnya adalah memfilter kata-kata yang memiliki jumlah huruf terlalu sedikit atau terlalu banyak menggunakan operator *Filter Token by Length* dengan memasukkan nilai parameter 3 pada nilai *min char* dan nilai 25 pada *max char*.

Langkah selanjutnya adalah memfilter kata *stopword* menggunakan operator *Filter Stopwords* (*Dictionary*) dengan menggunakan kamus yang berisi kumpulan kata *stopword*, kemudian peneliti menambahkan kata-kata yang ingin difilter pada kamus tersebut, sehingga akan didapati hasil sebagai berikut:

Tabel 4. Proses *Stopwords*

Sebelum	Setelah
aoskwosks aku smp kurikulum merdeka gak ada pts yeyeyeyeyey semangat kak semoga pts nya dilancarin	aku smp kurikulum merdeka pts semangat kak semoga pts dilancarin

Langkah selanjutnya adalah melakukan *stemming* yaitu menghilangkan imbuhan sehingga menjadi kata dasar. Operator yang digunakan adalah *Stem* (*Dictionary*). Kamus yang dimasukkan ke dalam operator dibuat sendiri oleh peneliti dan disimpan dalam format *txt*.

```
abai::diabaikan
abrek::seabrek
absurd::seabsurd
acak::acakacakan
acak::acakan
acak::diacak
acara::acaranya
acu::mengacu
acungi::applause
ada::adakah
ada::adakan
ada::are
ada::aya
ada::diadain
ada::diadakan
```

Gambar 8 Kamus *Stemming*

Hasil dari proses *stemming* adalah sebagai berikut.

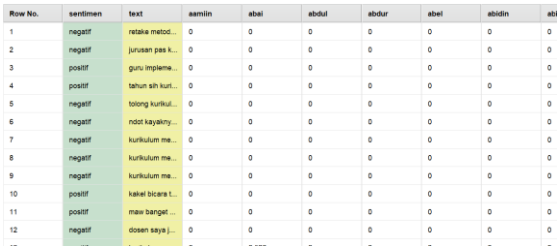
Tabel 5. Proses *stemming*

Sebelum	Sesudah
guru mengimplementasikan kurikulum merdeka dengan benar	guru implementasi kurikulum merdeka dengan benar

4.3. *Transformation*

Pada tahap *transformation* akan dilakukan pembobotan pada setiap kata menggunakan pembobotan *TF-IDF*. Proses pembobotan *TF-IDF* dilakukan pada operator *Process Document from Data*. Operator ini selain melakukan *preprocessing*

juga menghasilkan pembobotan *Term-Frequency* (*TF*) dan *Inverse document Frequency* (*IDF*). *TF* menunjukkan apakah suatu kata ada di dalam data *tweet* atau tidak, jika kata tersebut ada maka akan diberi nilai 1 namun apabila tidak ada maka akan diberi nilai 0. *IDF* menunjukkan relasi antara keberadaan suatu kata pada seluruh dokumen. Berikut merupakan hasil pembobotan *TF-IDF*:



Row No.	sentimen	text	aamiin	abai	abdul	abdur	abel	abidin	ab
1	negatif	retake metod...	0	0	0	0	0	0	0
2	negatif	jurusan pas k...	0	0	0	0	0	0	0
3	positif	guru impleme...	0	0	0	0	0	0	0
4	positif	tahun sih kuri...	0	0	0	0	0	0	0
5	negatif	tolong kurikul...	0	0	0	0	0	0	0
6	negatif	ndot kayakny...	0	0	0	0	0	0	0
7	negatif	kurikulum me...	0	0	0	0	0	0	0
8	negatif	kurikulum me...	0	0	0	0	0	0	0
9	negatif	kurikulum me...	0	0	0	0	0	0	0
10	positif	kakel bicara t...	0	0	0	0	0	0	0
11	positif	maw banget ...	0	0	0	0	0	0	0
12	negatif	dosen saya j...	0	0	0	0	0	0	0

Gambar 9. Hasil *TF-IDF*

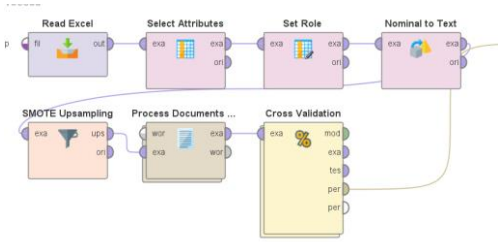
Setelah dilakukan perhitungan *TF-IDF*, langkah selanjutnya adalah menampilkan visualisasi berupa *wordcloud* untuk melihat frekuensi kata-kata yang diutarakan oleh pengguna *Twitter*. Berikut merupakan hasil *wordcloud* pada sentimen positif dan negatif.

