

# Analisis Sentimen Opini Publik Terhadap Program Merdeka Belajar Kampus Merdeka Dengan Algoritma *Naive Bayes* – *Support Vector Machine* (NBSVM)

Qurrotul Uyun<sup>1</sup>, Anita Qoiriah<sup>2</sup>

<sup>1,2</sup> Teknik Informatika, Fakultas Teknik, Universitas Negeri Surabaya

<sup>1</sup>[qurrotul.19089@mhs.unesa.ac.id](mailto:qurrotul.19089@mhs.unesa.ac.id)

<sup>2</sup>[anitaqoiriah@unesa.ac.id](mailto:anitaqoiriah@unesa.ac.id)

**Abstrak**— Pendidikan memegang peran penting untuk mendorong pengembangan dan penyebaran teknologi secara inklusif. Untuk mendorong hal itu Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi mengeluarkan kebijakan penyelenggaraan program Merdeka Belajar – Kampus Merdeka (MBKM). Permasalahan serta manfaat yang ingin didapatkan melalui program MBKM menimbulkan pro dan kontra dari akademisi maupun khalayak umum. Pro dan kontra tersebut diutarakan oleh publik melalui media sosial Twitter (X) dan membentuk beragam jenis sentimen. Pada penelitian ini dilakukan klasifikasi terhadap opini positif, negatif, dan netral dengan metode *Naive Bayes* - *Support Vector Machine* dengan data bersumber dari media sosial Twitter. Kombinasi antara metode *Naive Bayes* yang sesuai untuk melakukan klasifikasi pada cuplikan singkat dari dokumen dan *Support Vector Machine* yang cocok untuk teks dalam jumlah relatif banyak diharapkan mampu menunjukkan gambaran umum persepsi publik terhadap pelaksanaan Program Merdeka Belajar - Kampus Merdeka.

Penelitian ini merupakan penelitian eksperimen yang memanfaatkan metode kuantitatif. Dalam penelitian ini, data yang telah melewati tahap prapemrosesan dibagi menjadi data latih dan data uji. Data yang telah melalui proses ekstraksi fitur dirangkai menjadi unigram, bigram, dan trigram yang digunakan untuk melatih Model *Naive Bayes* - *Support Vector Machine*. Pengolahan data pada model NBSVM penelitian ini memanfaatkan pemrosesan serial. Pada tahap terakhir dilakukan uji performansi pada model yang telah dilatih.

Uji performansi pada model menunjukkan metode *Naive Bayes* - *Support Vector Machine* (NBSVM) dengan memanfaatkan K-fold Cross Validation untuk pembagian data menunjukkan akurasi yang paling optimal dengan nilai fold yaitu 25. Adapun hasil akurasi yang didapatkan sebesar 95%, precision 95%, recall dan f1-score sebesar 94%.

**Kata Kunci**— n-gram, analisis sentimen, NBSVM, kampus merdeka, kombinasi metode.

## I. PENDAHULUAN

Untuk menciptakan pendidikan yang berkualitas di era pesatnya perkembangan teknologi dan informasi, Kemendikbudristek mengeluarkan kebijakan penyelenggaraan program Merdeka Belajar – Kampus Merdeka (MBKM) guna membentuk mahasiswa menjadi lulusan dengan kompetensi yang tidak hanya relevan dengan standar industri masa kini, namun juga mampu beradaptasi dengan dinamika sosial, budaya, dan perubahan teknologi di masa mendatang [1]

Permasalahan serta manfaat yang ingin didapatkan melalui program ini menimbulkan pro dan kontra dari akademisi maupun khalayak umum [2]. Pro dan kontra tersebut diutarakan oleh publik melalui media sosial Twitter (X). Sebanyak 277,7 juta pengguna internet di Indonesia secara aktif mengakses platform jejaring sosial Twitter (X). Sejak Januari 2022, pengguna media sosial bertambah sebesar 12,35% dari jumlah pada tahun 2021[3]. Sebagai media sosial berbasis teks, Twitter (X) telah digunakan oleh banyak pemerintahan di dunia sebagai metode untuk berinteraksi dengan masyarakat. Melalui Twitter (X), pemerintah mengekstraksi aspirasi masyarakat sebagai salah satu landasan informasi untuk pengambilan keputusan [4]. Opini, tanggapan, kritik dan saran yang diutarakan publik membentuk sentimen yang bervariasi terhadap pelaksanaan program Merdeka Belajar – Kampus Merdeka [5]

Sentimen diartikan sebagai opini subjektif yang didasarkan pada emosi yang berlebihan terhadap sesuatu. Dalam hal ini, proses analisis terhadap sentimen berarti mengidentifikasi suatu kecenderungan sikap yang diambil oleh seorang individu atau sekelompok masyarakat terhadap suatu konteks yang spesifik. Sentimen tersebut selanjutnya diklasifikasikan ke dalam beberapa kategori: positif, negatif dan netral berdasarkan kecenderungan masing-masing sentimen [6].

*Naive Bayes* didefinisikan sebagai algoritma klasifikasi probabilistik yang memanfaatkan teorema Bayes sebagai landasan untuk melakukan klasifikasi entitas dengan mengasumsikan fitur yang diamati bersifat tidak terikat satu sama lain. Sedangkan, *Support Vector Machine* ialah metode klasifikasi dan regresi dengan prinsip kerja menemukan hyperplane terbaik untuk mengoptimalkan margin dua kelas dari data [7].

Adapun penggunaan metode gabungan *Naive Bayes* - *Support Vector Machine* merujuk pada sebuah penelitian yang memiliki judul *Sentiment Analysis of Positive and Negative of YouTube Comments Using Naive Bayes – Support Vector Machine (NBSVM) Classifier*. Penelitian tersebut menggabungkan algoritma *Naive Bayes* dan *Support Vector Machine* untuk melakukan pengelompokan antara komentar positif dan negatif pada YouTube. *Multinomial Naive Bayes* digunakan untuk menemukan nilai vector fitur yang kemudian diproses dalam persamaan dari *Support Vector Machine*. Penerapan kombinasi algoritma tersebut menghasilkan nilai uji performansi 91% untuk precision, 83% recall dan 87% untuk

f1score. Proses klasifikasi komentar terhadap video YouTube diimplementasikan melalui pembangunan model kombinasi antara metode *Naive Bayes - Support Vector Machine*. *Support Vector Machine* diperlukan untuk memproses nilai feature vector yang telah dihasilkan dari penghitungan menggunakan *Multinomial Naive Bayes* dari data yang telah diproses melalui tahap preprocessing. Selain itu, penggunaan sosial media Twitter (X) untuk mendapatkan data dalam jumlah besar serta bervariasi dinilai mampu meningkatkan akurasi nilai guna menemukan persamaan hyperplane terbaik [8].

Penelitian terdahulu menganalisis sentimen dari topik yang sama, dengan mengimplementasikan metode *Naive Bayes* dengan basis Particle Swarm Optimization mendapatkan temuan sentimen negatif sejumlah 38.08% dan 61.92% untuk sentimen positif dari total 428 data. Dalam hal ini, metode *Naive Bayes* dengan basis Particle Swarm Optimization mendapatkan akurasi 20% lebih kecil dibandingkan metode *Naive Bayes* tanpa Particle Swarm Optimization [9]. Kemudian terdapat penelitian sebelumnya mengenai analisis sentimen dengan data yang bersumber dari Twitter (X). Penelitian tersebut bertujuan untuk melakukan klasifikasi pada Tweet komentar masyarakat terhadap BMKG Nasional. Metode serupa yakni *Naive Bayes Classifier* dimanfaatkan dalam penelitian ini dan menghasilkan akurasi sebesar 68,97%. Hal ini dikarenakan tidak dilakukannya tahapan normalisasi Fitur pada tahap preprocessing [9].

Berdasarkan latar belakang yang telah diuraikan, penelitian ini ditujukan untuk melakukan klasifikasi terhadap opini positif dan negatif terkait program Merdeka Belajar - Kampus Merdeka dengan metode *Naive Bayes - Support Vector Machine* berdasarkan sumber data media sosial Twitter. Adapun perbedaan penelitian ini dengan penelitian berjudul, "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naive Bayes - Support Vector Machine (NBSVM) Classifier," sebagai jurnal rujukan utama penelitian ini yaitu: 1) Penggunaan media sosial Twitter sebagai sumber pengambilan data; 2) Penggunaan n-gram berupa bigram pada data referensi kata positif dan negatif; 3) Penggunaan pendapat publik berbentuk tweet mengenai pelaksanaan program Merdeka Belajar - Kampus Merdeka sebagai sentimen yang akan dianalisis; 4) Penggunaan jumlah data yang lebih banyak yakni 700 data, dan; 5) Penggunaan nilai tambahan accuracy sebagai ukuran performa selain precision, recall, dan f1-score. Kombinasi antara metode *Naive Bayes* yang cocok untuk mengelompokkan cuplikan singkat dari dokumen dan *Support Vector Machine* yang sesuai dengan teks dalam jumlah relative banyak diharapkan mampu menunjukkan gambaran umum persepsi publik terhadap pelaksanaan Program Merdeka Belajar - Kampus Merdeka.

## II. LANDASAN TEORI

### A. Natural Language Processing

*Natural Language Processing* didefinisikan sebagai salah satu cabang ilmu *Artificial Intelligent* dengan fokus pada kemampuan komputer untuk melakukan berbagai pekerjaan yang melibatkan bahasa alami manusia. Algoritma yang dirancang berdasarkan *Natural Language Processing*

menggunakan data bahasa alami (natural language) yang tidak terstruktur sebagai input maupun output. *Natural Language Processing* memungkinkan komputer melakukan tugas sederhana pemrosesan teks atau bahasa sehingga mampu menghasilkan keluaran yang lebih bermakna, meningkatkan komunikasi antar manusia, hingga memungkinkan terjadinya komunikasi antara mesin dan manusia.

### B. Text Mining

*Text Mining* adalah teknologi untuk mengekstraksi pola berisi informasi dan pengetahuan yang memiliki guna dari teks dalam jumlah besar atau dokumen yang tidak terstruktur [10]. *Text mining* menerima input berupa dokumen atau korpus teks. Korpus teks yang tidak terstruktur dapat berbentuk dokumen, email, pesan, atau *tweet*. Dalam proses implementasi *text mining* dengan menggunakan NLP diperlukan adanya tahapan *text preprocessing*. Tahapan *text preprocessing* penting untuk mempercepat proses analisis dan meningkatkan akurasi hasil analisis.

