TESIS

ANALISA PERBANDINGAN PENGARUH TEXTBLOB DAN VADER TERHADAP ANALISIS SENTIMEN MENGGUNAKAN METODE NAÏVE BAYES DAN SVM

Disusun oleh:

Nama : **Fernandus Palan Sitorus**
NIM : **21.55.2135**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2024

TESIS

ANALISA PERBANDINGAN PENGARUH TEXTBLOB DAN VADER TERHADAP ANALISIS SENTIMEN MENGGUNAKAN METODE NAÏVE BAYES DAN SVM

COMPARATIVE ANALYSIS OF THE EFFECT OF TEXTBLOB AND VADER ON SENTIMENT ANALYSIS USING NAÏVE BAYES AND SVM METHODS

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama : **Fernandus Palan Sitorus**
NIM : **21.55.2135**
Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2024

HALAMAN PENGESAHAN

ANALISA PERBANDINGAN PENGARUH TEXTBLOB DAN VADER TERHADAP ANALISIS SENTIMEN MENGGUNAKAN METODE NAÏVE BAYES DAN SVM

COMPARATIVE ANALYSIS OF THE EFFECT OF TEXTBLOB AND VADER ON SENTIMENT ANALYSIS USING NAÏVE BAYES AND SVM METHODS

Dipersiapkan dan Disusun oleh

Fernandus Paian Sitorus

NIM 21.55.2135

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jumat, 2 Agustus 2024

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 2 Agustus 2024

Rektor

Prof. Dr. M. Suyanto, M.M
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISA PERBANDINGAN PENGARUH TEXTBLOB DAN VADER TERHADAP ANALISIS SENTIMEN MENGGUNAKAN METODE NAÏVE BAYES DAN SVM

COMPARATIVE ANALYSIS OF THE EFFECT OF TEXTBLOB AND VADER ON SENTIMENT ANALYSIS USING NAÏVE BAYES AND SVM METHODS

Dipersiapkan dan Disusun oleh

Fernandus Paian Sitorus

NIM 21.55.2135

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari Jumat, 2 Agustus 2024

Pembimbing Utama	**Anggota Tim Penguji**
Prof. Dr. Ema Utami, S.Si., M.Kom	**Dhani Ariatmanto, S.Kom., M.Kom., Ph.D**
NIK. 190302037	**NIK. 190302197**
Pembimbing Pendamping	**Tonny Hidayat, S.Kom., M.Kom., Ph.D**
	NIK. 190302182
Mei P. Kurniawan, M.Kom	**Prof. Dr. Kusrini, M.Kom**
NIK. 190302187	**NIK. 190302106**

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 2 Agustus 2024
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom
NIK. 190302106

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama : **Fernandus Paian Sitorus**
NIM : **21.55.2135**
Konsentrasi : **Business Intelligence**

Menyatakan bahwa Tesis dengan judul berikut:
Analisa Perbandingan Pengaruh TextBlob dan VADER Terhadap Analisis Sentimen Menggunakan Metode Naive Bayes dan SVM

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si.,M.Kom.
Dosen Pembimbing Pendamping : Mei P. Kurniawan, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 2 Agustus 2024
Yang Menyatakan,

Fernandus Paian Sitorus

HALAMAN PERSEMBAHAN

Dengan segala kerendahan hari ini, tesis ini saya persembahkan untuk:

1. Alm. Ibu Mirna Luice Silitonga, yang melihat saya dari surga
2. Bapak saya, Edward L. Sitorus.
3. Istri dan anak saya tercinta, Renova Kristelina dan Elvano Kezie Sitorus.

HALAMAN MOTTO

Segala perkara dapat kutanggung di dalam Dia yang memberi kekuatan kepadaku (Filipi 4:13)

Segala sesuatu itu boleh, asal ada alasannya (Prof. Ema)

Pendidikan bukan tentang mengenai mengisi wadah yang kosong, tapi Pendidikan merupakan proses untuk menyalakan api pikiran." - B. Yeats

Memulai dengan penuh keyakinan, menjalankan dengan penuh keikhlasan, menyelesaikan dengan penuh kebahagiaan.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan yang Maha Esa, karena penyertaanNya penulis bisa menyelesaikan tesis dengan judul **Analisa Perbandingan Pengaruh TextBlob dan VADER Terhadap Analisis Sentimen Menggunakan Metode Naive Bayes dan SVM** ini.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada semua pihak yang telah berperan serta dalam penulisan laporan tesis ini, antara lain:

1. Bapak Prof. Dr. M. Suyanto, M.M sebagai Rektor Universitas Amikom Yogyakarta, yang telah memberikan arahan dan dukungan kepada penulis.
2. Ibu Prof. Dr. Kusrini, M.Kom sebagai Direktur Magister Teknik Informatika Universitas Amikom Yogyakarta.
3. Ibu Prof. Dr. Ema Utami, S.Si., M.Kom sebagai Dosen pembimbing 1 sekaligus Wakil Direktur 1 Bidang Akademik Magister Teknik Informatika Universitas Amikom Yogyakarta yang telah memberikan banyak arahan dan dukungan terhadap penulis.
4. Bapak Mei P. Kurniawan, M.Kom sebagai Dosen pembimbing 2 yang telah memberikan banyak arahan dan dukungan terhadap penulis.
5. Seluruh Dosen dan Staf Magister Teknik Informatika yang telah menjalankan sistem perkuliahan di Universitas Amikom Yogyakarta
6. Teman-teman mahasiswa MTI Universitas Amikom Yogyakarta

Akhirnya dengan kerendahan hati penulis menyadari bahwa penulisan tesis ini masih banyak kekurangan, sehingga kritik dan saran yang bersifat

membangun merupakan sesuatu yang sangat diharakan oleh penulis, sehingga nantinya tesis ini dapat memberikan manfaat bagi masyarakat, sekolah dan dinas Pendidikan pada umumnya.

Yogyakarta, 2 Agustus 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

VADER dan TextBlob adalah dua alat untuk analisis sentimen dalam teks. VADER, yang cocok untuk teks informal seperti media sosial, menggunakan kamus kata dan aturan untuk menentukan sentimen. Sedangkan TextBlob adalah pustaka Python yang menyediakan analisis sentimen dan fitur lainnya dengan pendekatan berbasis kamus dan model statistik. Keduanya menawarkan cara berbeda untuk memahami dan mengukur sentimen dalam teks.

Penelitian ini berfokus pada perbandingan model analisis sentimen menggunakan dataset dalam bahasa Indonesia dan bahasa Inggris. Dua model machine learning yang menonjol, yaitu Naive Bayes dan Support Vector Machine (SVM), dievaluasi menggunakan dua leksikon, VADER dan TextBlob, untuk menentukan performa terbaik dalam klasifikasi sentimen. Dataset yang digunakan diperoleh dari Twitter, dengan komentar terkait topik "Kurikulum Merdeka" yang telah dipreproses dan diterjemahkan untuk memastikan konsistensi antar bahasa. Hasilnya menunjukkan bahwa baik SVM maupun Naive Bayes memiliki performa yang lebih baik pada dataset bahasa Inggris dibandingkan dengan dataset bahasa Indonesia, terutama karena kekokohan dan kelengkapan leksikon bahasa Inggris di perpustakaan TextBlob dan VADER. Performa terbaik untuk dataset bahasa Inggris dicapai oleh model SVM, dengan akurasi sebesar 88,73% menggunakan leksikon TextBlob. Untuk dataset bahasa Indonesia, akurasi tertinggi adalah 98,36%, juga menggunakan leksikon TextBlob, namun hasil ini kurang dapat diandalkan karena keterbatasan dukungan bahasa Indonesia dalam leksikon. Dataset bahasa Inggris mengungguli dataset bahasa Indonesia terutama karena leksikon bahasa Inggris dalam TextBlob lebih matang dan lebih sesuai untuk analisis sentimen, sedangkan leksikon bahasa Indonesia masih kurang berkembang, yang menyebabkan anotasi sentimen yang kurang akurat.

Kata kunci: vader, textblob, analisis sentimen, support vector machine, naive bayes

ABSTRACT

VADER and TextBlob are two tools for sentiment analysis in text. VADER, suitable for informal text such as social media, uses a lexicon of words and rules to determine sentiment. TextBlob, on the other hand, is a Python library that provides sentiment analysis and other features using a dictionary-based approach and statistical models. Both offer different methods for understanding and measuring sentiment in text.

This research focuses on comparing sentiment analysis models using datasets in both Indonesian and English. Two prominent machine learning models, Naive Bayes and Support Vector Machine (SVM), are evaluated using the two lexicons, VADER and TextBlob, to determine the best performance in sentiment classification. The dataset used was obtained from Twitter, with comments related to the topic "Kurikulum Merdeka" that have been preprocessed and translated to ensure consistency across languages. The results show that both SVM and Naive Bayes performed better on the English dataset compared to the Indonesian dataset, mainly due to the robustness and completeness of the English lexicons in TextBlob and VADER. The best performance for the English dataset was achieved by the SVM model, with an accuracy of 88.73% using the TextBlob lexicon. For the Indonesian dataset, the highest accuracy was 98.36%, also using the TextBlob lexicon, but this result is less reliable due to the limited support for the Indonesian language in the lexicon. The English dataset outperformed the Indonesian dataset primarily because the English lexicon in TextBlob is more mature and better suited for sentiment analysis, whereas the Indonesian lexicon is still underdeveloped, leading to less accurate sentiment annotation.

Keywords: vader, textblob, sentiment analysis, support vector machine, naive bayes

BAB 1

PENDAHULUAN

1.1. Latar Belakang Masalah

Seiring dengan perkembangan teknologi internet yang semakin maju, banyak media sosial atau layanan jejaring sosial yang memberikan kemudahan kepada penggunanya untuk memposting opini dalam bentuk teks, gambar atau video (Ardiansah & Maharani, 2020). Media sosial mempunyai sifat dua arah yang dapat mempermudah interaksi diantara sesama penggunanya. Selain itu media sosial dapat digunakan sebagai salah satu media untuk meneliti sentimen masyarakat terhadap sebuah kebijakan yang dikeluarkan oleh pemerintah atau terhadap sebuah produk dari sebuah perusahaan. Khususnya pemerintah dapat mengetahui sentimen masyarakan terhadap sebuah kebijakan yang dikeluarkan oleh pemerintah melalui opini yang di berikan oleh masyarakat pada sebuah media sosial. Salah satu media sosial atau layanan jejaring sosial yang saat ini banyak digunakan di seluruh dunia adalah Twitter.

Twitter merupakan salah satu media sosial atau layanan jejaring sosial dibawah perusahaan Twitter Inc. Twitter memberikan layanan jejaring sosial dalam bentuk mikroblog yang memungkinkan penggunanya untuk mengirim dan membaca pesan berupa *Tweets* (Twitter, 2013). Pendapat atau opini dari masyarakat terhadap sebuah kebijakan merupakan yang berharga untuk dianalisis. Hasil dari analisis terhadap opini masyarakat dapat menjadi informasi yang sangat berharga dalam mengambil sebuah keputusan terhadap kebijakan terkait.

Analisis sentimen merupakan sebuah teknik Natural Language Processing yang digunakan untuk mengolah data berupa teks untuk memperoleh informasi dari teks tersebut. Analisis sentimen digunakan untuk mengklasifikasikan opini dalam bentuk teks kedalam 3 (tiga) sentimen yaitu positif, negatif dan netral. Pada proses klasifikasi sentimen, hal pertama yang dilakukan adalah melakukan *cleaning* data untuk mengubah data menjadi terstruktur agar dapat digunakan dalam proses klasifikasi. Setelah itu proses anotasi data, yaitu proses anotasi data berdasarkan sentimen dari opini. Anotasi data merupakan sebuah proses untuk memberi atribut, tanda atau label data dalam membantu algoritma pembelajaran mesin dalam memahami dan mengklasifikasi informasi yang akan diproses. Setelah data di anotasi maka dilakukan peros *preprocessing text* atau pra-pemrosesan teks. Proses ini dilakukan untuk menyeleksi data teks dan mengubah data tersebut menjadi lebih terstruktur dengan serangkaian tahapan diantaranya *cleaning, case folding, tokenizing, filtering* dan *stemming*, sehingga data hasil dari *preprocessing text* tersebut dapat digunakan dalam proses klasifikasi (Ren et al., 2019).

Anotasi data merupakan hal yang penting untuk dilakukan sebagai proses awal dalam pelabelan dari sebuah sentimen dalam kalimat. Proses anotasi memerlukan tenaga yang ahli dibidangnya yang bertugas untuk mengindentifikasi dan melabelkan sebuah kalimat kedalam kelompok sentimen negatif, positif atau negatif. Anotasi yang dilakukan secara manual memerlukan banyak tenaga, waktu dan juga biaya, sehingga banyak penelitian yang menjadi terbengkalai karena hal tersebut. Hal ini patut menjadi perhatian karena dalam sebuah penelitian semakin efektif waktu yang digunakan semakin cepat penelitian tersebut diselesaikan. Saat

ini telah ada *library* dalam *Natural Language Processing* yang dapat melaukan anotasi secara otomatis, diantaranya adalah *TextBlob* dan *VADER*.

TextBlob dan *VADER* merupakan *library* yang sering digunakan untuk melakukan anotasi dan analisis sentimen secara otomatis. Kedua *library* ini merupakan kamus analisis data berbahasa inggris dengan memperhatikan kekuatan emosional yang sesuai dengan kamus data *lexicon* yang tersedia. Perbedaan antara *TextBlob* dan *VADER* terletak pada proses anotasi yang dilakukan. *TextBlob* merupakan *library* dari *phyton* yang memiliki fitur dasar *Natural Language Processing* (NLP) yang digunakan untuk menganotasi dan menganalisis data sentimen, dimana emosi dianotasi menggunakan model *Text2Emotion* atau dengan menghitung nilai polaritas dan subjektivitas dari setiap kata atau kalimat (Aslam et al., 2022). Sedangkan *VADER* (*Valence Aware Dictionary for Sentiment Reasoning*) merupakan salah satu *library* untuk melakukan anotasi data teks menggunakan kamus *lexicon* berbahasa inggris (en) dimana emosi dianotasi menggunakan nilai *compound* (Abimanyu et al., 2022). Selain itu *VADER* befokus pada analisis sentimen dari data yang berada pada media sosial. Dalam sebuah analisis sentimen dengan menggunakan data atau kalimat yang sama kedua *library* yaitu *TextBlob* dan *VADER* selalu menghasilkan skor yang berbeda dan biasanya lebih tinggi skor pada *library VADER*. Selain itu, *TextBlob* dan *VADER* mempunyai perbedaan dalam menanggapi huruf kapitalis, dimana *VADER* menganggap versi kata yang dikapitalisasi memiliki sentimen yang lebih kuat dan meningkatkan skor sentimen, sedangkan *TextBlob* tidak membedakan sentimen antara versi huruf besar dan kecil dari kata tersebut. Selain itu *VADER* juga menanggapi kata berulang

dengan meningkatkan nilai skor *compound*, sedangkan *TextBlob* tetap menganggap kata berulang adalah kata yang sama dan tidak mengubah skor awal. Hasil akurasi klasifikasi dengan menggunakan beberapa metode algoritma klasifikasi pada beberapa penelitian sebelumnya dengan menggunakan dataset hasil anotasi dari *library TextBlob* dan *VADER* cenderung rendah atau berada pada rentang nilai 70% sampai 73%, walaupun pada beberapa penelitian yang menghasilkan nilai akurasi tinggi, namun penelitian tersebut tidak menjelaskan secara detail hasil yang diperoleh, berapa banyak dataset yang digunakan, bagimana cara menganotasi data menggunakan *library TextBlob* dan kebanyakan data yang digunakan sudah diberi *rating* atau skor dari pengguna sehingga lebih mudah dalam menganotasi data tersebut. Hal ini menjadi perhatian bagi peneliti untuk meneliti lebih dalam terkait kinerja kedua kamus *lexicon* yaitu *TextBlob* dan *VADER* dalam melakukan anotasi dan analisis.