Gambar 10. *Wordcloud* sentimen positif

Gambar 11. *Wordcloud* sentimen negatif

4.4. *Data Mining*

Sebelum menerapkan metode *AdaBoost*, dilakukan pengujian masing-masing algoritma dengan 10-*fold cross validation*. Berikut merupakan pemodelan untuk menentukan *n-fold* terbaik:



Gambar 12. Pemodelan algoritma

Keterangan:

Didalam operator *Cross Validation* terdapat dua bagian untuk menyimpan operator diantaranya *training* dan *testing*. Pada operator *training* ditempatkan operator *Naïve Bayes* dan *Support Vector Machine* secara bergantian, sementara pada bagian *testing* ditempatkan operator *apply model* dan *performance*. Berikut merupakan hasil dari pengujian 10-*fold cross validation* dari masing-masing algoritma:

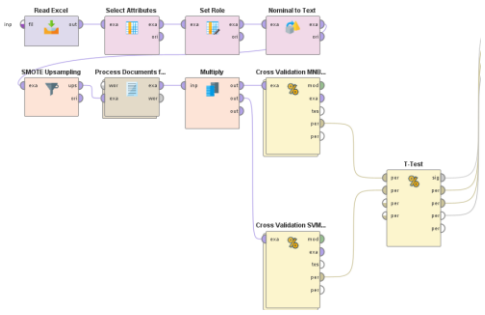
Tabel 6. Hasil 10-*fold cross validation multinomial naïve bayes*

N-Fold	**Akurasi**	**Presisi**	***Recall***
2	77,22%	85,92%	65,14%
3	77,96%	87,87%	64,91%
4	78,41%	88,68%	65,14%
5	78,66%	89,39%	65,04%
6	78,74%	90,12%	64,62%
7	78,66%	89,57%	64,88%
8	78,95%	89,86%	65,26%
9	78,93%	90,40%	64,76%
10	79,19%	90,87%	64,88%
Rata-rata	78,52%	89,19%	64,96%

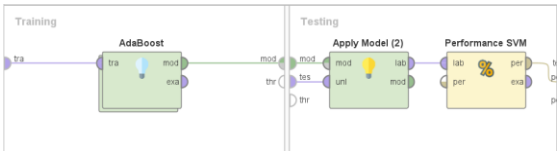
Tabel 7. Hasil 10-*fold cross validation Support vector machine*

N-Fold	**Akurasi**	**Presisi**	***Recall***
2	86,17%	87,43%	84,50%
3	87,19%	88,51%	85,49%
4	87,81%	89,27%	85,96%
5	88,04%	89,65%	86,03%
6	88,48%	90,03%	86,60%
7	88,45%	89,93%	86,63%
8	88,52%	90%	86,67%
9	88,68%	90,12%	86,89%
10	88,60%	90,28%	86,57%
Rata-rata	87,99%	89,47%	86,15%

Setelah dilakukan pengujian 10-*fold cross validation*, maka didapatkan hasil akurasi tertinggi algoritma *multinomial naïve bayes* berada pada *number of fold* 10 sementara *support vector machine* berada pada *number of fold* 9. Hasil akurasi tertinggi pada masing-masing algoritma akan dilakukan pengujian dengan menerapkan *AdaBoost*. Berikut merupakan pemodelan algoritma menggunakan *AdaBoost*.



Gambar 13. Pemodelan algoritma menggunakan *adaboost*



Gambar 14. Sub bagian operator *cross validation*

4.5. *Evaluation*

Tahapan ini akan menunjukkan hasil akurasi dari masing-masing algoritma sebelum dan setelah menerapakan *AdaBoost*.

accuracy: 79.19% +/- 1.81% (micro average: 79.19%)

	true negatif	true positif	class precision
pred. negatif	2931	1101	72.69%
pred. positif	204	2034	90.88%
class recall	93.49%	64.88%	

Gambar 15. *Confusion matrix* algoritma *multinomial naïve bayes* tanpa *adaboost*

accuracy: 83.75% +/- 1.33% (micro average: 83.75%)

	true negatif	true positif	class precision
pred. negatif	2877	761	79.08%
pred. positif	258	2374	90.20%
class recall	91.77%	75.73%	

Gambar 16. *Confusion matrix* algoritma *multinomial naïve bayes* dengan *adaboost*

Berdasarkan gambar 15 dan 16 menunjukkan hasil *confusion matrix* dari algoritma *Multinomial Naïve Bayes* sebelum menerapkan *AdaBoost* menghasilkan nilai akurasi sebesar 79,19%, presisi 90,87%, *recall* 64,88%, dan nilai *F-Measure* 75,71%. Hasil *confusion matrix* dari algoritma *Multinomial Naïve Bayes* setelah menerapkan *AdaBoost* menghasilkan nilai akurasi sebesar 83,75%, presisi 90,29%, *recall* 75,73%, dan nilai *F-Measure* 82,37%.