### C. Data Resampling

Salah satu pendekatan yang digunakan untuk menangani masalah ketidakseimbangan data adalah *Resampling*. Ketidakseimbangan data menyebabkan model klasifikasi memiliki kecenderungan untuk memperhatikan data kelas mayoritas dan mengabaikan kelas minoritas. Salah satu teknik *resampling* yang umum digunakan adalah *oversampling* (Penambahan Sampel). *Oversampling* adalah konsep yang mengacu pada penambahan duplikat sampel dari kelas minoritas untuk menyeimbangkan distribusi kelas. Contoh *Oversampling* adalah SMOTE (*Synthetic Minority Oversampling Technique*) *Oversampling*.

### D. Program Merdeka Belajar Kampus Merdeka (MBKM)

Program Merdeka Belajar Kampus Merdeka adalah salah satu kebijakan yang merupakan wujud implementasi dari berbagai landasan peraturan dan hukum pendidikan tinggi yang diusulkan oleh Menteri Pendidikan dan Kebudayaan, Nadiem Makariem. Program MBKM dilaksanakan guna menyediakan hak belajar dalam kurun waktu tiga semester di luar program studi kepada mahasiswa. Adapun tujuan diselenggarakannya program ini adalah untuk mempersiapkan lulusan yang unggul dan berkarakter serta memiliki kompetensi sesuai standar kebutuhan industri dari segi softskills maupun hardskills. Bentuk kegiatan yang difasilitasi oleh program Merdeka Belajar Kampus Merdeka meliputi: Pertukaran pelajar; Magang; Asistensi mengajar di satuan Pendidikan; Penelitian; Proyek kemanusiaan; Kegiatan wirausaha; Studi/proyek independent; Serta membangun desa/Kuliah Kerja Nyata Tematik.

### E. Algoritma Klasifikasi

*Naive Bayes* dan *Support Vector Machine* memiliki keunggulan masing-masing dalam melakukan klasifikasi sentimen. *Naive Bayes Classifier* mampu melakukan komputasi yang cepat dan efisien bahkan dengan jumlah dataset yang besar. Algoritma *Naive Bayes* juga memiliki

kinerja yang baik dalam mengatasi masalah overfitting pada data latih yang besar dan cocok digunakan pada dataset yang memiliki fitur yang banyak. Sedangkan, *Support Vector Machine* mampu menghasilkan hipotesis klasifikasi yang kompleks dan mengatasi permasalahan terkait data yang tidak seimbang [11]. Oleh karena itu, kombinasi antara metode *Naive Bayes* dan *Support Vector Machine* mampu meningkatkan akurasi pada hasil klasifikasi dibandingkan dengan menggunakan salah satu metode saja. Selain itu, permasalahan lain seperti kelas tidak seimbang dan waktu komputasi yang lambat dapat diatasi dengan menggabungkan kedua metode tersebut [12]. Berikut adalah formula kombinasi antara *Naive Bayes* dan *Support Vector Machine* yang menjadi model utama untuk melakukan klasifikasi terhadap kasus uji  $k$ :

$$y^{(k)} = \text{sign}(w^T x^{(k)} + b) \quad (1)$$

#### F. *Multinomial Naive Bayes*

*Multinomial Naive Bayes* menganggap events, yaitu kemunculan kata-kata dalam dokumen yang dijadikan acuan dalam analisis sentimen, terjadi secara acak dengan probabilitas diberikan  $P = (p_1, \dots, p_n)$ .

Distribusi *Multinomial P* menggambarkan probabilitas kemunculan setiap kata dalam seluruh dokumen pelatihan. Fitur vektor  $x$  adalah histogram yang merepresentasikan jumlah kemunculan setiap event dalam kalimat yang dianalisis. Fitur vector  $x = (x_1, \dots, x_2)$  dengan  $x_k$  adalah jumlah kemunculan event  $k$  dalam setiap *tweet* yang diamati. Perbandingan probabilitas kemunculan fitur vector  $x$  dalam kelas positif dan negatif yang dihitung dengan mengasumsikan kelas  $y$  (positif atau negatif) dipengaruhi oleh:

$$P(y|x) = \log p(y) + \sum_k x_k \log p_{yk} \quad (2)$$

$$y = \text{argmax} \log p(y) + \sum_k x_k \log p_{yk} \quad (3)$$

$$y = \text{sign}(\log \frac{N_+}{N_-} + r^T f) \quad (4)$$

#### G. *Support Vector Machine (SVM)*

*Support Vector Machine* didefinisikan sebagai algoritma pembelajaran mesin yang berfokus mencari *hyperplane* terbaik yang dapat memisahkan dua kelas data dengan margin maksimal. Pada penelitian ini digunakan SVM dengan kernel linear. Vektor fitur  $x^k = f^k$  dan parameter  $w$  dan  $b$  diperoleh melalui proses optimasi yang dilakukan dengan meminimalkan fungsi loss (loss function). Parameter  $w$  dan  $b$  yang optimal dapat digunakan untuk memisahkan kelas positif dan negatif. Untuk meminimalkan nilai loss function pada SVM digunakan persamaan berikut:

$$L(w, b) = w^T w + c \sum \max(0, 1 - y^{(i)}(w^T F^{(i)} + b)) \quad (5)$$

#### H. *Support Vector Machine* dengan fitur *Naive Bayes* (NBSVM)

Pengolahan Penggunaan *log likelihood ratio* dalam kombinasi dengan metode klasifikasi seperti *Multinomial Naive Bayes* dan *Support Vector Machine* dapat menghasilkan performa klasifikasi yang baik. *Log likelihood ratio* dikalikan dengan bobot  $r$  untuk membentuk vektor fitur yang diwakili oleh  $x = r \cdot f$ . Berikut adalah model yang digunakan dalam metode *Naive Bayes - Support Vector Machine*:

$$w^T = (1 - \beta)\omega + \beta\omega \quad (6)$$

Keterangan:

$$\beta = [0,1] \quad \omega = \frac{\|w\|}{|v|}$$

Dimana  $\omega$  merepresentasikan bobot yang telah didapatkan dari penghitungan model *Support Vector Machine* dan  $w$  adalah bobot yang dihasilkan oleh dari model *Naive Bayes*. Kemudian  $\beta$  digunakan untuk menghitung seberapa besar penggunaan masing-masing bobot dari *Naive Bayes* dan *Support Vector Machine*. Penggabungan bobot dari *Naive Bayes* dan *Support Vector Machine* memanfaatkan pendekatan gabungan (blended) dengan pengolahan serial yang ditunjukkan pada saat algoritma *Naive Bayes* melakukan ekstraksi fitur pada teks dan menghasilkan posterior probability kemudian hasil dari algoritma *Naive Bayes* digunakan sebagai fitur untuk melatih *Support Vector Machine*. Sementara pendekatan paralel diimplementasikan saat proses penggabungan bobot antara *Naive Bayes* dan *Support Vector Machine*.

#### I. Twitter (X)

Twitter (X) adalah platform jejaring sosial yang memungkinkan pengguna untuk saling melakukan pengiriman pesan atau posting yang dikenal sebagai "tweet" atau "cuitan" ke pengguna lain. Twitter (X) populer sebagai sumber data teks mengenai opini publik sehingga sering digunakan sebagai sumber berita, serta untuk mengikuti tren dan opini publik tentang topik tertentu. Twitter (X) dibuat pada tahun 2006 dan saat ini merupakan salah satu platform jejaring sosial paling populer di dunia [13].

#### J. Uji Performansi

Untuk mengetahui seberapa baik model klasifikasi dalam mengidentifikasi dan mengklasifikasikan sentimen dalam teks diperlukan evaluasi melalui uji performansi. Uji performansi pada umumnya menggunakan confusion metric untuk mengukur performansi model. Melalui confusion metric, dapat melakukan penghitungan metrik evaluasi diantaranya; Akurasi (accuracy); Precision; Recall, dan; F1-score [14]. Confusion metric melakukan evaluasi pada model dengan menunjukkan jumlah prediksi yang salah dan benar dari hasil klasifikasi yang telah dilakukan oleh model, dengan mengamati perbandingan nilai aktual dari data dengan nilai yang diprediksi oleh model.

Tabel 1 Confusion Matrics

	Predicted Positive	Predictied Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Akurasi adalah rasio jumlah ketepatan prediksi suatu kelas terhadap jumlah total dokumen yang diklasifikasi oleh model. Persamaan (7) ialah rumus umum akurasi (*accuracy*) yang digunakan untuk klasifikasi biner:

$$accuracy(a) = \frac{TP+TN}{Total\ Data} \quad (7)$$

*Precision* berguna untuk mengukur seberapa akurat model dalam melakukan prediksi yang benar terhadap kelas positif dari seluruh prediksi positif yang dilakukan. Rumus *precision* diuraikan dalam persamaan (8):

$$precisiom(p) = \frac{TP}{TP+FP} \quad (8)$$

*Recall* merupakan perbandingan jumlah kelas positif yang diprediksi dengan benar sebagai terhadap jumlah dokumen yang sebenarnya positif, baik dokumen yang diklasifikasikan secara benar sebagai positif maupun dokumen yang salah diklasifikasikan sebagai negatif.

$$recall = \frac{TP}{TP+FN} \quad (9)$$

Sementara itu, *f1-score* bertujuan menghasilkan deskripsi kinerja yang lebih unggul dari suatu model sebab menggabungkan *precision* dan *recall*.

### III. METODOLOGI PENELITIAN



Gambar 1 Diagram Alur Penelitian

#### A. Identifikasi Masalah

Penelitian ini merupakan penelitian eksperimen yang memanfaatkan metode kuantitatif. Penelitian eksperimen melakukan pengamatan yang sistematis dan terkontrol terhadap variabel yang dimanipulasi (variabel independen) untuk melakukan uji sebab-akibat antara variabel bebas dan variabel terikat. Sementara itu, metode penelitian kuantitatif merujuk pada bagaimana menghasilkan generalisasi yang dapat diterapkan pada populasi yang lebih luas. Metode ini didasarkan pada pengumpulan, analisis, dan interpretasi data kuantitatif menggunakan teknik statistik [12]. Gambar 1 menunjukkan tahapan penelitian sebagai acuan langkah-langkah dalam implementasi penelitian ini, yang selanjutnya akan dijelaskan ke dalam masing masing poin. Penelitian diawali dengan identifikasi permasalahan yang akan diteliti. Proses identifikasi masalah dilakukan dengan menganalisis tweet atau cuitan di media sosial Twitter yang berisi opini publik terkait implementasi program Merdeka Belajar Kampus Merdeka.. Penelitian ini membahas mengenai analisis sentimen untuk mengklasifikasikan opini masyarakat terkait pelaksanaan Program Merdeka Belajar Kampus Merdeka dengan memanfaatkan kombinasi algoritma *Naive Bayes* - *Support Vector Machine*. Dalam penelitian ini dilakukan perancangan dan pengimplementasian algoritma *Naive Bayes* - *Support Vector Machine* dengan tujuan mengklasifikasikan opini publik terkait pelaksanaan Program Merdeka Belajar Kampus Merdeka ke dalam sentimen positif dan negatif.