Beberapa penelitian telah dilakukan oleh para peneliti yang bertujuan untuk membandingkan hasil klasifikasi sebuah mesin pembelajaran dengan menggunakan *library TextBlob* dan *VADER* sebagai tools untuk melakukan anotasi otomatis pada data latih yang akan digunakan pada pelatihan mesin pembelajaran. Penelitian yang dihasilkan oleh (Lestari et al., 2020) menggunakan dataset yang di *crawling* dari twitter dengan *hastag* indihome dan first media dan *libary TextBlob* sebagai pustaka anotasi datanya menunjukkan bahwa indihome memperoleh respon negatif sebanyak 135 respon lebih besar daripada respon positif sebanyak 58 respon, sedangkan first media memperoleh 54 respon positif dan 141 respon negative. Dari penelitian tersebut didapatkan nilai akurasi sebesar 74%, *recall* 66% dan presisi

83%. Dari hasil akurasi yang didapatkan menunjukkan bahwa adanya pengaruh anotasi data menggunakan *TextBlob* terhadap hasil klasifikasi sehingga hasil akurasi yang diperoleh kurang optimal. Selain itu, penelitian yang dilakukan oleh (Azhar et al., 2022) menggunakan dataset yang di *crawling* dari *twitter* dengan *hastag #crypto* sebanyak 1032 baris *tweet*. Penelitian tersebut menggunakan *library TextBlob* untuk proses anotasi dan analisis sentimen data secara otomatis dan menghasilkan jumlah sentimen positif sebanyak 61.24%, sentimen netral sebanyak 26.68% dan sentimen negatif 10.07%. Dari proses pengujian yang dilakukan menggunakan metode *Naive Bayes* dengan pembagian data *testing* sebanyak 20% dan data *training* sebanyak 80% menghasilkan nilai akurasi sebesar 71,89%, presisi 83,04%, *recall* 60,88% dan *f1-score* 71,98%. Dari akurasi yang dihasilkan dapat dilihat bahwa anotasi menggunakan *TextBlob* masih belum menghasilkan akurasi terbaik. Penelitian yang dilakukan oleh (Afrillia et al., 2022) bertujuan untuk menganalisis tanggapan masyarakat terhadap penerapan permendikbud PPKS. Dataset yang digunakan berjumlah 159 baris data yang di *crawling* dari *twitter* dengan menggunakan *hastag* #Kesetaraan gender, kekerasan seksual, dan PPKS. Pada dataset dilakukan identifikasi dan pengkategorian dengan menggunakan *library TextBlob* untuk menentukan teks tersebut bernilai positif atau negatif. Pembagian data uji dan data latih sebesar 70:30 secara random. Hasil pengujian menggunakan algoritma Support Vector Machine (SVM) menghasilkan akurasi sebesar 70,8% dimana hasil tersebut belum mengalami peningkatan dari hasil penelitian sebelumnya.

Penelitian yang dilakukan oleh (Al-Shabi, 2020) menggunakan *library lexicon* diantaranya adalah *VADER, SentiWordNet, SentiStrength, Liu and Hu opinion* dan *AFINN-111*. Tujuan penelitian ini untuk mengevaluasi anotasi otomatis berbasis *library lexicon* yang paling berpengaruh dalam bidang analisis sentimen pada data *twitter*. Dataset yang digunakan dibagi menjadi 2 (dua) grup dimana grup pertama berisi 1.600 komentar dan grup kedua berisi 4.000 komentar terhadap permasalahan sistem pelacakan aplikasi JIRA. Hasil dari penelitian yang telah dilakukan menghasilkan akurasi terhadap masing-masing *lexicons* adalah *VADER* sebesar 72%, *SentiWordNet 53%, SentiStrength* sebesar 67%, *Liu and Hu opinion* sebesar 65% dan *AFINN-111* sebesar 65%. Hasil akurasi tersebut belum dapat dikatakan sebagai akurasi terbaik, karena masih berada dibawah 85%.

Penelitian yang dilakukan oleh (Baita & Cahyono, 2021) bertujuan untuk menganalisa sentimen dari opini masyarakat terhadap pemberian vaksin *sinovac* saat pandemi COVID-19. Dataset yang digunakan sebanyak 2105 baris *tweet* dengan kata kunci '***sinovac***', tanpa *retweet*. *Lybrary TextBlob* digunakan sebagai anotasi otomatis, dan algoritma klasifikasi yang digunakan untuk membangun model klasifikasi adalah *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN). Dari penelitian tersebut dihasilkan nilai akurasi sentimen sebesar 0,70 untuk kernel linear SVM, 0,57 untuk kernel polynomial SVM, 0,66 untuk kernel RBF SVM, sedangkan untuk akurasi KNN sebesar 0,55 untuk n=3, 0,55 untuk n=5 dan 0,56 untuk n=7. Selain itu, penelitian yang dilakukan oleh (Irfan et al., 2022) menggunakan *library TextBlob* sebagai *tools* untuk anotasi otomatis dan algoritma *Random Forest* sebagai metode yang digunakan dalam membangun model

klasifikasi teks. Pada penelitian tersebut dilakukan pembobotan kata menggunakan TF-IDF dengan tujuan agar dimengerti oleh mesin. Dari hasil penelitian tersebut diperoleh hasil akurasi dari performa model sebesar 76%. Hasil yang diperoleh sudah baik, namun masih belum memenuhi standar kelayakan dalam analisis sentimen, dimana nilai akurasi yang dianggap layak adalah diatas 80%.

Berdasarkan penelitian-penelitian sebelumnya, ditemukan beberapa masalah, antara lain rendahnya akurasi analisis sentimen yang dihasilkan menggunakan dataset yang di anotasi menggunakan *TextBlob* dan *VADER*, dimana nilai rata-rata akurasi berada dibawah 80%, dan tidak ditemukan adanya penelitian yang memperhatikan pengaruh dari hasil anotasi *TextBlob* dan *VADER* terhadap setiap pemrosesan data teks yang bertujuan untuk meningkatkan nilai akurasi analisis sentimen. Oleh karena itu, penelitian ini berusaha untuk menganalisis bagaimana peningkatan akurasi analisis sentimen dapat dicapai dengan mengoptimalkan proses anotasi menggunakan *TextBlob* dan *VADER*. Penelitian ini akan mengkaji berbagai metode dan teknik yang dapat diterapkan untuk meningkatkan kualitas anotasi sentimen, serta mengevaluasi dampaknya terhadap performa model analisis sentimen yang dihasilkan. Dataset yang digunakan pada penelitian ini adalah dataset bahasa Indonesia dan bahasa Inggris. Dataset yang di *crawling* pada awalnya adalah dataset yang berasal dari *tweet* bahasa Indonesia dan akan di translate ke bahasa Inggris, sehingga menghasilkan dataset bahasa Inggris. Hasil anotasi data bahasa Inggris dan bahasa Indonesia dalam analisis sentimen bertujuan untuk mengidentifikasi dan membandingkan pola-pola ekspresi emosional dalam kedua bahasa. Anotasi ini akan membantu dalam memahami

bagaimana sentimen positif, negatif, dan netral diekspresikan dalam bahasa yang berbeda. Dengan menganalisis hasil anotasi ini, penelitian bertujuan untuk meningkatkan akurasi model analisis sentimen dengan mempertimbangkan faktor linguistik dan budaya yang mempengaruhi sentimen dalam kedua bahasa. Selain itu, pemahaman yang lebih mendalam mengenai dinamika sentimen ini dapat berkontribusi pada pengembangan model analisis sentimen yang lebih efektif dan akurat untuk berbagai bahasa, sehingga dapat diterapkan secara luas dalam berbagai konteks penelitian. Algoritma yang digunakan sebagai metode klasifikasi guna memvalidasi dan menganalisis sentimen dari hasil anotasi sentimen otomatis berbasis *lexicon* yaitu *Naive Bayes* (NB) dan *Support Vector Machine* (SVM). Kedua metode ini merupakan metode yang mempunyai performa sangat baik dan sering digunakan oleh para peneliti dalam membangun model mesin klasifikasi sentimen. Penggunaan metode *Naive Bayes* (NB) dan *Support Vector Machine* (SVM) sebagai algoritma klasifikasi bertujuan untuk memvalidasi hasil anotasi sentimen otomatis menggunakan *TextBlob* dan *VADER*, dan membangun model mesin yang mampu menghasilkan tingkat performa terbaik dengan nilai akurasi maksimal atau berada diatas 80%. Dalam hal ini studi kasus yang diangkat adalah tentang kurikulum merdeka dan penerapan kurikulum merdeka sebagai kurikulum nasional tahun 2024.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan diatas, maka penulis dapat menyusun rumusan masalah sebagai berikut:

a. Variabel apa saja yang mempengaruhi hasil akhir dari berbagai macam anotasi menggunakan *library TextBlob* dan *VADER* yang dapat mempengaruhi akurasi klasifikasi sentimen menggunakan algoritma *Naive Bayes* (NB) dan *Support Vector Machine* (SVM)?
b. Berapakah peningkatan akurasi yang bisa capai dari penelitian yang diusulkan dengan melakukan berbagai anotasi dari beberapa tahapan pemrosesan teks dibandingkan dengan hasil dari *state of the art research?*
c. Bagaimana performa hasil anotasi terhadap akurasi klasifikasi sentimen pada dataset bahasa indonesia dan bahasa inggris yang di-*crawling* dari media sosial, khususnya dalam konteks kebijakan kurikulum merdeka?

1.3. Batasan Masalah

Batasan-batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut:

a. Dataset yang digunakan pada penelitian ini adalah opini publik berbentuk *tweet* tentang 'kurikulum merdeka' dan 'penerapan kurikulum merdeka sebagai kurikulum nasional tahun 2024'.
b. Proses anotasi data menggunakan alat anotasi otomatis yaitu *TextBlob* dan *VADER*.
c. Penelitian memperhatikan perubahan hasil anotasi sentimen dari setiap tahapan pemrosesan teks yaitu *Cleaning*, *Spellcheker*, *Casefolding*, dan *Drop duplicate data* pada dataset menggunakan *library TextBlob* dan *VADER*.

d. Metode pengklasifikasi yang di usulkan untuk membangun model klasifikasi adalah *Naive Bayes* (NB) dan *Support Vector Machine* (SVM).

1.4. Tujuan Penelitian

Adapun tujuan penulis dalam melakukan penelitian ini adalah sebagai berikut:

a. Untuk mengetahui penyebab rendahnya akurasi hasil klasifikasi yang dihasilkan dari sentimen analisis dengan menggunakan data hasil anotasi dari *library TextBlob* dan *VADER*.
b. Untuk menemukan tahapan terbaik dalam proses anotasi data otomatis menggunakan *library TextBlob* dan *VADER* yang dapat menghasilkan akurasi klasifikasi terbaik.
c. Untuk mengetahui perbandingan kinerja anotasi dari *library TextBlob* dan *VADER*.
d. Untuk mengetahui apakah ada peningkatan akurasi yang signifikan dari penelitian yang dilakukan dibandingkan dengan hasil dari *state of the art research?*

1.5. Manfaat Penelitian

a. Menjadi referensi bagi peneliti pada penelitian sentimen analisis dalam melakukan proses anotasi data dan klasifikasi sentimen menggunakan *library TextBlob dan VADER*.

b. Menjadi masukan kepada pemerintah dalam menentukan kebijakan terkait kurikulum pendidikan yang sesuai dengan opini masyarakat.

c. Berkontribusi secara ilmiah terhadap penelitian dibidang studi *Natural Language Processing* (NLP), khususnya pada penelitian tentang sentimen analisis dengan menggunakan *library TextBlob* atau *VADER* sebagai penganotasi data otomatis.

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian yang dilakukan oleh (Nur Syahirah Wan Min et al., 2020) bertujuan untuk menganalisa kinerja dari 2 (dua) *lexicon* yaitu *TextBlob* dan *VADER*. Dataset yang digunakan merupakan data *tweet* dari *twitter* oleh 1 (satu) pengguna sepanjang tahun 2013 sampai dengan 2019 yang berjumlah sebanyak 7.997 baris *tweet*. *TextBlob dan VADER* masing-masing digunakan untuk melabelkan sentimen dari 7.997 baris data dan diluar data tersebut dipilih 300 data *tweet* yang dipilih secara acak untuk dilabelkan oleh 3 (tiga) ahli psikolog secara manual. Dari penelitian tersebut ditemukan bahwa kedua *lexicon* menghasilkan tingkat akurasi yang masih dapat diterima yaitu 79% untuk *VADER* dan 73% untuk *TextBlob*. Untuk hasil akurasi dari data yang dilabelkan secara manual memiliki tingkat akurasi yang jauh lebih baik dari kedua *library lexicon*. Kesimpulan dari penelitian tersebut adalah kinerja dari *VADER* lebih baik dibandingkan dengan *TextBlob*. Karena analisis sentimen berbasis *lexicon* berbasis aturan, maka kinerja analisis sentimen sangat bergantung pada kualitas kamus *lexicon*. Pada penelitian ini, penulis akan menggunakan *TextBlob* dan *VADER* hanya sebagai penganotasi, sedangkan model klasifikasi akan dibangun menggunakan *Naive Bayes* dan SVM untuk mencari nilai akurasi yang lebih baik dari penelitian sebelumnya.