Nilai *AUC* yang dihasilkan pada algoritma *multinomial naïve bayes* sebelum menggunakan *adaboost* berada pada nilai 0,777, sedangkan dengan penerapan *adaboost* berada pada nilai 0,899 dengan selisih nilai 12,2%. Menurut [16], apabila nilai *AUC* berada pada nilai 0,70-0,80 maka dapat digolongkan klasifikasi sedang, sementara apabila berada pada nilai 0,80-0,90 dapat digolongkan klasifikasi baik.

accuracy: 88.68% +/- 1.23% (micro average: 88.68%)

	true negatif	true positif	class precision
pred. negatif	2836	411	87.34%
pred. positif	299	2724	90.11%
class recall	90.46%	86.89%	

Gambar 17. *Confusion matrix* algoritma *support vector machine* tanpa *adaboost*

accuracy: 88.85% +/- 1.24% (micro average: 88.85%)

	true negatif	true positif	class precision
pred. negatif	2833	397	87.71%
pred. positif	302	2738	90.07%
class recall	90.37%	87.34%	

Gambar 18. *Confusion matrix* algoritma *support vector machine* dengan *adaboost*

Berdasarkan gambar 17 dan 18 menunjukkan hasil *confusion matrix* dari algoritma *Support Vector Machine* sebelum menerapkan *AdaBoost* menghasilkan nilai akurasi sebesar 88,68%, presisi 90,12%, *recall* 86,89%, dan nilai *F-Measure* 88,48%. Hasil *confusion matrix* dari algoritma *Multinomial Naïve Bayes* setelah menerapkan *AdaBoost* menghasilkan nilai akurasi sebesar 88,85%, presisi 90,07%, *recall* 87,34%, dan nilai *F-Measure* 88,68%.

Nilai *AUC* yang dihasilkan pada algoritma *support vector machine* sebelum menggunakan *adaboost* berada pada nilai 0,957, sedangkan dengan penerapan *adaboost* mengalami penurunan sebesar 0,24% dengan nilai 0,933. Menurut [16], apabila nilai *AUC* berada pada nilai 0,90-1,00 dapat digolongkan klasifikasi sangat baik.

Berikut merupakan hasil uji *T-Test* perbandingan algoritma *multinomial naïve bayes* dan *support vector machine* dengan menerapkan *AdaBoost.*

A	B	C
	0.837 +/- 0.013	0.889 +/- 0.012
0.837 +/- 0.013		0.000
0.889 +/- 0.012		

Gambar 19. Hasil uji *T-Test*

Berdasarkan hasil uji *T-Test* didapati perbandingan di bawah *alpha* 0,050 yang artinya perbandingan kedua algoritma signifikan dan algoritma yang paling dominan adalah algoritma *Support Vector Machine* dengan menerapkan *AdaBoost*.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang dilakukan maka didapatkan nilai akurasi 79,19%, presisi 90,87%, *recall* 64,88%, *F-Measure* 75,71%, dan nilai *AUC* berada pada nilai 0,777 untuk algoritma *Multinomial Naïve Bayes* sebelum menerapkan *Adaboost*, sementara setelah menerapkan *AdaBoost* didapatkan nilai akurasi sebesar 83,75%, presisi 90,29%, *recall* 75,73%, *F-Measure* 82,37%, dan nilai *AUC* berada pada nilai 0,899. Algoritma *Support Vector Machine* sebelum menerapkan *AdaBoost* didapatkan nilai akurasi sebesar 88,68%, presisi 90,12%, *recall* 86,89%, *F-Measure* 88,48%, dan nilai *AUC* berada pada nilai 0,957, sementara setelah menerapkan *AdaBoost* didapatkan nilai akurasi sebesar 88,85%, presisi 90,07%, *recall* 87,34%, *F-Measure* 88,68%, dan nilai *AUC* berada pada nilai 0,933. Berdasarkan hasil perbandingan tersebut dapat ditarik kesimpulan bahwa algoritma *Support Vector Machine* dengan menerapkan *AdaBoost* merupakan algoritma terbaik karena memiliki nilai akurasi paling unggul dan masuk ke dalam golongan klasifikasi sangat baik. Saran untuk penelitian selanjutnya adalah dengan melakukan *convert negation* karena di dalam penelitian ini belum diterapkan.