#### B. Studi Literatur

Guna menyusun dasar teori yang mendukung penelitian dan melandasi pengujian, studi literatur dilaksanakan dengan mencari referensi berupa buku maupun penelitian terdahulu dalam bentuk skripsi, artikel maupun jurnal internasional dan jurnal nasional. Seluruh literatur yang digunakan merupakan literatur yang berhubungan dengan analisis sentimen opini masyarakat Indonesia di platform media sosial, algoritma klasifikasi *Naive Bayes* dan *Support Vector Machine*, penggabungan algoritma *Naive Bayes* dan *Support Vector Machine*, dan pembahasan lain yang memiliki relevansi dengan materi penelitian.

#### C. Analisis Kebutuhan

Setelah studi literatur dilakukan, dilakukan perencanaan kebutuhan yang diperlukan untuk melaksanakan penelitian agar tujuan dari penelitian dapat tercapai dengan mempertimbangkan batasan batasan yang sudah disepakati. Analisis kebutuhan dalam penelitian ini merupakan proses identifikasi dan pemahaman terhadap kebutuhan-kebutuhan yang perlu dipenuhi untuk mencapai tujuan penelitian yakni analisis sentimen opini publik terhadap program Merdeka Belajar Kampus Merdeka. Dalam pengembangan algoritma pembelajaran, dibutuhkan perangkat lunak dan perangkat keras dengan kriteria tertentu untuk mendukung rangkaian pelaksanaan penelitian sehingga tujuan yang diinginkan dapat tercapai. Berikut rincian kebutuhan perangkat:

1. Kebutuhan Perangkat Keras (Hardware)

Perangkat keras dalam penelitian yaitu perangkat laptop sebagai uji coba dengan spesifikasi berikut :

Prosesor : Intel Celeron CPU N3350 @ 1.10GHz (2 CPUs)  
RAM : 6.00 GB  
Penyimpanan : 500 GB HDD  
Sistem Operasi : Windows 10 Home Single Language 64 bit

2. Kebutuhan Perangkat Lunak (Software)

Guna melakukan pengoperasian sistem pada penelitian, diperlukan browser Google Chrome dengan versi 111.0.5563.65 (Official Build) (64-bit). Browser Google Chrome dimanfaatkan sebagai piranti peramban untuk untuk mengakses layanan cloud computing Google Colab yang memungkinkan pengguna untuk menjalankan dan mengembangkan kode dengan bahasa pemrograman Python

D. Jenis dan Sumber Data

Data latih diambil dari tweet berisi opini mengenai pelaksanaan program Merdeka Belajar Kampus Merdeka berjumlah 400 data tweet yang telah diklasifikasikan secara manual ke dalam dua kelas sentimen yakni positif dan negatif. Data uji didapatkan dari hasil *crawling* menggunakan *scraping tool* Tweet-Harvest. Sementara itu, data-data lain yang digunakan sebagai referensi *stopword* dan *rootword* memanfaatkan data penelitian terdahulu yang telah melalui proses validasi ahli bahasa. Untuk tahap *stemming* pada tahap pra-pemrosesan data, digunakan *library sastrawi* untuk melakukan pengembalian kata dasar sesuai dengan Kamus Besar Bahasa Indonesia. Tahap *stopword removal* memanfaatkan data dari penelitian berjudul *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia* oleh Fadhillah Z Tala dengan 758 data [15]. Selanjutnya, referensi kata yang bersifat positif dan negatif didapatkan melalui dari penelitian berjudul *InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs* oleh Fajri Koto dan Rahmaningtyas yang berisi 3,609 kata positif dan 6,609 kata negatif serta penelitian berjudul *Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity* dengan total 1,881 kata [16], [17].

E. Variabel dan Skala Pengukuran

Guna mengelompokkan data ke dalam kelas positif, negatif, dan netral, dilakukan penentuan variabel. Labelling pada data latih dilakukan dengan mengimplementasikan analisis sentimen berbasis leksikon yang telah diterapkan pada penelitian terdahulu [17]. Penentuan skala pengukuran positif, negatif dan netral dilakukan berdasarkan skor kata. Contoh pelabelan data latih digambarkan pada Tabel 2. Setelah pelabelan dilakukan, dilakukan pemeriksaan distribusi jumlah data dalam masing-masing kelas. Jika ditemukan ketidakseimbangan data dalam kelas maka akan diatasi dengan memanfaatkan Teknik *Oversampling*.

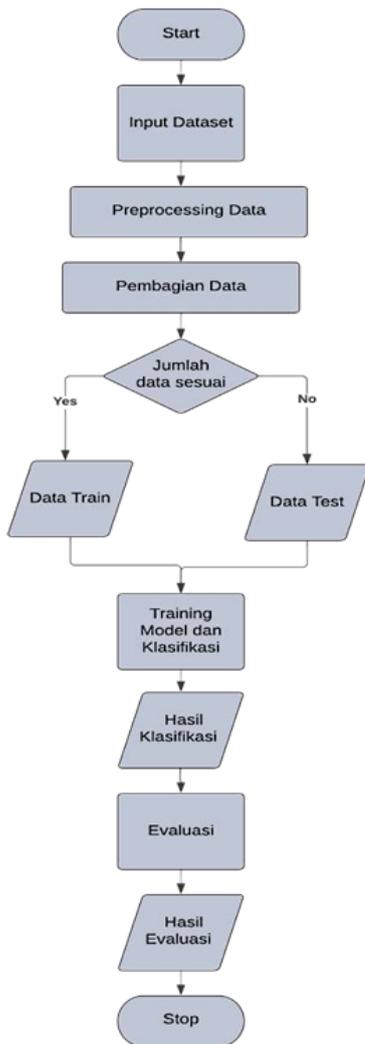
Tabel 2 Contoh Operasional Variabel

Variabel	Operasional Variabel	Nilai Variabel
Positif	Saya senang gebrakana pak Nadiem lewat kampus Merdeka	1
Positif	Gara gara ada kampus Merdeka aku jadi semangat cari uang	1
Netral	Cek aja program yang disediakan di ig kampus Merdeka	0
Negatif	Kampus Merdeka susah banget daftar	-1
Negatif	Sistem kampus merdeka nggak ramah buat semua kampus kecewa	-1

F. Perancangan Model

Model pada penelitian ini akan dirancang dan digunakan dengan memanfaatkan algoritma *Naive Bayes - Support Vector Machine* dalam melakukan analisis sentimen opini publik mengenai pelaksanaan program MBKM. Berikut adalah alur proses model yang digambarkan melalui diagram alir pada Gambar 2. Penjelasan tiap proses yang digambarkan oleh diagram alir pada Gambar 2 dijelaskan melalui Tabel 3. Populasi penelitian ini adalah tweet berisi opini masyarakat Indonesia terkait pelaksanaan program Merdeka Belajar Kampus Merdeka. Teknik *crawling* digunakan untuk mendapatkan data yang akan dijadikan bahan penelitian. Dalam peneletian ini *crawling* dilakukan dengan memanfaatkan *scraping tool* bernama Tweet-Harvest dengan tools Google Colab. Gambaran proses *Crawling* dijelaskan pada Gambar 3. Berdasarkan Gambar 3, program dimulai dengan melakukan konfigurasi sesuai dengan parameter yang diinginkan, sebelumnya pastikan modul *Pandas* telah diimport. Parameter yang digunakan adalah: 1) Kata kunci yang ingin ditemukan, yakni "kampus merdeka"; 2) Batas jumlah data tweet yang ingin di-crawl, yakni berjumlah "700" data tweet; 3) Bahasa tweet yang ingin di-crawl, yakni Bahasa Indonesia atau "id"; 4) Rentang tanggal tweet, yakni "2021-0615" hingga "2023-02-15".

Tahap *stemming* dimulai dengan menjadikan data tweet hasil *crawling* sebagai input. Sesuai dengan penjelasan pada Tabel 3, pada langkah pertama dilakukan proses pembersihan data *tweet* dari karakter khusus. Tanda baca dalam teks akan dibersihkan dengan fungsi *cleaning*. *Library* yang digunakan pada kode untuk menghapus tanda baca merupakan *library string*. *Library string* memungkinkan pengaksesan karakter atau tanda baca yang akan dihilangkan. Fungsi *stemming* dalam diagram pada Gambar 4 mengambil referensi dari fungsi yang telah ada dalam modul *Sastrawi*. Inialisasi objek *stemmer* dilakukan dengan 'StemmerFactory' yang diambil dari modul *Sastrawi*.