Penelitian yang dilakukan oleh (Mas Diyasa et al., 2021) adalah mengklasifikasi opini masyarakat khususnya pengguna jasa PT. Telkom indonesia

terhadap produk yang ditawarkan yaitu diantaranya indihome, myindihome, useetv, dan wifi.id. Penelitian tersebut bertujuan untuk menilai brand image, umpan balik terhadap kostumer, dan peluang pemasaran. Dataset yang digunakan adalah data dari *twitter* sebanyak 3324 baris *tweet* yang proses anotasi dilakukan secara otomatis menggunakan *library TextBlob*. Proses *preprocessing* data diantarnya visualisasi dalam bentuk histogram, diagram lingkaran, dan awan kata. Hasil dari analisa yang dilakukan dari 3324 *tweet* menghasilkan 1590 data netral diantaranya 1266 data valid dan 324 invalid, lalu 1082 data positif diantaranya 858 valid dan 224 invalid, selain itu 518 negatif diantarnya 443 valid dan 75 invalid. Dari hasil tersebut maka prosentasi hasil analisis adalah 34,4% *tweet* positif, 16,1% *tweet* negatif, dan 49,6% *tweet* netral dengan akurasi analisis sebesar 77,2%. *Preprocessing* teks tidak dilakukan dalam penelitian tersebut, sehingga penulis pada penelitian ini akan melakukan *preprocessing* teks sebelum menganotasi data teks menggunakan *TextBlob* dan *VADER* untuk membangun model klasifikasi sentimen.

Penelitian berikutnya dilakukan oleh (Suanpang et al., 2021) dimana penelitian tersebut bertujuan untuk melakukan analisis sentimen menggunakan *library TextBlob* pada bisnis pariwisata dengan studi kasus merangsang ekonomi pariwisata pasca COVID-19 di thailand. Dataset yang digunakan pada penelitian adalah dataset yang didapatkan dari *kagle* dengan jumlah data sebanyak 10.000 baris data dengan *rating* nilai dari 1, 2, 3, 4 dan 5 dari catatan perjalanan para turis yang sudah singgah atau berwisata di thailand. Informasi *review* dari kepuasan wisata oleh turis dapat digunakan untuk meningkatkan pelayanan bisnis pariwisata

yang ada di thailand. Algoritma yang digunakan dalam klasifikasi pada mesin pembelajaran adalah *Naive Bayes*. Hasil dari analisis dari model yang telah dibangun didapatkan nilai akurasi sebesar 89,32%. Namun data yang dihasilkan bisa jadi bervariasi tergantung pada banyaknya data latih dan data uji yang di implementasikan untuk proses klasifikasi sentimen. Dengan penelitian ini, penulis akan memperbanyak dataset untuk membandingkan hasil penelitian dengan penelitian terdahulu. Penelitian berikut yang dilakukan oleh (Su, 2022) bertujuan untuk melakukan analisis dari opini publik terhadap 2 (dua) film yaitu *The Return of the King* and *The Lord of the Rings*. Dataset yang digunakan merupakan data yang di *crawling* dari ulasan pada IMBD. Sebelum dataset digunakan, dilakukan proses penghapusan data duplikat dan *filtering*. Jumlah data yang dihasilkan setelah melalui proses penghapusan duplikat data dan *filtering* data sebanyak 1.500 ulasan. *Library TextBlob* digunakan untuk menilai skor sentimen dari kedua film tersebut. Term frequency-Inverse Document Frequency (TF-IDF) digunakan untuk mengevaluasi pentingnya setiap kata yang ada dalam ulasan. Algoritma yang digunakan dalam model klasifikasi sentimen yang dibangun terhadap ulasan film *The Return of the King* and *The Lord of the Rings* adalah *Support Vector Machine* (SVM) yang menghasilkan akurasi sebesar 85,2%. Penelitian tersebut menghasilkan nilai akurasi sedikit lebih tinggi dibandingkan penelitian sebelumnya, namun yang menjadi perhatian pada penelitian ini adalah setiap ulasan yang ada telah diberikan skor terlebih dahulu berbeda dengan penelitian lainnya yang dataset nya belum diberikan nilai. Nilai skor tersebut merupakan nilai skor antara 1 sampai 5, dimana nilai 1 dan 2 merupakan sentimen negatif, nilai 3

merupakan sentimen netral, sedangkan nilai 4 dan 5 merupakan sentimen positif. Pada penelitian ini penulis akan menggunakan dataset tanpa nilai skor dan dianotasi secara otomatis menggunakan *TextBlob* dan *VADER* dan membandingkan hasil akurasi yang diperoleh dengan hasil dari penelitian sebelumnya.

Penelitian yang dilakukan oleh (Alenzi et al., 2022) bertujuan untuk menganalisis dan memvalidasi kinerja dari 2 (dua) anotasi otomatis berbasis *lexicon* yang banyak digunakan yaitu *TextBlob* dan *Valence Aware Dictionary and Sentiment Reasoner* (*VADER*) dengan membandingkan kinerja kedua *library* dengan anotasi yang dilakukan secara manual. Dataset yang digunakan berasal dari *twitter* yang dikumpulkan sebanyak 25.800 *tweets* dengan kata kunci menggunakan bahasa dan tulisan arab. Dari 25.800 baris data teks berisi 3124 tweet positif, 1463 tweet negatif, dan 815 tweet netral. *Tweet* tersebut diterjemahkan kedalam bahasa inggris sehingga dapat di anotasisi secara otomatis menggunakan *TextBlob* dan *VADER*. Pada studi tersebut menunjukkan bahwa anotasi secara otomatis tidak dapat digunakan sebagai standar yang baik dalam anotasi data teks. Banyak kekurangan dan keterbatasan yang ditemukan pada anotasi otomatis menggunakan algoritma berbasis *lexicon*. Tingkat akurasi tertinggi yang diperoleh dari serangkaian percobaan yang telah dilakukan adalah 75% untuk *TextBlob* dan 70% untuk *VADER*. Pada penelitian ini penulis akan melakukan experimen terhadap penggunaan anotasi otomatis berbasis *lexicon* yaitu *TextBlob* dan *VADER*, lalu membandingkan hasil riset dengan hasil riset terdahulu.

Penelitian lainnya terkait *TextBlob* dan *VADER* dilakukan oleh (Abiola et al., 2023) tentang analisis sentimen dari *tweet* dengan kata kunci yang digunakan

adalah *COVID-19 in Nigeria*. Penelitian tersebut bertujuan untuk menganalisis respon emosional terhadap pandemi *virus corona* (*COVID-19*) menggunakan penganotasi dan penganalisis berbasis *lexiconi* yaitu *library TextBlob* dan *VADER*. Dataset yang digunakan sebanyak 1.048.575 baris *tweet* yang dikumpulkan dari *twitter* menggunakan teknik *crawling* data. Dataset sebelumnya diproses terlebih dahulu menggunakan *tokenizer tweet*, sementara *TextBlob* dan *VADER* digunakan untuk mengaonatis data secara otomatis. Dari penelitian tersebut didapatkan hasil sentimen menggunakan *VADER* adalah 39,8% sentimen positif, 31,3% netral, dan 28,9% sentimen negatif, sedangkan *TextBlob* menghasilkan 46,0% sentimen positif, 36,7% netral, dan 17,3%, sentimen negatif. Pada penelitian tersebut tidak dilakukan perhitungan akurasi, sehingga tidak diketahui seberapa akurat data yang dihasilkan dengan menggunakan *library TextBlob* dan *VADER* sebagai penganotasi dan penganalisis data teks. Pentingnya akurasi dalam analisis sentimen berpengaruh pada kepercayaan pengguna terhadap model yang dibangun, jika tidak ada nilai akurasi maka pengguna tidak akan percaya dengan hasil riset yang dipaparkan.

Penelitian lain yang dilakukan oleh (Dewi & Arianto, 2023) tentang sentimen analisis *twitter* terhadap Qatar sebagai tuan rumah piala dunia 2022 menggunakan *library TextBlob* sebagai *tools* anotasi otomatis dimana penelitian tersebut hanya menggunakan 2 (dua) label pada pengkategorian data yaitu positif dan negatif. Dataset yang digunakan adalah dataset yang di *scraping* dari *twitter* dalam 3 (tiga) tahap yaitu *scraping* pertama dilakukan dari tanggal 1 maret 2010 sampai 1 desember 2010 menghasilkan 3606 baris data *tweet*, lalu tahap kedua dari tanggal 2 desember 2010 sampai 19 november 2022 menghasilkan 41225 data

tweet, dan tahap ketiga dari tanggal 20 november 2022 sampai dengan 18 desember 2022 menghasilkan 200.000 data *tweet*. Total data yang *discraping* sebanyak 244.832 baris dari ketiga tahap *scraping* yang telah dilakukan. Hasil analisis dari tahap pertama sebelum Qatar menjadi tuan rumah adalah sentimen sentimen positif sebesar 88,46% dan negatif sebesar 11,54%, lalu hasil pada tahap kedua setelah Qatar terpilih menjadi tuan rumah terdapat sentimen positif 79,38% dan sentimen negatif 20,62%, dan tahap ketiga saat piala dunia 2022 berlangsung di Qatar terdapat sentimen positif 83,72% dan sentimen negatif 16,28%. Dari hasil penelitian didapatkan akurasi dalam klasifikasi sentimen sebesar 83% dimana hasil tersebut merupakan kemampuan dari model untuk memprediksi data uji. Pada penelitian ini penulis akan menganalisis seberapa besar pengaruh kinerja *TextBlob* dan *VADER* terhadap analisis sentimen menggunakan perhitungan nilai polaritas dan *compound* dari beberapa tahapan anotasi yang berbeda dengan memanfaatkan kamus *lexicon TextBlob* dan *VADER* sehingga dapat meningkatkan akurasi dari penelitian sebelumnya.

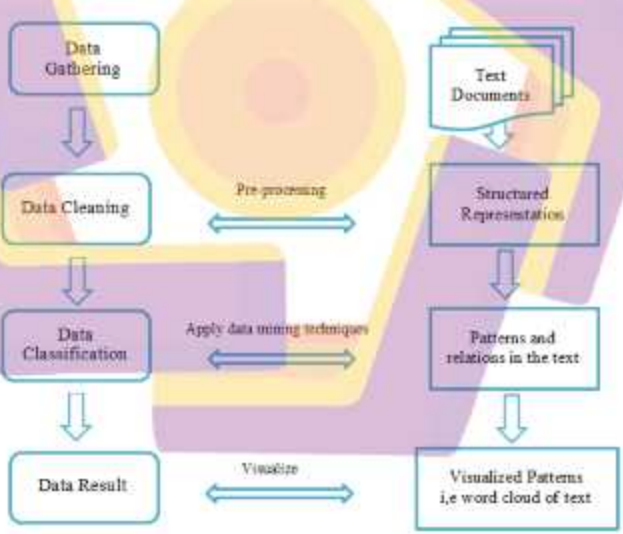
2.2. Landasan Teori

2.2.1. Analisis Sentimen

Analisis sentimen atau sering dikenal sebagai *Opinion Mining* merupakan pemrosesan teks *analitycs* dari berbagai sumber seperti opini, sentimen, perilaku, penilaian dan emosi terhadap suatu entitas, yaitu seperti produk, pelayanan, organisasi, dan individu opini dari pengguna pada sebuah platform dengan tujuan untuk memperoleh informasi (Ayu Muthia, 2017). Analisis sentimen merupakan

bagian dari *Natural Language Processing* (NLP) yang bertujuan untuk mengenali dan mengekstrasi opini kedalam bentuk teks.

Menurut (Aulia & Patriya, 2019) sentimen analisis merupakan proses *text mining* dalam menganalisa pendapat, opini, sentimen, sikap dan penilaian seseorang terhadap individu, kelompok, produk, organisasi, masalah, peristiwa atau topik. Selain itu analisis sentimen juga dapat diartikan sebagai sebuah riset komputasional dari satu atau berbagai opini yang diekspresikan secara tekstual. Nilai dari analisis sentimen dapat dikelompokkan dalam 3 (tiga) nilai sentimen yaitu sentimen positif, negatif dan netral atau dapat diperdalam lagi sehingga dapat menemukan dari mana sumber sentimen positif atau negatif berasal.



Gambar 2.1. Alur Analisis Sentimen (Arispe et al., 2019)

1. Data *Gathering* (Pengumpulan data)

   Data *Gathering* atau proses pengumpulan data adalah sebuah aktivitas dalam pencarian data yang akan digunakan dalam sebuah penelitian untuk mencapai tujuan dari peneilitan tersebut. Pengumpulan data pada penelitian ini dilakukan dengan cara *crawling data* dari *twitter*. Data yang dikumpulkan adalah data *tweet* berupa opini dari masyarakat tentang "Kurikulum merdeka" dan "Penerapan Kurikulum Merdeka sebagai Kurikulum Nasional 2024".

2. Data *Cleaning* (Pembersihan Data)

   Data *cleaning* atau pembersihan data merupakan prosedur dalam memastikan suatu kebenaran, konsistensi, dan kegunaan dari suatu data pada suatu dataset. Proses tersebut dilakukan dengan cara mendeteksi adanya error atau corrupt pada data, kemudian dilakukan perbaikan atau penghapusan data jika hal tersebut dibutuhkan.

3. Data *Classification* (Klasifikasi Data)

   Klasifikasi merupakan sebuah metode yang sering digunakan dalam proses data *mining*. Klasifikasi dapat diterapkan dengan dengan beberapa algoritma dalam pembuatan model mesin pembelajaran dengan tujuan untuk memprediksi data. Dengan kata lain klasifikasi merupakan teknik penambangan data yang mengkelompokkan data kedalam beberapa kelas yang telah ditentukan. Metode ini merupakan metode pembelajaran yang diawasai dengan menggunakan datalatih yang telah dilabelkan untuk menghasilkan aturan untuk mengklasifikasikan data uji ke dalam kelas yang telah ditentukan.

4. Data *Result* (Data Hasil)

Data hasil merupakan hasil dari sebuah penelitian bisa berbentuk hasil data, metode, model solusi, mesin, peralatan, perangkat lunak, zat, atau data lainnya yang serupa atau bagiannya, terlepas dari apakah hasil tersebut dilindungi atau mungkin dilindungi oleh hak-hak immaterial (Law Insider, 2023).

2.2.2. *TextBlob*

Textblob merupakan salah satu *library python* berbasis *lexicon* dikembangkan oleh Steven Loria dan sering digunakan untuk menagalisa sentimen dari sebuah teks dan mengkategorikan data tersebut kedalam 3 (tiga) sentimen yaitu positif, netral dan negatif (Barai, 2021). Dasar dari *TextBlob* adalah menggunakan *Natural Language Tools* (NLTK) dimana *input* nya berisi satu kalimat yang akan dianalisis dan *output* dari *TextBlob* adalah berupa polaritas dan subjektivitas.