DAFTAR PUSTAKA

[1] A. R. Maulana, S. H. Wijoyo, and Y. T. Mursityo, “Analisis Sentimen Kebijakan Penerapan Kurikulum Merdeka Sekolah Dasar dan Sekolah Menengah pada Media Sosial

Twitter dengan menggunakan Metode Word Embedding dan Long Short-Term Memory Networks (LSTM)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 3, pp. 523–530, 2023.

[2] W. Darmawan, M. F. Kurniawan, and W. Hapsoro, "Analisis Sentimen Penerapan Kurikulum Merdeka Pada Pengguna Twitter Menggunakan Metode K-Nearest Neighbor Dengan Forward Selection," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 12, no. 1, pp. 245–253, 2023.

[3] I. P. Rahayu, A. Fauzi, and J. Indra, "Analisis Sentimen Terhadap Program Kampus Merdeka Menggunakan Naive Bayes Dan Support Vector Machine," *J. Sist. Komput. dan Inform.*, vol. 4, no. 2, p. 296, 2022.

[4] A. Priyanto and M. R. Ma'arif, "Analisis Voice of Customer dari Ulasan Pengguna Produk Smartphone pada Online Marketplace menggunakan Text Mining," *J. Inform. Univ. Pamulang*, vol. 7, no. 1, pp. 2622–4615, 2022.

[5] Samsir, Ambiyar, U. Verawardina, F. Edi, and Watrianthos, "Analisis Sentimen Pembelajaran Daring pada Twitter di Masa Pamndemi COVID-19 menggunakan Metode Naive Bayes," *J. Media Inform. Budidarma*, vol. 5, no. 1, p. 149, 2021.

[6] W. Widayat, "Analisis Sentimen Movie Review menggunakan Word2Vec dan metode LSTM Deep Learning," *J. Media Inform. Budidarma*, vol. 5, no. 3, p. 1018, 2021.

[7] L. Mutawalli, M. T. A. Zaen, and W. Bagye, "Klasifikasi Teks Sosial Media Twitter menggunakan Support Vector Machine (Studi Kasus Penusukan Wiranto)," *J. Inform. dan Rekayasa Elektron.*, vol. 2, no. 2, p. 43, 2019.

[8] Yuyun, N. Hidayah, and S. Sahibu, "Algoritma Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter," vol. 5, no. 10, pp. 820–826, 2021.

[9] I. Verawati and B. S. Audit, "Algoritma Naïve Bayes Classifier Untuk Analisis Sentiment Pengguna Twitter Terhadap Provider By.u," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1411, 2022.

[10] Y. Septiawan and C. Chairani, "Perbandingan Akurasi Metode Deteksi Ujaran Kebencian dalam Postingan Twitter Menggunakan Metode SVM dan Decision Trees yang Dioptimalkan dengan Adaboost," *J. Tek.*, vol. 17, no. 2, pp. 287 – 299–287 – 299, 2023.

[11] J. Rusman, B. Z. Haryati, and A. Michael, "Optimisasi Hiperparamater Tuning pada Metode Support Vector Machine untuk Klasifikasi Tingkat Kematangan Buah Kopi," *J-Icon J. Inform. dan Komput.*, vol. 11, no. 2, 2023.

[12] M. I. Fikri, T. S. Sabrila, and Y. Azhar, "Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter," *Smatika J.*, vol. 10, no. 02, pp. 71–76, 2020.

[13] Rilinka, Indriati, and N. Yudistira, "Analisis Sentimen Penghapusan Ujian Nasional pada Twitter menggunakan Document Frequency Difference dan Multinomial Naïve Bayes," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 3, pp. 876–883, 2021.

[14] Y. A. Singgalen, "Analisis Sentimen Konsumen terhadap Food, Services, and Value di Restoran dan Rumah Makan Populer Kota Makassar Berdasarkan Rekomendasi Tripadvisor Menggunakan Metode CRISP-DM dan SERVQUAL," *Build. Informatics, Technol. Sci.*, vol. 4, no. 4, pp. 1899–1914, 2023.

[15] W. P. Anggraini and M. S. Utami, "Klasifikasi Sentimen Masyarakat Terhadap Kebijakan Kartu Pekerja Di Indonesia," *Fakt. Exacta*, vol. 13, no. 4, p. 255, 2021.

[16] T. Hidayatulloh and L. Yusuf, "Klasifikasi Tipe Berat Tubuh menggunakan Metode Support Vector Machine," *Inti Nusa Mandiri*, vol. 18, no. 1, pp. 71–77, 2023.