Gambar 2 Diagram Alir Model

Tabel 3 Deskripsi Alur Model

No	Tahap	Input	Proses	Output
1	Input Dataset		Pada tahap ini dilakukan input dataset yang telah didapatkan dari proses crawling menggunakan scraping tool Tweet-Harvest Google Colab.	Dataset
2	Preprocessing Data	Dataset	Pada tahap ini diambil	

			400 data tweet	
			<i>Data Cleaning dan Casefolding:</i> Pada tahap ini dilakukan pembersihan data dari karakter khusus, tanda baca atau angka dan pengkonversian karakter menjadi huruf kecil atau <i>lowercase</i>	Dataset tweet yang telah dibersihkan dari tanda baca atau angka serta seru
		Dataset tweet yang telah dibersihkan dari karakter khusus, tanda baca atau angka serta tanda seru	<i>Tokenization :</i> Tahapan untuk memisahkan kata yang masih berupa kalimat menjadi kata yang dipisahkan oleh kalimat	Dataset tweet yang telah dipotong menjadi kata yang dipisahkan oleh spasi
		Dataset tweet yang telah dipotong menjadi kata per kata yang dipisahkan oleh spasi	<i>Stopword Removal:</i> Proses menghilangkan kata-kata yang dianggap tidak signifikan dalam teks. Pada umumnya, kata-kata tersebut merupakan kata-kata penghubung, kata depan, dan kata ganti seperti "dan", "di",	Dataset tweet yang telah dibersihkan dari <i>stopword</i> sehingga hanya terdapat kata-kata yang memiliki sentimen

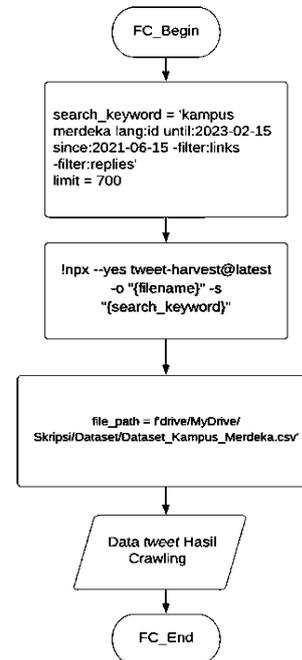
			"yang", "pada" dan sebagainya	
3	Pembagian Data	Dataset <i>tweet</i> yang telah dibersihkan dari <i>stopword</i> sehingga hanya terdapat kata kata yang memiliki sentimen	Membagi dataset menjadi data uji dan data latih menggunakan <i>K-Fold Cross Validation</i>	Data yang telah dibagi menjadi data uji dan data latih
4	Training Model	Data Training	Melatih model klasifikasi dengan menggunakan algoritma pembelajaran mesin yakni <i>Naive Bayes</i> dan <i>Support Vector Machine</i> . Model yang telah dilatih akan digunakan untuk mengklasifikasi sentimen pada data testing	Data hasil klasifikasi dan model yang telah dilatih
			Tahap pertama yang dilakukan adalah ekstraksi fitur dengan <i>Count Vectorizer</i> dan <i>TfidfTransformer</i> . Pada tahap ini dilakukan pembentukan bigram	<i>Feature Vector</i> berupa matriks TFIDF yang telah dihitung bobot nilai teksnya

			dengan memanfaatkan TFIDF menghasilkan representasi numerik dari teks	
		Matriks TF-IDF yang telah dihitung bobot nilai teksnya	Kombinasi <i>Multinomial Naive Bayes</i> dan rumus <i>Feature Vector</i> berfungsi untuk melakukan penghitungan <i>posterior probability</i> dengan mengklasifikasi sentimen dari dokumen dengan memanfaatkan kemunculan kata-kata tertentu dalam dokumen	Fitur probabilitas yang merupakan hasil prediksi <i>posterior probability</i> kelas terhadap data latih
		Fitur probabilitas yang merupakan hasil prediksi <i>posterior probability</i> kelas terhadap data uji	<i>Support Vector Machine</i> dilatih mengenali pola data latih dengan <i>posterior probability</i> sebagai fitur tambahan	Nilai <i>w</i> dan <i>b</i> untuk menentukan <i>hyperplane</i> yang optimal untuk memisahkan kelas positif dan negatif
		Vektor bobot <i>Naive Bayes</i> dan <i>Support Vector Machine</i>	Pada tahap akhir dilakukan penggabungan antara vektor bobot <i>Naive Bayes</i> dan <i>Support</i>	Vektor bobot gabungan antara <i>Naive Bayes</i> dan <i>Support</i>

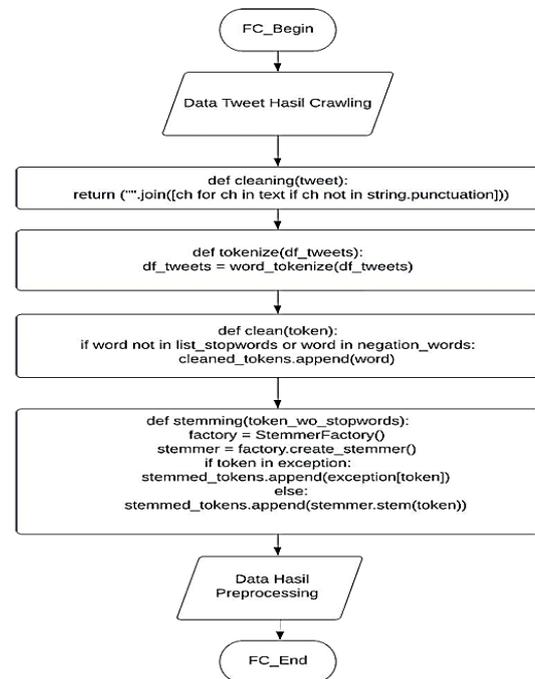
			Vector Machine dengan rumus Hybrid Method NBSVM untuk menghasilkan vektor bobot baru yang lebih optimal untuk melakukan klasifikasi	Vector Machine untuk melakukan klasifikasi
5	Testing	Data testing dan model yang telah dilatih	Pada tahap ini dilakukan pengujian model menggunakan data testing yang telah dipisahkan sebelumnya untuk diklasifikasikan	Data label hasil klasifikasi
6	Evaluasi Model	Data label sebenarnya dan label prediksi	Proses uji performansi dengan metrik uji performansi yakni confusion metrics, akurasi, presisi, recall, dan f1-score	Hasil Evaluasi
7	Hasil Evaluasi		Hasil dari pengujian dan evaluasi model dianalisis untuk mengetahui keakuratan dan performa model dalam melakukan analisis sentimen pada tweet.	Hasil uji performansi

Pada Gambar 5 dijelaskan bagaimana alur model klasifikasi dirancang pada penelitian terdahulu. Penelitian

terdahulu bertujuan untuk mengelompokkan data ke dalam kelas sentimen positif dan negatif dengan memanfaatkan kombinasi antara metode klasifikasi *Naive Bayes* dan *Support Vector Machine* serta Log-count ratio sebagai nilai fitur [18]. Berdasarkan *flowchart* pada Gambar 5, dataset dibagi menggunakan *K-fold Cross Validation*.



Gambar 3 Diagram alir crawling



Gambar 4 Diagram Alir Text Preprocessing

Pada tahap selanjutnya, data latih yang telah diekstraksi menjadi vektor fitur oleh *CountVectorizer* dan *TfidfTransformer* diproses dengan persamaan (2) sehingga ditemukan nilai  $r$  yakni *log-count ratio* yang membandingkan jumlah kemunculan kata yang mungkin memiliki sentimen positif atau negatif. Adapun pembeda penelitian ini terhadap penelitian terdahulu ditunjukkan melalui pemanfaatan fungsi *Count Vectorizer* dan *TF-IDF Transformer* secara bersamaan. *Count Vectorizer* digunakan untuk melakukan konversi teks menjadi matriks hitungan token atau fitur berdasarkan n-gram atau frekuensi kata. Dalam penelitian ini parameter *ngram\_range* diinisialisasi dengan nilai (1,3) dan menghasilkan *unigram* (kata tunggal), *bigram* (kombinasi 2 kata), *trigram* (kombinasi 3 kata) dari teks. Sedangkan *TfidfTransformer* mengubah matriks yang dihasilkan *Count Vectorizer* menjadi matriks TFIDF yang memberikan bobot lebih pada kata kata yang lebih signifikan. Perbedaan lainnya ditunjukkan dengan pemanfaatan *K-Fold Cross Validation* sebagai metode untuk membagi data latih dan data uji. Teknik ini bekerja dengan membagi dataset menjadi K bagian atau 'fold'. Dalam hal ini nilai diinisialisasi secara bergantian dengan nilai 5, 10, 15, 20 dan 25.. Perbedaan tersebut ditampilkan melalui Gambar 6. Setelah dilakukan proses klasifikasi pada data hasil prapemrosesan teks, selanjutnya dilakukan uji performansi pada model. Pada tahap ini dilakukan pengujian terhadap model yang telah dilatih dengan cara mengubah rasio perbandingan data. Selanjutnya, model akan diuji dengan *confusion matrix*. Hasil akurasi dari masing-masing model akan dibandingkan untuk menemukan model yang paling akurat. Tabel *Confusion matrix* mengenai perbandingan nilai aktual dan prediksi tersebut ditunjukkan pada Tabel 4.

Sedangkan berikut adalah rumus untuk menghitung nilai akurasi untuk mengetahui jumlah prediksi yang benar dari seluruh prediksi yang dilakukan oleh model.

$$accuracy(a) = \frac{TP+TN+True\ Neutral}{Total\ Data} \quad (10)$$

Precision dihitung dengan membagi jumlah kelas yang diprediksi dengan benar dengan jumlah seluruh kelas tersebut yang diprediksi oleh model. Rumus untuk menemukan nilai *precision* dapat ditulis sebagai berikut.

$$Precision_{Positive} = \frac{True\ Positives_{Positive}}{True\ Positives_{Positive}+False\ Positives_{Positive}+False\ Negative_{Positive}} \quad (11)$$

Sedangkan berikut adalah lanjutan rumus *precision* untuk *precision positive* dan *neutral*

$$Precision_{Positive} = \frac{True\ Positives_{Positive}}{True\ Positives_{Positive}+False\ Positives_{Positive}+False\ Negative_{Positive}} \quad (12)$$

$$Precision_{Neutral} = \frac{True\ Positives_{Neutral}}{True\ Positives_{Neutral}+False\ Positives_{Neutral}+False\ Negative_{Neutral}} \quad (13)$$

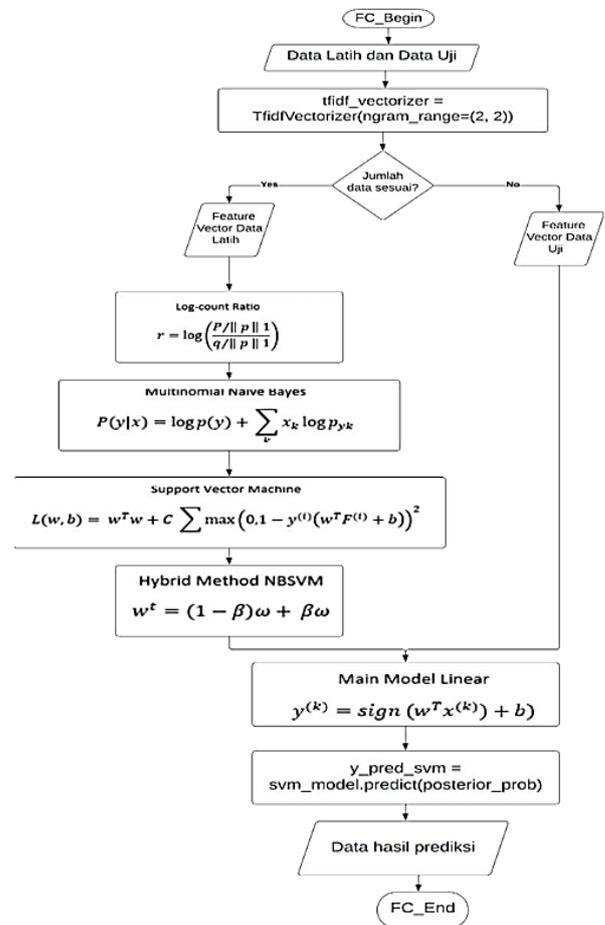
Rumus untuk menghitung nilai *recall* adalah sebagai berikut.