- Nilai polaritas terletak antara -1 sampai dengan 1, dimana nilai -1 mengidentifikasikan kata-kata yang paling negatif seperti kata 'menyedihkan', 'mengerikan', atau 'menjijikkan', dan nilai 1 mengidentifikasikan kata-kata yang paling positif seperti 'luar biasa', 'terbaik' dan lain sebagainya.
- Untuk nilai subjektivitas terletak antara nilai 0 dan 1. Kalimat subyektif umumnya mengacu pada pendapat, emosi, atau penilaian. Nilai subjektivitas menunjukkan julah dari opini pribadi, jika sebuah kalimat memiliki subjektivitas tinggi yaitu mendekati nilai 1, maka dapat dipastikan bahwa kalimat tersebut berisi pendapat pribadi dari pada informasi faktual.

Untuk memulai menggunakan *TextBlob* diperlukan *phyton* yang sudah di *install* dan dikonfigurasi sebelumnya. Paket yang diinstal adalah *TextBlob* dan mendownload korpora NLTK yang diperlukan dalam proses analisis.

```
pip install -U textblob
python -m textblob.download_corpora
```

Gambar 2.2. *Install TextBlob*

Untuk penggunaan *TextBlob* dapat dilakukan sesudah pembersihan dataset atau juga sebelum dilakukan pembersihan data. Hasil *output* dari penggunaan *TextBlob* aka berbeda, sesuai dengan tahapa yang digunakan. Dengan adanya *library TextBlob* maka proses anotasi data dapat dilakukan secara otomatis, sehingga dapat mempersingkat waktu dalam proses anotasi.

Selain anotasi dan analisis sentimen, *TextBlob* juga mempunyai fungsi lainnya, yaitu:

1. Tokenisasi.

   TextBlob dapat memecah sebuah paragraf menjadi kalimat atau kata.

```
>>> zen = TextBlob("Beautiful is better than ugly."
...                "Explicit is better than implicit."
...                "Simple is better than complex.")
>>> zen.words
WordList(['Beautiful', 'is', 'better', 'than', 'ugly', 'Explicit', 'is', 'better', 'than', 'implicit', 'Simple', 'is', 'better',
'than', 'complex'])
>>> zen.sentences
[Sentence("Beautiful is better than ugly."), Sentence("Explicit is better than implicit."), Sentence("Simple
is better than complex.")]
```

Gambar 2.3. Tokenisasi dengan *library TextBlob*

2. Infleksi kata dan Lemmatisasi (*Lemmatization*).

   Setiap kata dalam ***TextBlob*.words** atau **Sentence.words** merupakan objek kata (subkelas dari *unicode*) dengan metode yang bermanfaat dalam infleksi kata.

```
>>> sentence = TextBlob('Use 4 spaces per indentation level.')
>>> sentence.words
WordList(['Use', '4', 'spaces', 'per', 'indentation', 'level'])
>>> sentence.words[2].singularize()
'space'
>>> sentence.words[-1].pluralize()
'levels'
```

Gambar 2.4. Subkelas dari unicode *library TextBlob*

Kata-kata dapat dilemmatisasi dengan memanggil metode lemmatize.

```
>>> from TextBlob import Word
>>> w = Word("octopi")
>>> w.lemmatize()
'octopus'
>>> w = Word("went")
>>> w.lemmatize("v")  # Pass in WordNet part of speech (verb)
'go'
```

Gambar 2.5. Lemmatisasi dengan *library TextBlob*

3. Pengurai teks (*Parse*).

Library TextBlob juga dapat digunakan untuk mengurai teks menjadi beberapa bagian kata.

```
>>> b = TextBlob("And now for something completely different.")
>>> print(b.parse())
And/CC/O/O now/RB/B-ADVP/O for/IN/B-PP/B-PNP something/NN/B-NP/I-PNP completely/RB/B-
ADJP/O different/JJ/I-ADJP/O ./.O/O
```

Gambar 2.6. Lemmatisasi dengan *library TextBlob*

4. Koreksi Ejaan (*Spelling Correction*).

Pada *library TextBlob* terdapat perintah yang dapat mengoreksi ejaan dari sebuah kalimat. Dengan menggunakan perintah *correct* secara otomatis kamus *lexicon TextBlob* dapat mengoreksi ejaan dari kalimat yang dituju. Objek ***word*** memiliki metode yang dapat mengoreksi daftar kalimat dengan saran ejaan.spellcheck() Word.spellcheck()(word, confidence).

```
>>> b = TextBlob("I havv goood speling!")
>>> print(b.correct())
>>> from TextBlob import Word
>>> w = Word('falibility')
>>> w.spellcheck()
```

Gambar 2.7. *Spelling Correction*

2.2.3. *VADER*

VADER merupakan salah satu penganalisis dan penganotasi sentimen berbasis *lexicon* yang memiliki aturan yang telah ditentukan sebelumnya untuk sebuah kata atau leksikon. *VADER* tidak hanya memberi label bahwa leksikon tersebut positif, netral, atau negatif, tetapi juga memberitahu seberapa positif, netral, atau negatif sebua kalimat. *Output* dari *VADER* pada kamus phyton berupa 4 (empat) kunci utama, yaitu '*pos*', '*neu*', '*neg*', dan '*compound*'. *Compound* merupakan skor majemuk yang sangat diperlukan dalam anotasi sebuah kalimat yang didapat dengan menormalkan skor dari '*pos*', '*neu*', '*neg*' dan memberi nilai dari angka dari -1 dan +1. Semakin banyak skor gabungan yang mendekati nilai +1, maka semakin tinggi kepositifan sebuah teks.

```
print(sentiment.polarity_scores("This is a good car"))
{'neg': 0.0, 'neu': 0.508, 'pos': 0.492, 'compound': 0.4404}
```

Gambar 2.8. *Skor Polaritas VADER*

Dari gambar 8 terlihat skor dari kalimat "This is a good car" adalah 49,2% positif, 0% Negatif, 50,8% Netral. Sedangkan nilai *compound* atau skor majemuk adalah 44,04%.

Kriteria keputusan dari *library VADER* mirip dengan *TextBlob* yaitu -1 untuk sebagian besar yang negatif dan +1 untuk sebagian besar yang positif, namun

cara kerja dari *VADER* dan *TextBlob* adalah berbeda. Sama halnya dengan *TextBlob, VADER* merupakan penganalisis dan penganotasi sentimen secara otomatis.

```
pip install vaderSentiment
```

Gambar 2.9. *Install VADER*

Keuntungan dari penggunaan *VADER* dalam analisis sentimen adalah sebagai berikut:

- Tidak memerlukan data pelatihan
- *VADER* dapat memahami sentimen teks dengan sangat baik yang berisikan emotikon, slang, kata sambung, huruf kapital, tanda baca, dan masih banyak lagi.
- *VADER* berfungsi dengan sangat baik pada analisis teks media sosial dan dapat berkeja dengan banyak domain.

2.2.4. Text Preprocessing

Text preprocessing merupakan serangkaian proses bertahap yang bertujuan untuk mengubah bentuk sebuah data yang tidak terstruktur menjadi lebih terstruktur sesuai dengan kebutuhan dalam proses *data mining* dan hasilnya berupa data numerik (Ashari et al., 2020). *Text preprocessing* pada umumnya terbagi dalam beberapa tahapan atau proses diantaranya adalah *cleaning* (penghapusan tanda baca dan simbol), *casefolding*, *tokenizing*, *filtering*, *normalization* dan *stemming*.

1. *Cleaning* (penghapusan tanda baca dan simbol)

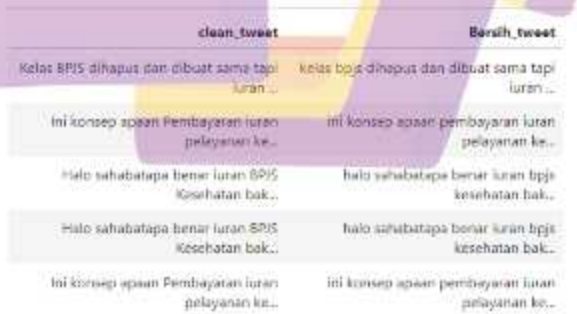
Pada Tahap *Cleaning* atau penghapusan tanda baca dan simbol merupakan proses menghilangkan semua tanda baca dan simbol dari setiap baris agar tidak mengganggu proses *preprocessing* data.

Tweet	clean_tweet
RT @hipohan: Kelas BPJS dihapus dan dibuat sam...	Kelas BPJS dihapus dan dibuat sama tapi iuran ...
RT @msaid_didu: Ini konsep apaan ? \nPembayaran...	Ini konsep apaan Pembayaran iuran pelayanan ke...
RT @BPJSKesehatanRI: Halo sahabat...apa benar iu...	Halo sahabatapa benar iuran BPJS Kesehatan bak...
RT @BPJSKesehatanRI: Halo sahabat...apa benar iu...	Halo sahabatapa benar iuran BPJS Kesehatan bak...
RT @msaid_didu: Ini konsep apaan ? \nPembayaran...	Ini konsep apaan Pembayaran iuran pelayanan ke...

Gambar 2.10. *Cleaning* (penghapusan tanda baca dan simbol)

2. *Casefolding*

Tahap *casefolding* merupakan proses pengubahan semua huruf baik kapital maupun huruf kecil dari setiap baris diseragamkan dalam bentuk huruf kecil. Tujuan dari dilakukan *casefolding* adalah agar data yang digunakan lebih terstruktur.



clean_tweet	Bersih_tweet
Kelas BPJS dihapus dan dibuat sama tapi iuran ...	kelas bpjs dihapus dan dibuat sama tapi iuran ...
Ini konsep apaan Pembayaran iuran pelayanan ke...	ini konsep apaan pembayaran iuran pelayanan ke...
Halo sahabatapa benar iuran BPJS Kesehatan bak...	halo sahabatapa benar iuran bpjs kesehatan bak...
Halo sahabatapa benar iuran BPJS Kesehatan bak...	halo sahabatapa benar iuran bpjs kesehatan bak...
Ini konsep apaan Pembayaran iuran pelayanan ke...	ini konsep apaan pembayaran iuran pelayanan ke...

Gambar 2.11. *Casefolding* (proses penghilangan tanda baca dan simbol)

3. *Tokenizing*

Tahap *tokenizing* merupakan proses pemecahan kalimat-kalimat menjadi kata atau disebut token untuk memudahkan proses analisis data. Dengan tokenizing maka pemisah kata atau bukan dapat dibedakan.

	Bersih_tweet	Tweets
0	kelas bpjs dihapus dan dibuat sama tapi iuran ...	[kelas, bpjs, dihapus, dan, dibuat, sama, tapi...
1	ini konsep apaan pembayaran iuran pelayanan ke...	[ini, konsep, apaan, pembayaran, iuran, pelaya...
2	halo sahabatapa benar iuran bpjs kesehatan bak...	[halo, sahabatapa, benar, iuran, bpjs, kesehat...
3	iuran bpjs kesehatan sesuai besaran gaji mulai...	[iuran, bpjs, kesehatan, sesuai, besaran, gaji...
4	iuran bpjs beda tapi fasilitas sama iwan sumul...	[iuran, bpjs, beda, tapi, fasilitas, sama, iwa...

Gambar 2.12. *Tokenizing*

4. *Filtering*

Tahap Filtering merupakan lanjutan dari tahapan *tokenizing*, dimana *filtering* digunakan untuk menghilangkan kata-kata/istilah-istilah yang masuk dalam kategori *Stop Words*. *Stop Words* merupakan sekumpulan kata-kata lazim yang mempunyai kontribusi minim untuk dijadikan diskriminan efektif sebagai fitur dalam proses klasifikasi.

	Bersih_tweet	Tweets
0	kelas bpjs dihapus dan dibuat sama tapi iuran ...	[kelas, bpjs, dihapus, iuran, berbeda, beda, t...
1	ini konsep apaan pembayaran iuran pelayanan ke...	[konsep, pembayaran, iuran, pelayanan, kesehat...
2	halo sahabatapa benar iuran bpjs kesehatan bak...	[halo, sahabatapa, iuran, bpjs, kesehatan, jut...
3	iuran bpjs kesehatan sesuai besaran gaji mulai...	[iuran, bpjs, kesehatan, sesuai, besaran, gaji...
4	iuran bpjs beda tapi fasilitas sama iwan sumul...	[iuran, bpjs, beda, fasilitas, iwan, sumule, p...

Gambar 2.13. *Filtering*

5. *Normalization*

Normalization atau normalisasi merupakan proses pengembalian kata-kata yang tidak baku kedalam bahasa baku dalam kamus bahasa indonesia.

6. *Stemming*

Tahap *Stemming* adalah tahap dimana kata-kata/istilah-istilah dalam setiap baris teks yang tidak dalam bentuk kata dasar (kata-kata berimbuhan atau dalam bentuk tidak tak beraturan bukan akar) dirubah kedalam bentuk dasarnya (stem).

Gambar 2.14. *Stemming*

Proses ini bekerja dengan menghilangkan imbuhan dan merubah bentuk kata-kata yang tidak beraturan dan tidak dalam bentuk kata dasarnya menjadi kata-kata dalam bentuk kata dasar sebagaimana sesuai dengan kaidah yang terdapat di dalam *lexicon Stemmer*.

2.2.5. Pembobotan TF-IDF

TF-IDF (Term Frequency Inverse Document Frequency) adalah sebuah metode yang digunakan untuk menentukan nilai frekuensi dari sebuah kata didalam sebuah dokumen atau artikel dan juga frekuensi tersebut berada dibanyak dokumen.

Perhitungan ini sebagai penentu seberapa relevan sebuah kata didalam dokumen (Hendy Evan & Sigit Purnomo, 2014). Dalam pengklasifikasian kata, hasil yang diperoleh menggunakan pembobotan TF-IDF lebih baik dan akurat dibandingkan dengan yang tidak menggunakan pembobotan TF-IDF.

Metode ini menggunakan 2 (dua) konsep dasar dalam melakukan perhitungan bobot dari sebuah teks yaitu frekuensi kemunculan sebuah kata didalam sebuah dokumen dan *inverse* dari frekuensi dokumen yang mengandung kata tersebut. Kata yang terkandung didalam sebuah frekuensi dokumen menunjukkan seberapa umum kata tersebut. Nilai bobot diantara sbuah kata dan dokumen akan menjadi tinggi jika frekuensi kata tersebut tinggi didalam dokumen dan didalam keseluruhan frekuensi dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen (dataset) (Robertson, 2004).