$$Recall_{Positive} = \frac{True\ Positives_{Positive}}{True\ Positives_{Positive}+False\ Positives_{Positive}+False\ Negative_{Positive}} \quad (14)$$

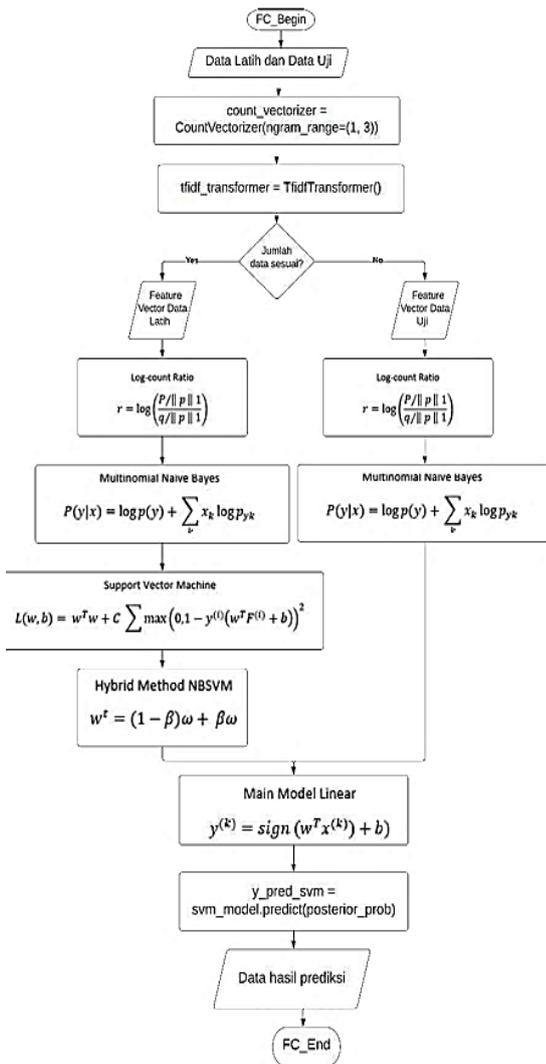
$$Recall_{Negative} = \frac{True\ Positives_{Negative}}{True\ Positives_{Negative}+False\ Positives_{Negative}+False\ Negative_{Negative}} \quad (15)$$

$$Recall_{Neutral} = \frac{True\ Positives_{Neutral}}{True\ Positives_{Neutral}+False\ Positives_{Neutral}+False\ Negative_{Neutral}} \quad (16)$$

Nilai *recall* yang tinggi merupakan indikator performa model cukup baik dalam mengenali sentimen positif yang sebenarnya. Namun, nilai *recall* yang tinggi juga mengindikasikan peningkatan jumlah *false positive*, sehingga diperlukan penghitungan *f1-score*. Nilai *f1-score* menunjukkan keseimbangan antara *precision* dan *recall*. Dalam penghitungannya, *f1-score* memberikan bobot yang lebih besar pada nilai yang rendah, sehingga *f1-score* hanya akan tinggi apabila nilai *precision* dan *recall* keduanya tinggi. Berikut adalah rumus untuk menghitung nilai *f1-score*. Dengan demikian, semakin tinggi nilai akurasi, *precision*, *recall* dan *f1-score*, maka semakin baik kinerja atau performansi dari model dalam melakukan klasifikasi sentimen. Pada Gambar 7 digambarkan proses evaluasi model.



Gambar 5 Diagram Alir Model Klasifikasi Penelitian Terdahulu



Gambar 6 Diagram Model Klasifikasi

Tabel 4 Confusion Matrix

	Predicted Positive	Predicted Negative	Predicted Neutral
Actual Positive	True Positive (TP)	False Positive (FP)	False Positive (FP)
Actual Negative	False Negative (FN)	True Negative (TN)	False Negative (FN)
Actual Neutral	False Neutral	False Neutral	True Neutral

Keterangan :

True Positive (TP) : Jumlah kelas positif yang diprediksi dengan benar sebagai positif

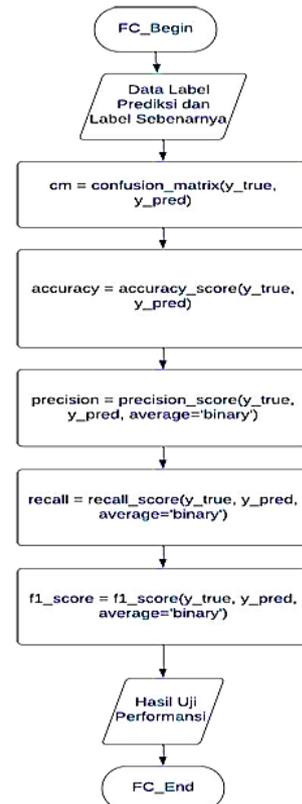
False Positive (FP) : Jumlah kelas positif yang salah diprediksi sebagai negatif atau netral

True Negative (TN) : Jumlah kelas negatif yang diprediksi dengan benar sebagai negatif

False Negative (FN) : Jumlah kelas negatif yang salah diprediksi sebagai positif atau netral

True Neutral : Jumlah kelas netral yang diprediksi sebagai netral

False Neutral : Jumlah kelas netral yang salah diprediksi sebagai negatif atau positif



Gambar 7 Diagram Alir Uji Performansi

### G. Implementasi Model

Pada tahap ini, model klasifikasi yang telah dirancang akan diterapkan sesuai rancangan. Proses penulisan kode untuk tahap text processing serta model klasifikasi menggunakan bahasa pemrograman Python dengan berbagai macam library sesuai dengan kebutuhan penelitian.

```
# @title Import Package
import pandas as pd
import re, string
import nltk
import Sastrawi
import matplotlib.pyplot as plt
import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer
```

```
from sklearn.model_selection import KFold
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from google.colab import drive
from nltk.tokenize import word_tokenize
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

1. Scikit-learn  
*Library* Scikit-learn dirancang untuk menyediakan algoritma pembelajaran mesin termasuk klasifikasi. Untuk pemrosesan data Scikit-learn menyediakan fungsi untuk normalisasi serta ekstraksi fitur.
2. Pandas  
*Library* Pandas memudahkan dalam analisis dan manipulasi data sebab menyediakan fitur-fitur seperti dataframe, series, operasi data, pengolahan data, serta berbagai format input dan output. Dalam penelitian ini *Library*.
3. NumPy  
*Library* NumPy dimanfaatkan untuk menyimpan dan memanipulasi data dalam array, mengakses elemen-elemen individu dari array menggunakan indeks dan slicing untuk mengambil subset dari data.
4. NLTK (Natural Language Toolkit)  
*Library* NLTK umumnya digunakan dalam serangkaian proses pemrosesan bahasa alami atau Natural Language Processing (NLP). Fitur NLTK yang digunakan dalam penelitian ini adalah tokenisasi atau pembagian data teks menjadi token, yakni bagian yang lebih kecil seperti kata atau frasa
5. Sastrawi  
Sastrawi dikembangkan oleh tim dari Universitas Indonesia dan digunakan dalam bahasa pemrograman Python untuk pemrosesan bahasa alami (Natural Language Processing atau NLP). Fitur Sastrawi yang digunakan dalam penelitian ini adalah *stemming* yang memungkinkan perubahan kata-kata ke dalam bentuk dasar atau kata dasar.
6. Matplotlib  
*Library* Matplotlib merupakan pustaka visualisasi data yang populer digunakan dalam ekosistem python. Dalam penelitian ini matplotlib digunakan untuk memvisualisasikan berbagai kata yang muncul dalam dataset yang akan dianalisa melalui *word cloud*.

#### IV. HASIL DAN PEMBAHASAN

##### A. Hasil Crawling Dataset

Pada penelitian ini dilakukan analisis sentimen terhadap opini publik di media sosial Twitter (X) mengenai Program Merdeka Belajar Kampus Merdeka dengan memanfaatkan metode kombinasi Algoritma *Naive Bayes - Support Vector Machine* (NBSVM). Adapun tujuan dari penelitian ini adalah mengetahui cara mengimplementasikan metode *Naive Bayes - Support Vector Machine* untuk melakukan klasifikasi opini publik ke dalam tiga kelas yakni positif, negatif, dan netral.

Berikut hasil dan pembahasan mengenai proses analisis sentimen sesuai dengan alur yang telah dijabarkan sebelumnya.

Pengumpulan dataset dilakukan dengan memanfaatkan *tweet-harvest crawling tool*. Dataset yang berhasil didapatkan berjumlah 621 tweets yang didapatkan melalui media sosial Twitter. Adapun kata kunci yang digunakan untuk melakukan crawling yakni setiap tweets yang mengandung “kampus merdeka”.

##### B. Preprocessing Data

Pemrosesan data bertujuan untuk menghasilkan data yang bersih dari tanda baca, angka, kata-kata noise dan kata-kata stopword, sehingga data sampel hanya berisi kata-kata mengandung sentimen dan memudahkan proses klasifikasi. Adapun tahapan yang dilakukan dalam pre-processing text yakni sebagai berikut: 1) Cleaning dan Casefolding; 2) Tokenizing; 3) Stemming; 3) Stopword Removal.

Tabel 5 Contoh Hasil Text Preprocessing

Data Tweet	
Berkat kampus merdeka akhirnya bisa dpt IPS 4 ❤️	
Tahap Pre-processing Text	Hasil
Cleaning dan Casefolding	Berkat kampus merdeka akhirnya bisa dpt ips
Tokenizing	[berkat, kampus, merdeka, akhirnya, bisa, dpt, ips]
Stemming	[berkat, kampus, merdeka, akhir, bisa, dapat, ips]
Stopword Removal	[berkat, kampus, merdeka]

##### C. Hasil Pelabelan Lexicon Based

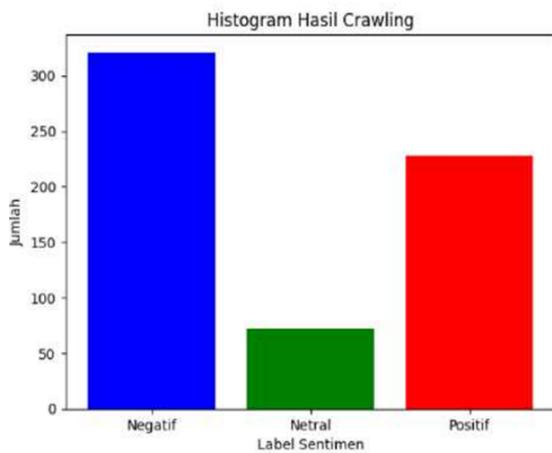
Pengelompokkan dilakukan dengan memberikan skor pada setiap kata berdasarkan skor pada daftar leksikon. Daftar leksikon yang digunakan bersumber dari penelitian berjudul “*InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs oleh Fajri Koto dan Gemala Y. Rahmanytyas*”. Skor setiap kata yang telah dijumlahkan kemudian dibulatkan menjadi nilai 1 untuk setiap dokumen dengan skor lebih dari 0, sedangkan untuk dokumen dengan nilai kurang dari 0 dibulatkan menjadi -1. Contoh pemberian skor pada dokumen dan pembulatan nilai ditampilkan pada Tabel 6. Kemudian dilakukan verifikasi kesesuaian label secara manual terhadap hasil pelabelan. Hasil menunjukkan sebanyak 35% data *tweets* tidak dilabeli dengan tepat. Sehingga perlu dilakukan pelabelan secara manual terhadap data *tweets* tersebut.

Tabel 6 Contoh Penghitungan Skor

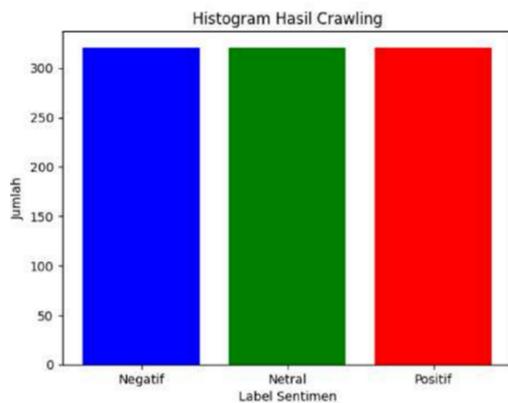
	Tweet
--	-------

		indahny kampus merdeka	nangis ditinggalin kampus merdeka
Skor Kata	indahny	3	-
	nangis	-	-5
	kampus merdeka	0	0
Skor Total		3	-5
Skor Pembulatan		1	-1

Adapun jumlah data yang didapatkan setelah melalui proses pelabelan akhir berjumlah 228 data positif, 321 data negatif, dan 72 data netral. Perbandingan jumlah masing-masing kelas yang direpresentasikan melalui grafik pada Gambar 5.



Gambar 8 Grafik Hasil Pelabelan



Gambar 9 Grafik Hasil Pelabelan Setelah Modifikasi

Berdasarkan Gambar 8, dapat disimpulkan adanya ketidakseimbangan antara data berlabel sentimen positif, negatif dan netral. Hal ini dapat memicu terjadinya overfitting akibat kecenderungan model untuk memperhatikan kelas yang dominan dan mengabaikan kelas yang kurang dominan, sehingga akurasi prediksi dapat menjadi rendah. Oleh karena itu, dilakukan resampling dengan menambahkan data kelas

berlabel sentimen netral secara manual. Adapun jumlah akhir data yang telah mengalami modifikasi adalah 321 data positif, 321 data negatif, dan 321 data netral.

Tabel 7 Contoh Hasil Pelabelan

Tweet	Sentimen	Skor
Berkat kampus merdeka	Positif	1
Gak bisa berharap kampus merdeka magenta bumh	Negatif	-1
Magang merdeka kampus mengajar bisa diikuti mahasiswa semester 4 ga	Netral	0

#### D. Hasil Ekstraksi Fitur CountVectorizer dan TF-IDF Transformer

Ekstraksi fitur dilakukan dengan CountVectorizer dan TF-IDF (Term Frequency-Inverse Document Frequency) Transformer pada teks yang telah dikelompokkan ke dalam bag of words berbentuk unigram. CountVectorizer berguna untuk menghasilkan matriks fitur di mana setiap baris mewakili sebuah dokumen dalam korpus dan setiap kolom mewakili sebuah unigram, bigram, atau trigram yang ditemukan dalam korpus. Nilai dalam matriks adalah frekuensi kemunculan dari token tersebut dalam dokumen. TF-IDF menunjukkan pentingnya token dalam dokumen tertentu dan seberapa jarang token tersebut digunakan di seluruh korpus. Dalam proses ini unit per kata dirangkai untuk membentuk trigram yang terdiri dari dua dan tiga kata. Setiap token akan dihitung frekuensi kemunculannya (Term Frequency).

```
# Inisialisasi CountVectorizer dengan ngram_range untuk unigram, bigram dan trigram
count_vectorizer = CountVectorizer(ngram_range=(1, 3))
tfidf_transformer = TfidfTransformer()
```

```
Streaming output truncated to the last 5000 lines.
berani zona nyaman: 0.13199060386497335
beraniin: 0.13199060386497335
beraniin ikut: 0.13199060386497335
beraniin ikut kampus: 0.13199060386497335
dikiiit: 0.13199060386497335
dikiiit sii: 0.13199060386497335
dikiiit sii wkwkwk: 0.13199060386497335
```

Gambar 10 Contoh Hasil Ekstraksi Fitur dengan TF-IDF

#### E. Hasil Pembagian Data

Pembagian data dilakukan dengan memanfaatkan KFold Cross-Validation dimana dataset dibagi menjadi beberapa fold. Dalam penelitian ini jumlah fold yang digunakan bergantian mulai dari 5, 10, 15, 20 dan 25 fold untuk memastikan nilai mana yang mampu menghasilkan performaterbaik. Dalam penelitian ini jumlah seluruh data tweets adalah 621 tweets, sehingga dimensi yang akan didapatkan untuk data uji adalah seperlima dari jumlah seluruh data. Pada Gambar 11

ditampilkan panjang baris dan dimensi dari data latih dan data uji dalam penelitian ini.

```
# Inisialisasi KFold Cross Validation
kf = KFold(n_splits=25, shuffle=True, random_state=42)

# KFold Cross Validation
for fold, (train_index, test_index) in enumerate(kf.split(X, 1)):
    X_train, X_test = np.array(X)[train_index], np.array(X)[test_index]
    y_train, y_test = np.array(y)[train_index], np.array(y)[test_index]
```

```
dimensi_X_train = X_train.shape
print('Dimensi X_train : ', dimensi_X_train)
panjang_y_train = len(y_train)
print('Panjang y_train : ', panjang_y_train)
dimensi_X_test = X_test.shape
print('Dimensi X_test : ', dimensi_X_test)
panjang_y_test = len(y_test)
print('Panjang y_test : ', panjang_y_test)

Dimensi X_train : (497, 11607)
Panjang y_train : 497
Dimensi X_test : (124, 11607)
Panjang y_test : 124
```

Gambar 11 Contoh Dimensi dan Panjang Vektor Fitur Data Latih dan Data Uji dengan 5 fold

F. Hasil Pelatihan Model dan Klasifikasi Teks  
1. *Multinomial Naive Bayes*

Persamaan (2) dimanfaatkan untuk mengimplementasikan algoritma *Multinomial Naive Bayes*. Penghitungan log-count ratio dan posterior Probability dari setiap kelas disimpan sebagai atribut dari *Multinomial Naive Bayes*. Log-count ratio membandingkan jumlah kemunculan kata yang mungkin memiliki sentimen positif atau negatif dalam kelas sentimen positif dan negatif untuk menghitung peluang kemunculan kata dalam setiap kelas sentimen. Sehingga nilai P didefinisikan sebagai array berisi jumlah kemunculan fitur dalam kelas C. Kemudian Q merepresentasikan array berisi total jumlah kemunculan fitur dalam semua kelas. Sebelum penghitungan log count ratio dilakukan, nilai P dan Q dinormalisasi dengan cara membagi setiap nilai dalam suatu array dengan jumlah total semua nilai dalam array. Terakhir, log-count ratio dihitung dengan mengambil logaritma dari pembagian nilai dalam vektor P yang telah dinormalisasi oleh nilai yang sesuai dalam vektor Q yang telah dinormalisasi.

```
# compute posterior probability
self.class_prior_ = Array of zeros with length equal to the
number of classes
for each class c in list of unique classes:
    compute the occurrence of class c in y_train
    assign the computed occurrence to self.class_prior [1]
# compute log count ratio
self.log_count_ratio = 2D array of zeros with dimensions
(number of classes, number of features)
```

```
for each class c in the list of unique classes
    compute the count of each feature in the current class
    compute the total count of all features in the current class
    compute P (occurrence of each feature in class C)
    compute Q (total occurrence of each feature across all
classes)
    normalize P and Q
    compute log count ratio
    store the log count ratio for class c in
self.log_count_ratio [i]
```

```
Class Prior:
[0.55533199 0.0945674 0.3501006 ]
Log Count Ratio:
[[-0.14835744 0.10663716 0.10663716 ... -0.08902366 -0.09156218
-0.04850569]
[-0.05008985 -0.03788137 -0.03788137 ... 0.00924393 0.00670541
0.0497619 ]
[0.17515904 -0.08648532 -0.08648532 ... 0.07757827 0.08207914
0.00115795]]
```

Gambar 12 Contoh Hasil Nilai Fitur Multinomial Naive Bayes

2. *Support Vector Machine* dengan fitur Naive Bayes Model *Support Vector Machine Classifier* dengan kernel Linier dilatih menggunakan fitur dari dari *Multinomial Naive Bayes*. Vektor bobot SVM dihasilkan dari pelatihan model. Persamaan (5) dimanfaatkan dalam pelatihan model. Selama proses pelatihan, SVM menghitung hyperplane yang memaksimalkan margin. Vektor bobot dalam SVM merupakan koefisien yang menentukan orientasi dan arah dari hyperplane tersebut. Bobot SVM dihitung berdasarkan lokasi sampel data dan pelatihan dalam ruang fitur.