Rumus pembobotan TF-IDF ditulisakan oleh (Rolly Intan & Andrew Defeng, 2006) sebagai berikut:

$$wtd = tf_{td} * idf \quad (1)$$

$$wtd = tf_{td} * log\left(\frac{N}{df_t}\right) \quad (2)$$

Keterangan:

w_{td} = bobot kata/token t_t terhadap dokuemen d_d

tf_{td} = jumlah kemunculan kata/token t_t dalam dokumen d_d

N = jumlah semua dokumen dalam database

df_t = jumlah dokumen yang mengandung kata/token t_t

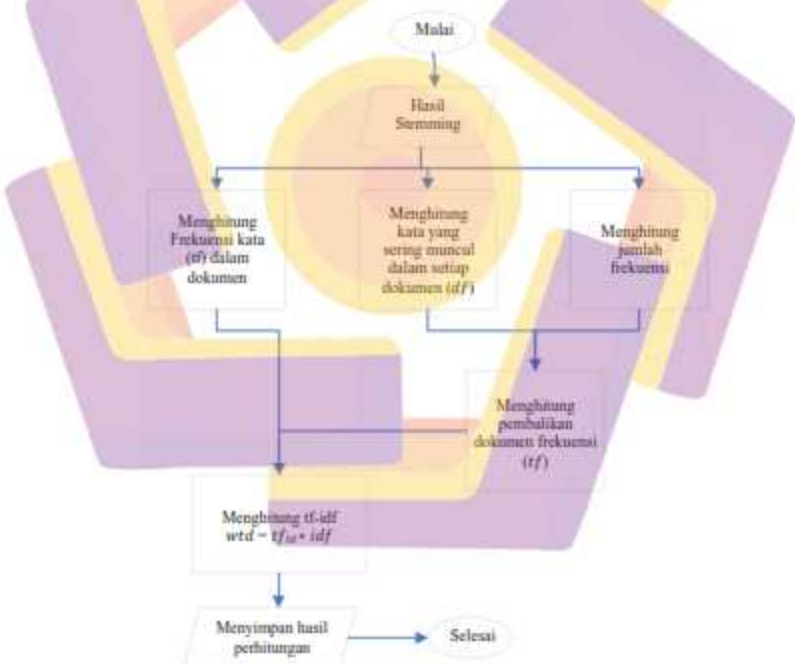
Dari rumus (7.5.2) dapat ditarik kesimpulan bahwa berapapun besarnya nilai tf_{td} dan bila $N = df_t$ dari sebuah kata muncul disetiap dokumen, maka untuk

perhitungan idf hasilnya adalah 0 (nol), oleh sebab itu perhitungan bobotnya berubah seperti berikut:

$$wtd = tf_{td} * (\log(\frac{N}{df_t}) + 1 \quad (3)$$

Rumus (4) merupakan normalisasi dari rumus (3) yang bertujuan untuk menstandarisasi nilai bobot (w_{td}) kedalam interval 0 s.d. 1.

$$wtd = \frac{wtd = tftd * \log(\frac{N}{df_t})}{\sqrt{\sum_{k=1}^{t}(tf_{td})^2 * [(\log(\frac{N}{df_t})+1)]^2}} \quad (4)$$



Gambar 2.15. Flowchart TF-IDF (Prihatini, 2016)

Pada gambar 2.15 mengilustrasikan tahapan pembobotan kata dengan menggunakan metode *term frequency-inverse document frequency* (TFIDF),

dimana daftar kata yang telah di *stemming* dihitung untuk menentukan bobot dari setiap kata dengan menghitung jumlah frekuensi kata dalam dokumen (*tf*) terlebih dahulu, kemudian menghitung nilai jumlah dokumen yang memiliki kata tersebut (df) dengan rumus *log* = *N* / df. Setelah nilai *tf* dan *i*df didapatkan, langkah terakhir adalah menentukan bobot kata dengan mengalikan TF dan IDF dengan rumus $wtd = tf_{td} * idf$ lalu Hasil dari proses perhitungan disimpan didalam database.

2.2.6. *Naive Bayes*

Naive Bayes merupakan algoritma *supervised learning* yang digunakan untuk proses klasifikasi melalui pendekatan probabilistik. Penemu teori *Naive Bayes* adalah seorang ilmuwan inggris yang bernama Thomas Bayes, dimana teori tersebut adalah memprediksi peluang dimasa depan berdasarkan pengalaman sebelumnya sehingga dikenal sebagai Teorema Bayes. Metode ini sangat luas dipakai dalam berbagai bidang, khususnya dalam proses klasifikasi dokumen (Pratiwi & Widodo, 2017).

Persamaan umum dari prediksi bayes pada theorema bayes adalah sebagai berikut:

$$P(Y|X) = \frac{P(X|Y)*P(Y)}{P(X)} \quad (5)$$

Keterangan:

X : Data dengan class yang belum diketahui.

Y : Hipotesis data X merupakan suatu kelas spesifik.

P(Y|X) : Probabilitas akhir bersyarat (conditional probability) atau suatu hipotesis Y terjadi jika diberikan bukti X (evidence) terjadi.

P(X|Y) : Probabilitas sebuah bukti X akan mempengaruhi hipotesis.

Y P(Y) : Probabilitas awal (priori) hipotesis y terjadi tanpa memandang bukti apapun.

P(X) : Probabilitas awal (priori) bukti X terjadi tanpa memandang bukti atau hipotesis yang lain.

Konsep dasar dari aturan *Bayes* adalah hasil dari hipotesis atau peristiwa (H) yang dapat diperkirakan berdasarkan bukti (E) yang diamati. Hal yang harus diperhatikan dalam aturan *bayes*, yaitu sebagai berikut:

1. Sebuah probabilitas awal atau priosi H atau P(H) adalah probabilitas dari suatu hipotesis sebelum bukti diamati.
2. Sebuah probabilitas akhir H atau P(H|E) adalah probabilitas dari suatu hipotesis setelah bukti diamati.

Naive Bayes dapat dilatih dalam *supervised learning* bergantung pada model probabilitas. Model *Naive Bayes* dapat dikerjakan tanpa mempercayai probabilitas Bayesian atau menggunakan metode Bayesian yang lain. Kelebihan dari Naïve Bayes adalah dalam penggunaanya hanya membutuhkan sedikit data latih dalam penentuan estimasi parameter yang diperlukan untuk proses klasifikasi. Karena diasumsikan sebagai variabel independen, hanya varian dari variabel dalam suatu kelas yang diperlukan untuk menentukan pengklasifikasian, bukan seluruh dari matriks kovarians. Hal ini menjadikan model Naive Bayes sangat efisien dalam pengolahan data berskala besar, terutama pada data yang memiliki banyak fitur namun dengan ukuran sampel yang relatif kecil, sehingga mampu memberikan hasil klasifikasi yang cepat dan akurat.

2.2.7. *Naive Bayes* Classifier

Naive Bayes dan klasifikasi mempunyai hubungan dalam korelasi hipotesis dan bukti dengan klasifikasi, artinya hipotesis dalam teorema Bayes adalah label kelas yang menjadi target pemetaan pada klasifikasi, sedangkan bukti merupakan fitur-fitur yang menjadi masukkan dalam model klasifikasi. Jika X adalah vektor masukkan yang berisi fitur dan Y adalah label kelas, maka simbol *Naïve Bayes* dituliskan sebagai P(Y|X), dimana notasi tersebut berarti bahwa probabilitas label kelas Y didapatkan setelah mengamati fitur-fitur X. Notasi ini disebut juga probabilitas akhir (posterior probability) untuk Y, sedangkan P(Y) disebut probabilitas awal (prior probability) Y. Formula atau rumus untuk *Naive Bayes Classifier* adalah sebagai berikut:

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{q} P(X_i|Y)}{\mathrm{P(X)}} \qquad (6)$$

Keterangan:

P(X|Y) = Probabilitas data dengan vektor X pada kelas Y.

P(Y) = Prrobabilitas kelas awal Y.

$\prod_{i=1}^{q} P(X_i|Y)$ = Probabilitas independen kelas Y dari semua fitur dalam vektor X.

Nilai P(X) adalah nilai tetap sehingga dalam perhitungan prediksi hanya menghitung bagian yang terbesar sebagai kelas yang dipilih sebagai hasil prediksi. Sementara probabilitas independen tersebut merupakan pengaruh semua fitur dari data terhadap setiap kelas Y, yang dinotasikan dengan :

$$P(Y|X = y) = \prod_{i=1}^{q} P(X_i|Y) = y \qquad (7)$$

Setiap set fitur X = {X_1,X_2,X_3,..., Xq} terdiri atas q atribut (q dimensi). Bayes merupakan metode yang mudah untuk menghitung fitur bertipe kategoris

seperti klasifikasi, tetapi untuk fitur numerik (kontinyu) ada perlakuan khusus sebelum dimasukkan pada Naïve Bayes, yaitu:

1. Nilai dari setiap fitur kontinyu harus diganti dengan nilai interval diskret denganc cara mentransformasikan fitur kontinu ke dalam fitur ordinal.
2. Menggunakan distribusi Gaussian untuk mempresentasikan probabilitas bersyarat dari fitur kontinyu pada sebuah kelas.

Sementara itu rumus yang digunakan dalam Naïve Bayes klasifikasi teks dengan pembobotan kata TF untuk menghitung prababilitas kata yang akan muncul pada salah satu kelas adalah sebagai berikut:

$$P(w_k|c_i) = \frac{(n_{wk})c_i+1}{n_{ci}+n_k} \quad (8)$$

Keterangan:

$P(w_k|c_i)$ = Peluang kemunculan pada kategori c_i

Wk = Kata pada data latih yang dicari peluang kemunculannya

c_i = Kategori yang ada pada data

$(n_{wk})c_i$ = Jumlah kemunculan kata pada kelas/kategori

n_{ci} = Total kata yang ada pada kelas

n_k = Total kata yang ada pada data latih

Pada saat tahap pengujian apabila terdapat kata yang belum muncul pada data latih maka nilai = 1.

2.2.8. Support Vector Machine (SVM)

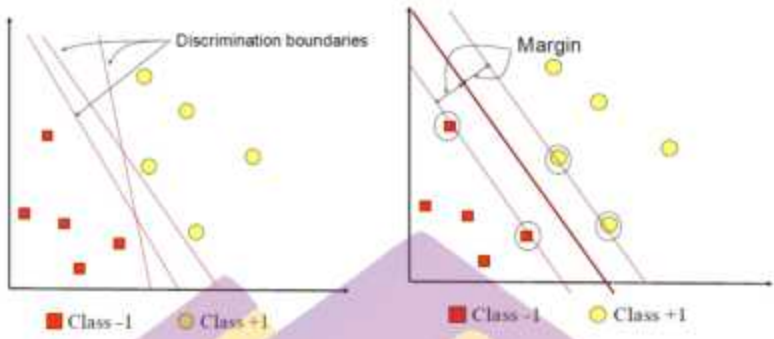
Support vector machine (SVM) merupakan sebuah sistem pembelajaran yang pertama kali diperkenalkan oleh Vapnik pada tahun 1992, dimana teori SVM

adalah menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (feature space) berdimensi tinggi yang di latih dengan algoritma pembelajaran berdasarkan teori optimasi dengan mengimplementasikan learning bias yang berasal dari teori statistik. Secara konsptual, SVM merupakan mesin linear yang di bekali degan fitur-fitur khusus dan berdasarkan *metode structural risk-minimization* (SRM) dan pembelajaran teori statistik. Sehingga SVM dapat memberikan kinerja generalisasi yang baik dalam masalah pengenalan pola. Metode klasifikasi SVM merupakan metode yang diskriminatif yang sangat tepat digunakan dalam proses klasifikasi (Nugroho et al., 2003).

Berbeda dengan metode lainya, jika selama proses pelatihan semua data latih akan dipelajari dan dilibatkan dalam setiap iterasi, maka pada SVM tidak semuya data latih dipandang untuk dilibatkan pada setiap iterasi pelatihannya. Setiap data yang mempunyai kontribusi dalam pelatihan disebut dengan *support vector*, sehingga metode ini disebut dengan *Support Vector Machine* (SVM).

2.2.8.1 Konsep dasar Support Vector Machine (SVM)

Konsep dasar dari SVM adalah memaksimalkan batas *hyperplane* (*maximal margin hyperplane*) seperti yang diilustrasikan pada gambar .... Pada gambar .... (a) terlihat sejumlah pilihan *hyperplane* yang dapat digunakan menjadi set data, sedangkan pada gambar 16 (b) merupakan *hyperplane* dengan nilai margin paling maksimal. Meskipun Gambar 16 (a) dapat menggunakan *hyperplane* sembarang, namun *hyperplane* yang mempunyai nilai margin maksimal yang mampu memberikan generalisasi yang lebih baik pada metode klasifikasi.



a. Decision boundary yang mungkin

b. Decision boundary dengan margin maksimal

Gambar 2.16. Ilustrasi SVM menemukan hyperline terbaik untuk memisahkan kelas (Nugroho et al., 2003)

Konsep dasar dari SVM merupakan usaha dalam mencari *hyperlane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada ruang input. Pada gambar ... memperlihatkan beberapa pola yang merupakan anggota dari 2 (dua) kelas atau kelompok data yaitu kelas -1 dan kelas +1. Pada kelas -1 setiap data disimbolkan dengan bujur sangkar, sedangkan simbol kelas -1 berbentuk lingkaran. *Hyperplane* (batas keputusan) terbaik dapat ditemukan dengan melakukan pengukuran *margin* *hyperplane* tersebut dengan mencari titik maksimalnya. *Margin* merupakan jarak antara *hyperplane* dengan data terdekat dari masing-masing kelas dan data yang terdekat disebut sebagai *support vector*. Pada gambar... (b) garis tebal menunjukkan *hyperplane* terbaik karena berada ditengah kedua kelas, sedangkan 2 (dua) garis lainnya yang lebih tipis merupakan *support vector*. Pencarian lokasi *hyperplane* merupakan proses pelatihan utama dari SVM.

2.2.8.2 Support Vector Machine (SVM) Linier

Data latih pada SVM dinyatakan oleh (x_i, y_i), dimana $i = 1, 2, ..., N$, dan $x_i = \{ x_{i1}, x_{i2}, ..., x^{iq} \}^T$ merupakan atribut (fitur) untuk data latih ke-i, sedangkan $y_i \in \{-1, +1\}$ menyatakan label kelas. Persamaan *hyperplane* klasifikasi SVM linier dianotasikan dengan persamaan (9) dibawah ini.

$$w * x_i + b = 0 \quad (9)$$

w dan b merupakan parameter model. $w * x_i$ merupakan *inner-product* antara w dan x_i.