```
#initialize a linear support vector classification (LinearSVC)
model
svm_model = initialize a LinearSVC model

#train the LinearSVC model using combined features from
training data and final labels
svm_model.fit(combined features training, label final)
```

```
Vektor Bobot SVM adalah : [[ 1.76614979e-03 1.57939480e-03 6.07316e-
-5.80041246e-04 -1.55211906e-03]
[-9.93939530e-03 2.37966275e-03 1.09885046e-03 ... -2.06624881e-04
1.65511188e-03 3.74299591e-04]
[-3.17966538e-03 1.03686409e-03 5.31668957e-05 ... 1.83203174e-03
1.50249666e-03 5.18799458e-04]]
Bias untuk SVM adalah : [-0.00989215 -0.01303392 -0.01001039]
```

Gambar 13 Contoh Hasil Vektor Bobot SVM dan Bias SVM

3. Penggabungan Bobot SVM dan *Multinomial Naive Bayes* Selanjutnya dilakukan pengambilan bobot yang dihasilkan oleh SVM. Bobot tersebut akan digabungkan dengan bobot *Multinomial Naive Bayes* sesuai dengan persamaan (6). Bobot ini selanjutnya akan menjadi fitur input untuk prediksi dari Main Model Linear. Adapun untuk proses klasifikasi data uji, digunakan persamaan (1) untuk melakukan klasifikasi terhadap data uji ke-k. Berdasarkan Gambar 12, nilai prediksi didapatkan dari perkalian dot produk antara vektor fitur

TF-IDF data uji dengan bobot NBSVM. Hasil perkalian ini kemudian dijumlahkan dengan nilai bias yang dihasilkan oleh SVM yang telah dilatih oleh fitur dari *Naive Bayes*.

```
#combine naïve bayes and SVM weight vectors
beta = 0.1
NBSVM_weight = (1 - beta) * log_count_ratio_final + beta * w
```

```
Bobot NBSVM :
[[-0.13334508 -0.04492293 0.15770387 ... -0.12038161
 0.13684169]
 [-0.12119412 -0.03152138 0.13710679 ... -0.04367578
 0.00107959]
 [-0.15400554 -0.06514306 0.18747761 ... -0.04347191
 0.00109404]]
```

Gambar 14 Contoh Hasil Penggabungan Fitur NBSVM

G. Hasil Uji Performansi

1. Uji Performansi Algoritma NBSVM

Tabel 8 Hasil Uji Performansi NBSVM

Nilai K	Evaluasi			
	Accuracy	Precision	Recall	F1-Score
5	81%	81%	81%	81%
10	83%	83%	83%	82%
15	83%	83%	84%	83%
20	84%	84%	84%	83%
25	83%	83%	84%	83%

Tabel 8 menunjukkan performa NBSVM secara umum. Dalam hal ini NBSVM diimplementasikan dengan memanfaatkan teknik validasi K-Fold Cross Validation, dengan ekstraksi fitur menggunakan CountVectorizer dan TF-IDF Transformer, serta memanfaatkan SVM dengan Kernel Linear.

Secara umum, kesimpulan yang didapatkan yakni Nilai K yang lebih tinggi umumnya memberikan sedikit keuntungan dalam hal stabilitas dan evaluasi metrik, tetapi setelah K=20, peningkatan tersebut menjadi minimal. Penggunaan K=20 dapat menjadi pilihan optimal, karena memberikan performa tertinggi berdasarkan Tabel 8

Namun, untuk kepraktisan dan menghindari *computational cost* yang terlalu tinggi, K=10 atau K=15 juga merupakan pilihan yang baik karena mereka memberikan hasil yang sangat mirip dengan K=20. Secara keseluruhan, model menunjukkan performa yang kuat dan konsisten di berbagai nilai K, dengan peningkatan stabilitas yang sedikit lebih baik pada nilai K yang lebih tinggi..

2. Perbandingan Rerata Akurasi Algoritma NBSVM untuk kelas Negatif, Netral, dan Positif

Tabel 9 Hasil Uji Performansi NBSVM Berdasarkan Kelas

Nilai K	Algoritma NBSVM		
	Kelas Negatif	Kelas Netral	Kelas Positif
5	81%	83%	80%
10	84%	85%	80%
15	85%	84%	81%
20	84%	85%	81%
25	85%	84%	81%

Ditinjau dari akurasi yang didapatkan untuk setiap kelas, terdapat stabilitas kelas netral dan positif. Akurasi untuk kelas netral dan positif cukup stabil di berbagai nilai K. Ini menunjukkan bahwa NBSVM memiliki performa yang konsisten untuk sentimen netral dan positif. Adapun nilai K yang lebih tinggi (seperti K=15 dan K=25) cenderung meningkatkan akurasi terutama untuk kelas negatif dan mempertahankan stabilitas untuk kelas netral dan positif. Ini menunjukkan bahwa penggunaan nilai K yang lebih tinggi dalam K-Fold Cross Validation dapat bermanfaat untuk meningkatkan keandalan klasifikasi sentimen negatif tanpa mengorbankan akurasi untuk sentimen lainnya.

Secara umum dapat disimpulkan bahwa memilih nilai K yang lebih tinggi seperti K=15 atau K=25 bisa menjadi strategi yang efektif untuk memastikan bahwa semua kelas sentimen diidentifikasi dengan akurasi yang tinggi. NBSVM terbukti sebagai algoritma yang mampu memberikan performa baik dan konsisten dalam klasifikasi sentimen, khususnya pada dataset yang digunakan dalam penelitian ini. Secara keseluruhan, NBSVM menunjukkan akurasi yang baik dan konsisten dalam mengklasifikasikan sentimen negatif, netral, dan positif, dengan performa terbaik dicapai pada nilai K yang lebih tinggi.

3. Perbandingan Rerata Akurasi

Tabel 10 Hasil Uji Performansi Algoritma NBSVM dengan Data Balance dan Imbalance

Nilai K	Algoritma NBSVM	
	Data Balance	Data Imbalance
5	81%	63%
10	83%	63%
15	83%	64%

20	84%	63%
25	83%	63%

Secara umum dapat disimpulkan bahwa data yang seimbang menghasilkan akurasi yang lebih tinggi dan konsisten untuk berbagai nilai K. Ini menunjukkan bahwa NBSVM lebih efektif dalam mengklasifikasikan sentimen ketika data yang digunakan memiliki distribusi kelas yang seimbang. Sedangkan, data yang tidak seimbang menghasilkan akurasi yang jauh lebih rendah, sekitar 63% hingga 64%, yang relatif stabil tanpa adanya peningkatan signifikan dengan nilai K yang lebih tinggi. Ketidakseimbangan data menyebabkan algoritma NBSVM kurang efektif, karena model cenderung lebih condong ke kelas mayoritas dan mengabaikan kelas minoritas. Ditinjau dari pengaruh Nilai K, pada data yang seimbang, peningkatan nilai K dari 5 hingga 20 meningkatkan akurasi, dengan performa terbaik pada K=20 (84%). Pada data yang tidak seimbang, nilai K tidak memberikan pengaruh signifikan terhadap akurasi, menunjukkan bahwa pemilihan nilai K menjadi kurang relevan dalam kondisi ketidakseimbangan data. Secara keseluruhan, Tabel 10 menyoroti pentingnya keseimbangan data dalam mencapai akurasi yang lebih tinggi dengan algoritma NBSVM.

4. Perbandingan Hasil Uji Performansi Algoritma NBSVM menggunakan K-Fold Cross Validation dan Split + K-Fold Cross Validation

Tabel 11 Hasil Uji Performansi Algoritma NBSVM dengan K-Fold Cross Validation dan Split + K-Fold Cross Validation

Nilai K	Algoritma NBSVM	
	K-Fold Cross Validation	Split + K-Fold Cross Validation
5	81%	75%
10	83%	78%
15	83%	79%
20	84%	79%
25	83%	79%

Secara umum, metode K-Fold Cross Validation memberikan hasil yang lebih baik dibandingkan dengan Split + K-Fold Cross Validation dalam hal akurasi. Pemilihan metode validasi harus mempertimbangkan kebutuhan spesifik dari eksperimen, seperti ketersediaan data, kebutuhan komputasi, dan tujuan evaluasi. Berdasarkan Tabel 4.8 dapat disimpulkan bahwa K-Fold Cross Validation memberikan hasil akurasi yang lebih tinggi dan lebih stabil dibandingkan dengan metode Split + K-Fold Cross Validation untuk algoritma NBSVM.

Oleh karena itu, untuk aplikasi analisis sentimen atau tugas serupa, disarankan untuk menggunakan K-Fold Cross Validation untuk mendapatkan evaluasi model yang lebih baik.

5. Perbandingan Hasil Uji Algoritma NBSVM Dengan Data Menggunakan Ekstraksi Fitur Dan Tanpa Ekstraksi Fitur

Tabel 12 Hasil Uji Algoritma NBSVM Dengan Ekstraksi Fitur Dan Tanpa Ekstraksi Fitur

Nilai K	Algoritma NBSVM	
	Dengan Feature Extraction	Tanpa Feature Extraction
5	81%	76%
10	83%	78%
15	83%	77%
20	84%	78%
25	83%	78%

Tabel 12 membandingkan akurasi algoritma NBSVM dengan dua pendekatan berbeda yakni model algoritma NBSVM dengan feature extraction dan tanpa feature extraction. Dalam metode ini, fitur diekstraksi menggunakan CountVectorizer dan TF-IDF sebelum melatih model NBSVM. Hasil akurasi berkisar antara 81% hingga 84% untuk berbagai nilai K. Akurasi tertinggi dicapai pada K=20 dengan 84%.

Sedangkan pada percobaan lain model NBSVM dilatih langsung pada data tanpa ekstraksi fitur tambahan. Hasil akurasi lebih rendah, berkisar antara 76% hingga 78% untuk berbagai nilai K. Tidak ada peningkatan signifikan dalam akurasi dengan peningkatan nilai K. Hal ini menunjukkan penggunaan Menggunakan teknik feature extraction secara konsisten menghasilkan akurasi yang lebih tinggi (81%-84%) dibandingkan tanpa menggunakan feature extraction (76%-78%). Feature extraction membantu meningkatkan representasi data dan mendukung model dalam membuat prediksi yang lebih akurat. Feature extraction meningkatkan kemampuan model untuk menangkap informasi penting dari data, sehingga membantu model untuk belajar lebih baik dan menghasilkan prediksi yang lebih tepat.

Perbedaan akurasi yang signifikan antara dua metode menunjukkan pentingnya penggunaan teknik feature extraction dalam preprocessing data sebelum melatih model NBSVM. Adapun pengaruh nilai K ditinjau tidak memberikan dampak signifikan pada peningkatan nilai K pada percobaan tanpa ekstraksi fitur. Dengan feature extraction, akurasi meningkat secara bertahap dengan peningkatan nilai K, mencapai puncaknya pada K=20 (84%). Tanpa feature extraction, akurasi tidak mengalami peningkatan yang signifikan dengan peningkatan nilai K, tetap rendah dan stabil sekitar 76%-78%. Secara

keseluruhan, penting menggunakan feature extraction dalam meningkatkan performa model NBSVM. Penggunaan feature extraction terbukti lebih efektif dibandingkan tidak menggunakannya, menghasilkan akurasi yang lebih tinggi dan konsisten.

6. Perbandingan Hasil Uji Algoritma NBSVM Kernel Linear dan Polynomial

Tabel 13 Hasil Uji Algoritma NBSVM Kernel Linear dan Polynomial

Nilai K	Algoritma NBSVM	
	SVM Kernel Linear	SVM Kernel Polynomial
5	81%	82%
10	83%	83%
15	83%	84%
20	84%	83%
25	83%	84%

Tabel 13 membandingkan akurasi algoritma NBSVM ketika menggunakan dua jenis kernel pada SVM yakni SVM Kernel Linear dan SVM Kernel Polynomial. Kernel linear merupakan kernel yang sederhana dan langsung memisahkan data dengan hyperplane linier. Akurasi berkisar antara 81% hingga 84% untuk berbagai nilai K. Akurasi tertinggi dicapai pada K=20 dengan 84%. Sementara itu, kernel polynomial dapat menangkap hubungan non-linier dalam data dengan memetakan data ke ruang fitur berdimensi lebih tinggi. Akurasi berkisar antara 82% hingga 84% untuk berbagai nilai K. Akurasi tertinggi dicapai pada K=15 dan K=25 dengan 84%.

Secara keseluruhan, kedua kernel memberikan performa yang baik dengan NBSVM, namun kernel polynomial menunjukkan sedikit keunggulan dalam hal akurasi. Memilih kernel yang tepat sangat tergantung pada sifat dan kompleksitas data yang digunakan.

Berdasarkan Tabel 4.10, dapat disimpulkan bahwa untuk aplikasi analisis sentimen menggunakan NBSVM, kernel polynomial memberikan sedikit keunggulan dalam hal akurasi dibandingkan dengan kernel linear. Namun, perbedaan akurasi tidak terlalu signifikan, sehingga pemilihan kernel juga dapat dipertimbangkan berdasarkan faktor lain seperti kompleksitas data dan kebutuhan komputasi.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Adapun kesimpulan yang dapat diambil dari penelitian analisis sentimen terhadap opini publik mengenai Program Merdeka Belajar Kampus Merdeka dengan memanfaatkan Algoritma *Naive Bayes - Support Vector Machine* (NBSVM) diantaranya:

1. Metode *Naive Bayes - Support Vector Machine* diimplementasikan untuk melakukan klasifikasi data tweets dengan pembangunan model kombinasi metode *Naive Bayes - Support Vector Machine*. Dataset tweets yang telah melalui tahap preprocessing dibagi menjadi data *train* dan data *test* menggunakan K-Fold Cross Validation. Nilai log-count ratio dan probabilitas diekstraksi dari vektor fitur data latih menggunakan *Multinomial Naive Bayes*. Hasil nilai posterior probability dan log-count ratio selanjutnya dijadikan fitur untuk melatih *Support Vector Machine*. Kemudian, SVM mencari loss function untuk menemukan hyperplane terbaik yang bisa menjadi input untuk persamaan main model linear. Persamaan Hybrid Method NBSVM menggabungkan vektor bobot yang dihasilkan oleh *Naive Bayes* dan *Support Vector Machine* sehingga dihasilkan hyperplane terbaik yang dapat memisahkan antara dua kelas secara optimal. Selanjutnya log-count ratio dari data latih dan vektor fitur data uji akan dijadikan input untuk model klasifikasi NBSVM yang telah dilatih. Terakhir, model NBSVM yang telah dilatih menggunakan fitur gabungan dari *Naive Bayes* dan *Support Vector Machine* digunakan untuk melakukan klasifikasi pada data uji.
2. Implementasi metode *Naive Bayes - Support Vector Machine* (NBSVM) menghasilkan akurasi yang paling optimal ketika pembagian data dilakukan menggunakan K-fold Cross Validation dengan nilai fold yaitu 25. Adapun hasil akurasi yang didapatkan sebesar 95%, nilai precision sebesar 95%, recall 93% dan f1-score sebesar 94%.

B. Saran

Berdasarkan penelitian yang telah dilakukan, analisis sentimen positif, negatif, dan netral terhadap opini publik mengenai program Merdeka Belajar Kampus Merdeka, disarankan pengembangan penelitian meliputi:

1. Untuk meningkatkan akurasi dengan memanfaatkan nilai hyperplane terbaik, penggunaan variasi data dengan jumlah yang lebih besar akan membantu model untuk melakukan proses pelatihan dengan data yang lebih beragam.
2. Penggunaan teknik pembagian data yang tepat seperti Stratified K-Fold dapat menjadi solusi untuk data dengan variabilitas data aktual dan data prediksi antar kelas pada setiap fold. Sehingga variabilitas dapat berkurang dan keandalan evaluasi model dapat meningkat
3. Penting untuk memastikan data yang digunakan dalam pelatihan model memiliki distribusi kelas yang seimbang untuk mendapatkan performa yang optimal dari algoritma NBSVM.
4. Dalam situasi di mana data tidak dapat diseimbangkan, teknik penanganan ketidakseimbangan data seperti oversampling kelas minoritas atau undersampling kelas mayoritas harus dipertimbangkan untuk meningkatkan akurasi model. Ketidakseimbangan data secara signifikan menurunkan performa model,

- menunjukkan kebutuhan untuk teknik penyeimbangan data dalam praktek analisis sentimen.
5. Untuk hasil optimal dengan algoritma NBSVM, disarankan untuk selalu menggunakan teknik feature extraction. Meskipun tanpa feature extraction masih dapat digunakan, hasil akurasi lebih rendah dan mungkin tidak cukup memadai untuk banyak aplikasi.
  6. Jika data yang digunakan memiliki kompleksitas yang lebih tinggi atau non-linier, kernel polynomial mungkin lebih cocok karena cenderung memberikan akurasi yang lebih baik. Kernel linear dapat digunakan untuk data yang lebih sederhana dan linier, atau jika kecepatan komputasi menjadi prioritas karena kernel linear biasanya lebih cepat untuk diproses.
  7. Jika tujuan utama penelitian adalah untuk mendapatkan evaluasi yang lebih akurat dari model, K-Fold Cross Validation adalah pilihan yang lebih baik karena memanfaatkan data secara penuh dan memberikan hasil yang lebih andal. Metode Split + K-Fold Cross Validation bisa digunakan ketika ada kebutuhan untuk memisahkan data uji yang tetap dan tidak ingin mengubah set data uji, tetapi harus diperhatikan bahwa ini mungkin menghasilkan akurasi yang sedikit lebih rendah.

#### REFERENSI

- [1] Dirjen Dikti Kemendikbud, "Buku Panduan Pelayanan Merdeka Belajar dan Kampus Merdeka," *Merdeka Belajar-Kampus Merdeka*, pp. 1–33, 2020, [Online]. Available: <http://dikti.kemdikbud.go.id/wp-content/uploads/2020/04/Buku-Panduan-Merdeka-Belajar-Kampus-Merdeka-2020>
- [2] A. Rozaq, Y. Yunitasari, K. Sussolaikah, E. R. N. Sari, and R. I. Syahputra, "Analisis Sentimen Terhadap Implementasi Program Merdeka Belajar Kampus Merdeka Menggunakan Naïve Bayes, K-Nearest Neighbors Dan Decision Tree," *Jurnal Media Informatika Budidarma*, vol. 6, no. 2, p. 746, 2022, doi: 10.30865/mib.v6i2.3554.
- [3] N. R. A. Lubis, "Informasi Berbasis Media Sosial Pada," vol. 8, pp. 53–56, 2022.
- [4] I. Surya, N. Nurmiyati, S. Suranto, and N. Hasanah, "Analysis of The Implementation of Smart Governance through Twitter in Local Governments in Indonesia," *Jurnal Administrasi Publik: Public Administration Journal*, vol. 11, no. 2, pp. 169–184, 2021, doi: 10.31289/jap.v11i2.5596.
- [5] Y. Al Amrani, M. Lazaar, and K. E. El Kadirp, "Random forest and *Support Vector Machine* based hybrid approach to sentiment analysis," *Procedia Comput Sci*, vol. 127, pp. 511–520, 2018, doi: 10.1016/j.procs.2018.01.150.
- [6] R. A. Kashfia Sailunaz, "Emotion and sentiment analysis from Twitter text," *J Comput Sci*, vol. 36, no. 101003, 2019, doi: <https://doi.org/10.1016/j.jocs.2019.05.009>.
- [7] G. S. Andreas C. Muller, "Introduction to Machine Learning with Python: a guide for data scientist," *O'Reilly Media, Inc.*, 2017.
- [8] A. N. Muhammad, S. Bukhori, and P. Pandunata, "Sentiment Analysis of Positive and Negative of YouTube Comments Using Naïve Bayes – *Support Vector Machine* (NBSVM) Classifier," *International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, pp. 199–205, 2019, doi: 10.1109/ICOMITEE.2019.8920923.
- [9] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma *Naive Bayes* Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional," *Jurnal Tekno Kompak*, vol. 15, no. 1, p. 131, 2021, doi: 10.33365/jtk.v15i1.744.
- [10] A. N. Srivasta and M. Sahami, *Text Mining: Classification, Clustering, and Applications*. CRC Press, 2009.
- [11] D. Jurafsky and J. H. Martin, "Hidden Markov and Maximum Entropy Dra," *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*, p. 1004, 2006, [Online]. Available: <papers3://publication/uuid/531A3835-7600-4448-853F-34C2CDA40E8D>
- [12] W. Abdillah, "Metode Penelitian Terpadu Sistem Informasi Pemodelan Teoritis," in *Pengukuran dan Pengujian Statistis*, Yogyakarta: Penerbit And, 2018.
- [13] "About Twitter." Accessed: Mar. 16, 2023. [Online]. Available: <https://help.twitter.com/en/using-twitter>
- [14] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.
- [15] F. Z. Tala, "A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia," 2003.
- [16] F. Koto and G. Y. Rahmaningtyas, "InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs," in *IEEE in the 21st International Conference on Asian Language Processing (IALP)*, Singapore, 2017.
- [17] D. H. Wahid and S. N. Azhari, "Peringkasan Sentimen Esktraktif di Twitter Menggunakan Hybrid TF-IDF dan Cosine Similarity," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 10, no. 2, pp. 207–218, 2016.
- [18] S. Wang, "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers*, 2012, pp. 90–94.