Data x_i yang masuk kedalam kelas -1 merupakan data yang memenuhi pertidak samaan, sehingga dianotasikan dengan pertidaksamaan (10) dibawah ini.

$$w * x_i + b \leq -1 \quad (10)$$

Sedangkan data x_i yang masuk kedalam persamaan +1 merupakan data yang memenuhi pertidaksamaan, sehingga dianotasikan dengan pertidaksamaan (11) dibawah ini.

$$w * x_i + b \geq +1 \quad (11)$$

Terlihat pada gambar ....., jika data kelas -1 bertempat di *hyperplane* (x_a), maka persamaan 1 terpenuhi. Untuk persamaan data kelas -1 dianotasikan pada persamaan (7.7.4) dibawah ini.

$$w * x_a + b = 0 \quad (12)$$

Sedangkan untuk data kelas +1 (xb) akan memenuhi persamaan (13) dibawah ini.

$$w * x_b + b = 0 \quad (13)$$

Dengan mengurangi persamaan (13) dengan (12), maka didapatkan perasamaan (14) dibawah ini.

$$w * (x_b - x_a) = 0 \quad (14)$$

$x_b - x_a$ merupakan vector paralel pada *hyperplane* yang diarahkan dari x_a ke x_b. Karena *inner-product* bernilai nol, maka arah w harus tegak lurus terhadap *hyperplane*. Dengan memberikan label -1 untuk kelas pertama dan +1 untuk kelas kedua, maka prediksi semua data uji akan menggunakan Persamaan (15) dibawah ini.

$$y = \begin{cases} -1, jika\ w.z + b > 0 \\ +1, jika\ w.z + b > 0 \end{cases} \quad (15)$$

Dilihat dari gambar .... *hyperplane* kelas -1 (garis tipis) adalah data *support vector* yang memenuhi persamaan (16).

$$w * x_a + b = -1 \quad (16)$$

Sementara *hyperplane* kelas +1 (garis tipis) memenuhi pada Persamaan (17).

$$w * x_a + b = +1 \quad (17)$$

Dengan demikian, nilai margin dapat dicari dengan mengurangi persamaan 9 dan 8, sehingga mendapatkan persamaan (18).

$$w * (x_b + x_a) = 2 \quad (18)$$

Margin *hyperplane* diberikan oleh jarak antara dua hyperplane dari dua kelas tersebut. Notasi diatas diringkas menjadi Persamaan (19).

$$|w| * d = 2 \text{ atau } d = 2 / |w| \quad (19)$$

2.2.8.3 Support Vector Machine (SVM) Nonlinier

Pada dasarnya SVM merupakan *hyperplane* linier yang bekerja hanya pada data yang dapat dipisahkan secara linier, sehingga data yang tidak dapat dipisahkan secara linier biasanya menggunakan kernel pada fitur awal set data. Kernel merupakan fungsi pemetaan data dari dimensi awal ke dimensi yang lebih tinggi. Pendekatan ini berbeda dengan metode klasifikasi yang mengurangi dimensi awal dalam menyederhanakan proses komputasi guna mendapatkan akurasi prediksi yang lebih baik. Algoritma pemetaan kernel ditunjukkan pada persamaan (21).

$$\Phi: D\ q \rightarrow D\ r\ ,\ x \rightarrow \Phi(x) \qquad (21)$$

Φ merupakan fungsi kernel pada pemetaan, D merupakan data latih, q merupakan set fitur lama, dan r merupakan set fitur baru dari hasil pemetaan untuk data latih. Sementara x merupakan bagian dari data latih, dimana $x_1, x_2, \ldots x_n \in D$ q merupakan fitur-fitur yang akan dipetakan ke fitur berdimensi tinggi r, sedangkan set data pelatihan dengan algoritma yang ada berasal dari dimensi fitur yang lama D ke dimensi baru r, dam n merupakan sampel data.

$$(\Phi(x_1), y_1, \Phi(x_2), y_2, \ldots \Phi(x_1), y_n) \in D\ r \qquad (22)$$

Pemetaan fitur lama pada set data ke fitur baru merupakan analogi dari layer tersembunyi pada ANN dimana jumlah neuron dalam layer tersembunyi biasanya lebih banyak dari pada jumlah vektor masukan. Pada fase ini Proses pemetaan perhitungan *dot-product* diperlukan pada dua buah data dari ruang fitur baru. *Dot-product* vektor (x_i) dan (x_j) dinotasikan sebagai $\Phi(x_i)$. $\Phi(x_j)$. Nilai dot-product dari kedua vektor dapat langsung dihitung tanpa melakukan dan mengetahui transformasi Φ. Teknik komputasi ini disebut sebagai trik kernel. Trik kernel

menghitung *dot-product* dua buah vektor dari ruang dimensi baru menggunakan kedua vektor tersebut dari ruang asal seperti berikut ini:

$$K(x_i , x_j) = \Phi(x_i). \Phi(x_j) \quad (23)$$

Sedangkan untuk prediksi set data dari untuk dimensi fitur baru diformulasikan dengan Persamaan 24.

$$f(z) = \sum_{i=1}^{n} a_i y_i K(x_i, z) + b \quad (24)$$

N merupakan jumlah data *support vector* data, *xi* adalah *support vector*, dan z adalah data uji yang akan diprediksi kelasnya.

Beberapa kernel yang terdapat pada svm, antara lain:

a. Kernel linier

$$K(xi , x) = x. xi\ T \quad (25)$$

b. Polynomial

$$K(xi , x) = (Y\ xi\ T\ x + r)\ p\ , Y > 0 \quad (26)$$

c. Radial basis function (RBF)

$$K(xi , x) = \exp(-Y|xi - x|\ 2\ ), Y > 0 \quad (27)$$

d. Sigmoid kernel *K*

$$(xi , x) = tanh(Y\ xi\ T\ x + r) \quad (28)$$

2.3. Keaslian Penelitian

ANALISA PERBANDINGAN PENGARUH *TEXTBLOB* DAN *VADER* TERHADAP ANALISIS SENTIMEN MENGGUNAKAN METODE NAÏVE BAYES DAN SVM

Tabel 2.1. Matriks literatur review dan posisi penelitian

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Comparative Evaluation of Lexicons in Performing Sentiment Analysis	Wan Nur Syahirah Wan Min & Nur Zareen Zulkarnain, Journal of Advanced Computing Technology and Application, 2020.	Menganalisa kinerja anotasi otomatis berbasis *lexiconi* yaitu *TextBlob* dan *VADER* dalam melakukan analisis sentimen terhadap data teks yang berasal dari *twitter*.	Dari penelitian tersebut ditemukan bahwa kedua *lexicon* menghasilkan tingkat akurasi yang masih dapat diterima yaitu 79% untuk *VADER* dan 73% untuk *TextBlob*.	Peneliti: 1. Tidak menggunakan metode atau algoritma lainnya dalam membuat model mesin pembelajaran untuk prediksi kelas. 2. Dataset yang digunakan adalah data random dari *tweet* dan tidak berasal dari *hastag* yang sama.	Tindak lanjut: 1. Membangun model klasifikasi menggunakan algoritma lainnya setelah dataset dianotasi menggunakan *TextBlob* dan *VADER*. 2. Menggunakan dataset yang berasal dari *hastag* yang sama, sehingga ada perbandingan antara
2	Evaluating the performance of the most important Lexicons used to	M. A. Al-Shabi, International Journal of Computer	Mengevaluasi kinerja *library lexicon* (*VADER, SentiWordNet, SentiStrength, Liu*	Hasil dari evaluasi yang telah di lakukan menghasilkan akurasi terhadap masing-masing *lexicons* adalah *VADER*	Peneliti: 1. Akurasi yang dihasilkan masih belum memenuhi standar kelayakan	Tindak lanjut: 1. Melakukan *Cleaning* data agar data menjadi terstruktur.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
	Sentiment analysis and opinions Mining	Science and Network Security (IJCSNS), 2020.	*and Hu opinion* dan *AFINN-111)* yang mempunyai pengaruh paling baik pada analisis sentimen publik terhadap permasalahan sistem pelacakan aplikasi JIRA.	sebesar 72%, *SentiWordNet 53%, SentiStrength* sebesar 67%, *Liu and Hu opinion* sebesar 65% dan *AFINN-111* sebesar 65%, sehingga dari kelima *library* yang paling baik kinerjanya adalah *VADER*.	dalam klasifikasi analisis sentimen. 2. Penerapan *lexicon* pada dataset tanpa melalui proses *preprocessing* terlebih dahulu. 3. Hasil akurasi yang diperoleh dari anotasi manual tidak ditampilkan.	2. Melakukan tahapan *preprocessing* pada data dataset sebelum penerapan *lexicon* dilakukan. 3. Melakukan percobaan dengan membangun modek mesin pembelajaran menggunakan algoritma lainnya setelah data dianotasi menggunakan *lexicon*. 4. Menganotasi data secara manual dan membandingkan hasil analisis dengan hasil penelitian sebelumnya.
3	Metode Naive Bayes Classifier dengan *TextBlob* untuk Anlisis Sentimen Terhadap Pelayanan Indihome dan First Media	Navi Atri Lestari, Tubagus Mohammad Akhriza dan Eka Yuniar, Seminar Nasional Teknologi Informasi dan	Menganalisa dan mengetahui sentimen publik mengenai layanan yang diberikan oleh 2 (dua) vendor Internet Service Provider (ISP) di Indonesia yaitu indihome dan First	Dari dataset yang digunakan yaitu sebanyak 193 baris *tweet* untuk indihome dan 195 baris *tweet* untuk first media dan diuji menggunakan algoritma klasifikasi *Naive Bayes Classification* (NBC)	Peneliti: 1. Hanya menggunakan 1 (satu) algoritma klasifikasi, sehingga tidak ada data pembanding. 2. Dengan menggunakan 2 (dua) anotasi sentimen yaitu positif dan	Tindak lanjut: 1. Menambah dataset agar hasil yang diperoleh lebih baik dan lebih variatif. 2. Menambahkan algoritma klasifikasi dalam pembuatan model mesin pembelajaran sebagai pembanding dengan hasil sebelumnya.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Komunikasi STI&K (SeNTIK), 2020.	Media dengan memanfaatkan data *tweet* dari *twitter* yang dianotasi secara otomatis menggunakan *library TextBlob* dan algoritma yang digunakan dalam klasifikasi adalah *Naive Bayes Classification* (NBC).	menghasilkan respon negatif sebanyak 135 respon dan respon positif sebanyak 58 respon untuk indihome, sedangkan first media memperoleh 54 respon positif dan 141 respon negative. Akurasi yang diperoleh sebesar 74%, *recall* 66% dan presisi 83%.	negatif, scharusnya tingkat akurasi lebih tinggi dari yang dihasilkan. 3. Dataset yang digunakan untuk setiap kata kunci sangat sedikit, sehingga akurasi yang dihasilkan masih belum optimal.	3. Proses anotasi sebaiknya diperluas menjadi 3 (tiga) sentimen, yaitu positif, negatif dan netral.
4	Analisis Sentimen Mengenai Vaksin Sinovac Menggunakan Algoritma Support Vector Machine (SVM) dan K-Nearest NEighbor (KNN)	Anna Baita, Yoga Pristyanto, Nuri Cahyono, Information System Journal (INFOS), 2021.	Menganalisis sentimen dari opini masyarakat khususnya pengguna *twitter* tentang pemberian Vaksin *Sinovac* dengan proses anotasi data menggunakan *TextBlob* sebagai penganatosi otomatis dan algoritama pengklasifikasi	Dari model yang telah dibuat menggunakan 2 (dua) algoritma yang berbeda menghasilkan akurasi sebesar 70% menggunakan kernel liner, 57% untuk kernel polynomial, dan 66% untuk RBF dari metode SVM dan untuk akurasi algoritma KNN sebesar 56.	Peneliti: 1. Akurasi yang dihasilkan masih kurang optimal, sehingga model mesin pengklasifikasi yang dibangun belum dapat digunakan sebagai model dalam analisis sentimen. 2. Hasil akurasi dari setiap kernel yang digunakan dari algoritma SVM masih menghasilkan	Tindak lanjut: 1. Perlu dilakukan analisa ulang terhadap setiap kernel yang digunakan pada algoritma SVM dengan menggunakan data latih yang lebih banyak. 2. Melakukan analisa menggunakan algoritma lainnya sebagai pembanding.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			Support Vector Machine (SVM) dan *K-Nearest Neighbor* (KNN).		akurasi yang sangat rendah.	
5	Twitter Sentiment Analysis as an Evaluation and Service Base On Python *TextBlob*	I Gede Susrama Mas Diyasa, Ni Made Ika Marini Mandenni, Mohammad Idham Fachrurrozi, Sunu Ilham Pradika, Kholilul Rachman Nur Manab, & Nyoman Rahadi Sasmita, IOP Conference Series: Materials Science and Engineering, 2021.	Mengklasifikasi opini masyarakat khususnya pengguna jasa PT. Telkom indonesia terhadap yang ditawarkan yaitu diantaranya indihome, myindihome, useetv, dan wifi.id menggunakan *library TextBlob* sebagai penganotasi dan pengklasifikasi data.	Hasil dari analisa yang dilakukan dari 3324 *tweet* menghasilkan 1590 data netral diantaranya 1266 data valid dan 324 invalid, lalu 1082 data positif diantaranya 858 valid dan 224 invalid, selain itu 518 negatif diantarnya 443 valid dan 75 invalid. Dari hasil tersebut maka prosentasi hasil analisis adalah 34,4% *tweet* positif, 16,1% *tweet* negatif, dan 49,6% *tweet* netral dengan akurasi analisis sebesar 77,2%.	Peneliti: 1. Anotasi dan klasifikasi data dilakukan sebelum melalui proses *cleaning* data. 2. Terlalu banyak data yang invalid dari setia sentimen. 3. Akurasi analisis masih perlu ditingkatkan. 4. Dataset yang digunakan tidak balance antara jumlah sentimen positif, negatif dan netral.	Tindak lanjut: 1. Menganotasi data dengan beberapa tahapan yang berbeda agar mendapatkan akurasi anotasi terbaik dan mengurangi data invalid. 2. Menganalisis dengan menggunakan algoritma klasifikasi sebagai pembanding. 3. Menggunakan dataset yang balance agar model yang dibagun saat pembelajaran mesin lebih akurat.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Sentiement Analysis With a *TextBlob* Package Implications for Tourism	Pannee Suanpang, Pitchaya Jamjuntr & Phuripoi Kaewyong, Journal of Management Information and Decision Sciences, 2021.	Menganalisis sentimen dari ulasan turis menggunakan *library TextBlob* sebagai penganotasi otomatis dan algoritma *Naive bayes* sebagai pengklasifikasi pada bisnis pariwisata dengan studi kasus merangsang ekonimi pariwisata pasca COVID-19 di thailand.	Hasil analisis yang diperoleh dari model yang telah dibangun pada saat penelitian menghasilkan nilai akurasi sebesar 89,32%.	Peneliti: 1. Hasil akurasi yang diperoleh sudah sangat baik, namun akan berbeda jika dataset yang dianotasi tidak diberikan *rating* terlebih dahulu. 2. Perlu dilakukan perbandingan dengan menggunakan algoritma lain, sehingga dapat diketahui seberapa besar pengaruh algoritma *Naive Bayes* dibandingkan dengan algoritma lainnya.	Tindak lanjut: 1. Menguji model menggunakan dataset yang belum diberi *rating* terlebih dahulu. 2. Membangun model baru menggunakan algoritma klasifikasi lainnya dan membandingkan hasil akurasi dengan algoritma sebelumnya.
7	Sentiment Analysis as Assessment of the COVID-19 Social Assistance Pollemic using Random Forest Algorithm	Mohamad Irfan, Pramadita Sielda Dewi, Wildan Budiawan Zulfikar, Cepy Slamet	Menganalisis opini publik terhadap polemik bantuan sosial COVID-19 dengan menggunakan *library TextBlob* untuk melabelkan	Hasil dari penelitian tersebut dengan mengevaluasi algoritma menggunakan metode 10 *k-fold cross validation* sebagai validasi performa dari	1. Hasil akurasi sudah baik, namun masih belum memenuhi standar kelayakan dalam analisis sentimen. 2. perbandingan dengan menggunakan	Tindak lanjut: 1. Membangun model baru menggunakan algoritma klasifikasi lainnya dan membandingkan hasil akurasi dengan algoritma sebelumnya.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Ichsan Taufik, Proceeding of 2022 8th International Conference on Wireless and Telematics (ICWT), 2022.	data teks secara otomatis dan algoritma *Random Forest* untuk memodelkan mesin klasifikasi.	hasil *testingi* diperoleh akurasi sebesar 76%.	algoritma lain, sehingga dapat diketahui seberapa besar pengaruh algoritma *Random Forest* dibandingkan dengan algoritma lainnya.	
8	Sentimental Analysis Applied on Movie Reviews	SichangSu, Journal of Education, Humanities and Social Sciences, 2022.	Menganalisis opini publik terhadap 2 (dua) film yaitu *The Return of the King* and *The Lord of the Rings* dengan menggunakan *library TextBlob* sebagai penganotasi manual dan menggunakan Document Frequency (TF-IDF) untuk mengevaluasi setiap kata dalam ulasan, serta menggunakan algoritma *Support*	Penelitian tersebut menghasilkan nilai akurasi sebesar 85,2% sedikit lebih tinggi dibandingkan penelitian sebelumnya, namun yang menjadi perhatian pada penelitian ini adalah setiap ulasan yang ada telah diberikan skor terlebih dahulu dan berbeda dengan penelitian lainnya yang dataset nya belum diberikan nilai.	Peneliti: 1. Akurasi yang dihasilkan sudah baik dan perlu dilakukan percobaan menggunakan dataset yang masih belum diberikan nilai *rating* untuk mengetahui tingkat akurasi yang diperoleh jika data tersebut belum diberikan *rating* ulasan. 2. Perlu dilakukan perbandingan dengan menggunakan algoritma lain,	Tindak lanjut: 1. Menguji model menggunakan dataset yang belum diberi *rating* terlebih dahulu. 2. Membangun model baru menggunakan algoritma klasifikasi lainnya dan membandingkan hasil akurasi dengan algoritma sebelumnya.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			Vector Machine (SVM) sebagai algoritma pengklasifikasi.		sehingga dapat diketahui seberapa besar pengaruh algoritma *Support Vector Machine* (SVM) dibandingkan dengan algoritma lainnya.	
9	Automatic Annotation Performance of *TextBlob* and *VADER* on Covid Vaccination Dataset	Badriya Murdhi Alenzi, Muhammad Badruddin Khan, Mozaherul Hoque Abul Hasanat, Abdul Khader Jilani Saudagar, Mohammed AlKhathami & Abdullah AlTameem, Intelligent Automation and Soft	Menganalisis dan memvalidasi kinerja dari 2 (dua) anotasi otomatis berbasis *lexicon* yang banyak digunakan yaitu *TextBlob* dan Valence Aware Dictionary and sEntiment Reasoner (*VADER*) dengan membandingkan kinerja kedua *library* dengan anotasi yang dilakukan secara manual	Tingkat akurasi tertinggi yang diperoleh dari serangkaian percobaan yang telah dilakukan adalah 75% untuk *TextBlob* dan 70% untuk *VADER*.	Peneliti: 1. Proses anotasi data menggunakan *library TextBlob* dianggap sebagai hasil akhir dalam klasifikasi. 2. Nilai akurasi masih belum optimal dan perlu ditingkatkan menggunakan algoritma pengklasifikasi.	Tindak lanjut: Melakukan klasifikasi ulang menggunakan algoritma pengklasifikasi, sehingga memperoleh hasil yang optimal dan model mesin pengklasifikasi dapat digunakan untuk mengklasifikasi dataset berikutnya.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
		Computing, 2022.				
10	Twitter Sentiment Analysis Towards Qatar as Host of the 2022 World Cup Using *TextBlob*	Syarafina Dewi & Dede Brahma Arianto, Journal of Social Research, 2023	Menganalisis *tweet* pada *twitter* menggunakan *library TextBlob* sebagai penganotasi dan pengkalisfikasi dan menggunakan dataset yang diperoleh dengan 3 (tiga) tahapan pengumpulan data yaitu *scraping* pertama dilakukan dari tanggal 1 maret 2010 sampai 1 desember 2010 menghasilkan 3606 baris data *tweet*, lalu tahap kedua dari tanggal 2 desember 2010 sampai 19 november 2022 menghasilkan 41225 data *tweet*, dan tahap ketiga dari tanggal 20	Hasil analisis dari tahap pertama sebelum Qatar menjadi tuan rumah adalah sentimen sentimen positif sebesar 88,46% dan negatif sebesar 11,54%, lalu hasil pada tahap kedua setelah Qatar terpilih menjadi tuan rumah terdapat sentimen positif 79,38% dan sentimen negatif 20,62%, dan tahap ketiga saat piala dunia 2022 berlangsung di Qatar terdapat sentimen positif 83,72% dan sentimen negatif 16,28% dengan tingkat akurasi klasifikasi sentimen sebesar 83%.	Peneliti: 1. Penelitian tidak menggunakan algoritma pengklasifiksai, namun menggunakan *library TextBlob* sebagai pengklasifikasi.	Tindak lanjut: Perlu dilakukan penelitian selanjutnya menggunakan algoritma pengklasifikasi guna memvalidasi akurasi anotasi sentimen yang dilakukan menggunakan *library TextBlob*.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			november 2022 sampai dengan 18 desember 2022 menghasilkan 200.000 data *tweet* dengan total keseluruhan sebanyak 244.832 data *tweet*.			

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat dan Pendekatan Penelitian

Tahapan ini bertujuan untuk melakukan pencarian sumber rujukan yang berhubungan dengan penelitian yang hendak dilakukan, berkaitan dengan penggunaan *library lexicon* yaitu *TextBlob* dan *VADER* menggunakan algoritma *Naive Bayes* dan *Support Vector Machine* (SVM). Disamping itu penting untuk mencari sumber rujukan mengenai pemodelan klasifikasi menggunakan algoritma *Naive Bayes* dan *Support Vector Machine* (SVM). Studi pustaka bertujuan untuk memberikan gambaran dan informasi terhadap pembangunan model klasifikasi teks. Dengan metode penelitian maka dapat diperoleh berbagai macam sumber data yang merupakan hasil dari sumber rujukan yang telah tercatat sebelumnya.

3.2. Metode Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah dataset yang di*crawling* dan di *scraping* dari *twitter* dengan kata kunci 'kurikulum merdeka' dan 'Penerapan kurikulum merdeka sebagai kurikulum nasional tahun 2024'.

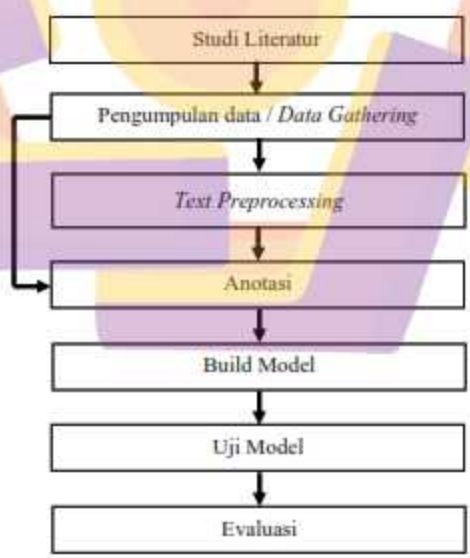
3.3. Metode Analisi Data

Library TextBlob dan *VADER* digunakan dalam proses analisa awal dan anotasi data secara otomatis, sedangkan algoritma *Naive Bayes dan SVM* digunakan

sebagai metode klasifikasi dalam pembangunan model mesin pembelajaran untuk memprediksi hasil sentimen. *Platform* eksperimen yang akan digunakan pada penelitian ini adalah *Google Colab*. *Preprocessing* yang dilakukan pada dataset berupa penghilangan tanda baca, *casefolding*, *tokenizing*, *filtering* dan *stemming*.

3.4. Alur Penelitian

Penelitian ini terdiri dari beberapa tahapan yang dilakukan, diantaranya yaitu: Studi literatur, Pengumpulan data / *Data Gathering*, *Text Preprocessing* (*Cleaning data, Spelling Correction, Drop duplicate data*), anotasi data, Pemrosesan Data dan *build model*, Pengujian Model, Evaluasi, dan penarikan kesimpulan jika hasil dari eksperimen-eksperimen sudah ada yang sesuai dengan target penelitian.



Gambar 3.1. Alur Penelitian

1. Studi literatur.

   Tahapan ini merupakan tahapan dalam memahami dan menerapkan format laporan yang tepat, me*review* penelitian sebelumnya yang berhubungan dengan penelitian yang akan dilakukan, memahami teori tentang NLP, analisis sentiment, Lexicon, *TextBlob*, *VADER*, *Naive Bayes*, SVM.

2. Pengumpulan data / *Data Gathering.*

   Tahapan ini merupakan proses pengumpulan data yang cukup untuk diteliti.

3. Tahapan *Text Preprocessing* (*Cleaning data, Spelling Correction, Drop duplicate data*).

   Pada tahapan preprocessing teks, yang dilakukan adalah membersihkan dan mempersiapkan teks mentah untuk analisis lebih lanjut. Ini mencakup penghapusan tanda baca, stopwords, dan karakter khusus, normalisasi teks, koreksi ejaan, serta tokenisasi. Proses ini bertujuan untuk meningkatkan kualitas data sehingga lebih mudah dianalisis oleh algoritma pengolahan bahasa alami, memastikan hasil analisis yang lebih akurat dan bermakna.

4. Tahapan anotasi data.

   Tahapan anotasi data adalah proses pemberian label pada data teks yang akan dikelompokkan dalam kelompok sentimen yaitu positif, netral dan negatif. Anotasi data teks ini dilakukan menggunakan *library VADER* dan *TextBlob* dengan berbagai tahapan yaitu sebelum dan sesudah pemrosesan teks untuk melihat perubahan nilai hasil akhir anotasi dan hubungannya terhadap akurasi sentimen.

5. Tahapan selanjutnya pembobotan dan pemodelan.

   Tahapan ini merupakan tahapan dalam menghitung bobot dari setiap kata dan melakukan proses training pada model yang dibangun setelah itu menguji model menggunakan data tes yang telah disiapkan.

6. Tahap terakhir adalah evaluasi.

   Tahap Evaluasi adalah proses penting yang bertujuan untuk menilai kinerja model dalam memprediksi atau mengklasifikasikan data dengan benar.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1 Pengumpulan Data

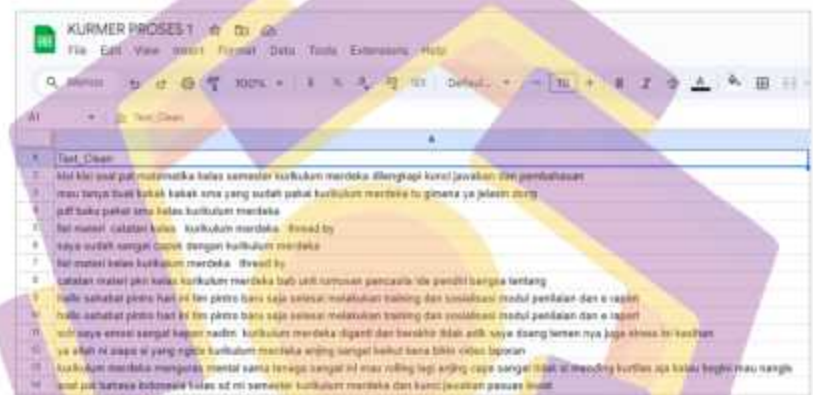
Pada penelitian ini, dataset diambil dari postingan cuitan (komentar) atau *tweet* pengguna *twitter* yang di *crawling* mulai dari tanggal 30 Maret 2023 sampai dengan 30 Juli 2023. Data yang diperoleh dari proses *crawling* tersebut sebanyak 1463 baris data. Kata kunci yang digunakan adalah "kurikulum merdeka".

Gambar 4.1. Tampilan halaman awal *Twitter*

4.2 Spelling Correction (Koreksi Ejaan)

Setelah dataset terkumpul, selanjutnya dilakukan pengecekan atau koreksi terhadap ejaan pada setiap kalimat. Spell chek atau koreksi ejaan yang dilakukan adalah untuk memperbaiki kesalahan ketikan atau *typo*. Hal ini bertujuan untuk menjaga nilai dan makna dari sebuah kalimat dalam proses analisis sentimen, karena seringkali kesalahan pengetikan dari sebuah kata dapat mempengaruhi nilai

polaritas, subjektivitas dan compund pada saat anotasi data menggunakan *TextBlob* dan *VADER* dilakukan. Koreksi ejaan pada file 'csv' dilakukan secara manual menggunakan *google sheets.* Pada *google sheets* pada menu *tools* terdapat menu *spell check* yang dapat digunakan untuk mengoreksi tulisan baik yang berbahasa Indonesia maupun bahasa inggris. Walaupun demikian, untuk koreksi ejaan menggunakan *google sheets* hanya dilakukan pada dataset bahasa Indonesia.



Gambar 4.2. Proses koreksi ejaan menggunakan *Google Sheets*

4.3 Translate

Menerjemahkan dataset yang berisi teks berbahasa indonesia menjadi teks berbahasa inggris merupakan langkah penting sebelum melakukan analisis sentimen, terutama saat melakukan anotasi menggunakan *library* seperti *VADER* dan *TextBlob* yang dirancang untuk bahasa Inggris. Proses ini dimulai dengan penerjemahan dataset, yang dapat dilakukan menggunakan alat penerjemahan otomatis seperti *google translate* atau manual oleh ahli bahasa untuk memastikan

akurasi. Setelah dataset diterjemahkan, tahap berikutnya adalah mempersiapkan data untuk anotasi sentimen. Anotasi ini melibatkan pengelompokan teks ke dalam kategori sentimen tertentu seperti positif, negatif, atau netral. *VADER* (Valence Aware Dictionary and sEntiment Reasoner) dan *TextBlob* adalah dua alat yang sering digunakan dalam analisis sentimen. *VADER*, yang dirancang khusus untuk media sosial, memberikan skor sentimen yang sangat akurat berdasarkan pola linguistik, sementara *TextBlob* menyediakan analisis sentimen serta alat pengolahan bahasa alami lainnya. Dengan menerjemahkan dataset terlebih dahulu, kedua alat ini dapat memanfaatkan kekayaan data teks dalam bahasa Inggris untuk memberikan hasil analisis yang lebih akurat dan bermakna. Selama proses anotasi, penting untuk mempertimbangkan konteks dan nuansa bahasa agar hasil analisis mencerminkan perasaan dan opini asli yang diungkapkan dalam teks asli berbahasa Indonesia.

Gambar 4.3. Proses *translate* dataset menggunakan *google translate*

4.4 Text Preprocessing

Text Preprocessing dilakukan pada Dataset yang telah di *crawl* dari *twitter* bertujuan untuk mengubah bentuk sebuah data yang tidak terstruktur menjadi lebih terstruktur sesuai dengan kebutuhan.

a. Proses *cleaning*

Proses *cleaning* pada dataset dilakukan untuk menghilangkan semua tanda baca dan simbol dari setiap baris agar tidak mengganggu proses *preprocessing* data nantinya.

Gambar 4.4. Dataset (bahasa Indonesia) setelah dilakukan proses *cleaning* data

Gambar 4.5. Dataset (bahasa Inggris) setelah dilakukan proses *cleaning* data

b. Proses *Case folding*

Setelah dataset bersih dari tanda baca dan simbol, kemudian dilakukan perubahan dari gabungan huruf besar dan kecil menjadi huruf kecil semua disetiap baris data.

Perubahan seluruh huruf dari data pada gambar 4.3 menjadi huruf kecil dapat dilihat pada gambar 4.6 dan 4.7 dibawah ini.

Gambar 4.6. Dataset (bahasa Indonesia) setelah dilakukan proses *case folding*

Gambar 4.7. Dataset (bahasa inggris) setelah dilakukan proses *case folding*

c. Proses pelabelan data

Setelah proses *case foleding* pada dataset dilakukan, maka selanjutnya dilakukan proses pelabelan data. Pada proses pelabelan data dilakukan menggunakan anotasi otomatis yangitu *VADER Sentimen* dan *TextBlob*.

	Text_Cleans	Compound_Score	LABVDR	Subjectivity	Polarity	LABTB
0	kisi pat mathematics class semester independen...	0.0000	Netral	0.5625	0.00	Netral

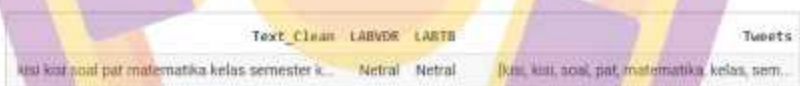
Gambar 4.8. Pelabelan bahasa Inggris

	Text_Clean	Compund_Score	LABVDR	Subjectivity	Polarity	LABTB
0	kisi kisi soal pat matematika kelas semester k...	0.0	Netral	0.0	0.0	Netral

Gambar 4.9. Pelabelan Bahasa Inggris

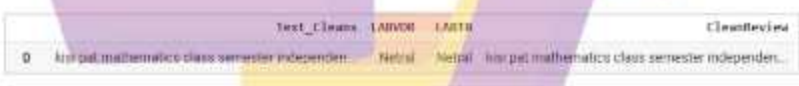
d. Proses *Tokenizing*

Proses ini bertujuan untuk memecah kalimat menjadi kata, dan juga pada proses ini pemisah kata atau bukan dapat dibedakan.



	Text_Clean	LABVDR	LABTB	Tweets
	kisi kisi soal pat matematika kelas semester k...	Netral	Netral	[kisi, kisi, soal, pat, matematika, kelas, sem...

Gambar 4.10. Hasil *tokenizing* Bahasa Indonesia



	Text_Cleans	LABVDR	LABTB	CleanReview
0	kisi pat mathematics class semester independen...	Netral	Netral	kisi pat mathematics class semester independen...

Gambar 4.11. Hasil *tokenizing* Bahasa Inggris

e. Proses *Filtering*

Tahap selanjutnya adalah *filtering*, dimana proses *filtering* ini bertujuan untuk menghilangkan kata-kata atau istilah-istilah yang masuk dalam kategori *Stop Words*.

	Text_Clean	LABVDR	LABTB	Tweets
0	kisi kisi soal pat matematika kelas semester k...	Netral	Netral	[kisi, kisi, pat, matematika, kelas, semester,...

Gambar 4.12. Hasil *filtering* Bahasa Indonesia

	Text_Cleans	LABVDR	LABTB	CleanReview
0	kisi pat mathematics class semester independen...	Netral	Netral	kisi pat mathematics class semester independen...

Gambar 4.13. Hasil *filtering* Bahasa Inggris

f. Proses *Stemming*

Tahap berikut adalah proses pengembalian kata-kata yang tidak dalam bentuk dasar kedalam bentuk kata dasar.

	Text_Clean	LABVDR	LABTB	Tweets
0	kisi kisi soal pat matematika kelas semester k...	Netral	Netral	[kisi, kisi, pat, matematika, kelas, semester,...

Gambar 4.14. Hasil *stemming* Bahasa Inggris

	Text_Cleans	LABVDR	LABTB	CleanReview
0	kisi pat mathematics class semester independen...	Netral	Netral	kisi pat mathematics class semester independen...

Gambar 4.15. Hasil *stemming* Bahasa Indonesia

4.5 Pembahasan

4.5.1. Dataset

Dalam penelitian ini, dataset yang digunakan adalah data cuit atau *tweet* yang di *crawling* dari *Twitter* dengan kata kunci "Kurikulum Merdeka" dan hasil dari *crawling* data tersebut diperoleh data *tweet* atau komentar sebanyak 1463 baris data dalam format file Comma Separated Value (CSV). Variabel yang digunakan dari dataset ini adalah data teks pada kolom komentar dari pengguna *twitter*.

```python
import tweepy
import pandas as pd
import numpy as np


access_token = "1518973592150888449-4Ut4RoBVmJ8pOURHfdsfHNKMoFsPTa"
access_token_secret = "4vz8mn7tO5ETARwQePXv5hgQ70Hk1qz2IlH5c80gc67dH"
API_key = "j0ILZRubk2D6RJaq1a7OMfYe2"
API_secret = "lrOxfdA8ZxjgZcxJ4p4MM93Qma0fKDvtu8nEirKqgOkIu1Z9OU"


auth = tweepy.OAuthHandler(API_key, API_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

tweets = api.search_tweets("kurikulum merdeka", count=3000, lang="id")
```

Gambar 4.16. *Script Crawling* data dari *Twitter*

Gambar 4.16 merupakan gambar *script crawling* data *tweet* dari sosial media *twitter*, dimana proses tersebut membutuhkan akses token dan API Key dari *twitter*. Pada *script* tersebut jumlah data yang ingin di *crawling* sebanyak 3000 data yang dan jenis data yang dikumpulkan adalah berupa teks yang menggunakan bahasa indonesia.

	Time	Username	Tweet
1	23-06-05 11:40:35+00:00	[illegible]	2 Kurikulum Merdeka: Dilengkapi Kunci Jawaban dan Pembahasan... https://t.co/[illegible]
2	23-06-05 11:35:34+00:00	[illegible]	[illegible] kurikulum merdeka [illegible]
3	23-06-05 11:28:38+00:00	[illegible]	GRADES: PDF Buku Paket SMA Kelas 10 Kurikulum Merdeka ... https://t.co/[illegible]
4	23-06-05 11:26:32+00:00	[illegible]	[illegible] Merdeka — a thread by @[illegible] #studytwt #studyacc #kelas10 https://t.c...
5	23-06-05 11:19:25+00:00	[illegible]	[illegible] kurikulum merdeka
6	23-06-05 10:59:30+00:00	[illegible]	[illegible] kelas 10 (kurikulum merdeka) — a thread by @[illegible] https://t.co/[illegible]
7	23-06-05 10:48:38+00:00	[illegible]	kurikulum merdeka [illegible] pancasila / ide pendiri bangsa tenta...
8	23-06-05 10:48:30+00:00	[illegible]	[illegible] Merdeka — a thread by @[illegible] #studytwt #studyacc #kelas10 https://t.c...
9	23-06-05 10:44:27+00:00	[illegible]	[illegible] melakukan training dan sosialisasi modul penilaian dan e-Raport d...
10	23-06-05 10:41:05+00:00	[illegible]	[illegible] melakukan training dan sosialisasi modul penilaian dan... https://t.co/[illegible]

Gambar 4.17. Hasil *Crawling* data dari *Twitter* (10 data teratas)

Gambar 4.17 merupakan hasil *Crawling* data dari *Twitter*. Data yang terkumpul dari proses tersebut hanya sebanyak 1463 dari 3000 data yang diinginkan. Hal ini dikarenakan, saat ini *twitter* telah membatasi jumlah data yang

dapat di *crawl* oleh pengguna. Disamping itu dataset yang terkumpul hanya bersifat komentar dan belum ada pelabelan sentimen dari setiao baris data komentar.

Dataset hasil *crawling* ini akan di gunakan dalam analisis sentimen menggunakan bahasa indonesia dan inggris. Dataset akan di translate menggunakan *google translate* dalam bentuk dokumen ".csv" yang di konversi terlebih dahulu dedalam bentuk ".xlsx" lalu dilakukan penerjemahan bahasa indonesia kedalam bahas ainggris dan setelah itu disimpan kembali dalam bentuk ".csv".

4.5.2. Text Preprocessing

a. Cleaning dataset

Dalam proses *cleaning*, *script* yang digunakan adalah sebagai berikut:

```
#Remove mention username dan Retweet
def remove_pattern(text,pattern_regex):
    r = re.findall(pattern_regex,text)
    for i in r:
        text = re.sub(i, '', text)
    return text

df['Clean'] = np.vectorize(remove_pattern)(df['Tweet'], " *RT* | *@[\w]*")
df
```

Gambar 4.18. *Script* membersihkan teks dalam kolom

Script pada gambar 4.18 digunakan untuk membersihkan teks dalam kolom '*tweet*' dari sebuah *DataFrame* (df) dengan menghapus pola teks tertentu, seperti *retweet* (RT) dan *mention* (*username* yang disebutkan), menggunakan fungsi *remove_pattern*. Fungsi *cleaning* ini kemudian digunakan dalam *script* untuk membersihkan kolom teks dalam suatu *DataFrame*.

Pada fungsi *remove_pattern* terdapat *input*, proses dan *output*, dimana *input* '*text*' merupakan *string* yang akan dibersihkan dan '*pattern_regex*' merupakan pola *regex* untuk pencarian *substring* yang akan dihapus. Pada bagian proses, kode program 're.findall(pattern_regex, text)' digunakan untuk menemukan semua *substring* yang cocok dengan *pattern_regex*, setelah itu iterasi melalui setiap *substring* yang cocok akan diganti dengan *string* kosong ' ' menggunakan kode program *re.sub(i, ' ', text)*, sedangkan yang menjadi *output* adalah pengembalian teks yang telah dimodifikasi, di mana *substring* yang cocok dengan *pattern_regex* telah dihapus.

Kode *np.vectorize* digunakan untuk mengubah fungsi *remove_pattern* menjadi fungsi yang dapat diterapkan pada array atau series dari pandas *DataFrame*. Dengan vektorisasi, fungsi *remove_pattern* dapat diterapkan ke setiap elemen dalam kolom *'Tweet'* secara efisien.

Pada fungsi 'df['Clean'] = np.vectorize(remove_pattern)(df['*Tweet*'], '' *RT* | *@[\w]*'')' bertujuan untuk mengambil kolom *'Tweet'* dari *DataFrame* 'df', lalu Menerapkan fungsi remove_pattern pada setiap elemen dalam kolom *'Tweet'* dengan pola *regex* '" *RT* | *@[\w]*"'. Pola *regex* '" *RT* "' digunakan untuk menghapus *retweet* yang ditandai dengan "RT" dan '" *@[\w]*"' digunakan untuk menghapus mention (username yang diawali dengan @ diikuti oleh karakter alfanumerik) setelah itu hasilnya disimpan dalam kolom baru *'Clean'* dalam *DataFrame* 'df'. Proses ini membantu dalam membersihkan teks agar analisis sentimen lebih akurat dan tidak terpengaruh oleh retweet atau mention yang tidak relevan.

```
def remove_special(text):
    #remove tab, new line, ans back slice
    text = text.replace('\\t',' ').replace('\\n',' ').replace('\\u',' ').replace('\\','')
    #remove non ASCII (emotion, chinese word, .etc)
    text = text.encode('ascii', 'replace').decode('ascii')
    #remove mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)"," ", text).split())
    #remove incomplete URL
    return text.replace("http://", " ").replace("https://", " ")
df['Text_Clean'] = df['Clean'].apply(remove_special)

#remove number
def remove_number(text):
    return re.sub(r"\d+", " ", text)
df['Text_Clean'] = df['Text_Clean'].apply(remove_number)

#remove_punctuation
def remove_punctuation(text):
    return text.translate(str.maketrans(' ',' ',string.punctuation))
df['Text_Clean'] = df['Text_Clean'].apply(remove_punctuation)
#remove whitespace leading & trailing
def remove_white_LT(text):
    return text.strip()
df['Text_Clean'] = df['Text_Clean'].apply(remove_white_LT)

#remove multiple whitespace into a single whitespace
def remove_white_M(text):
    return re.sub('\s+',' ',text)
df['Text_Clean'] = df['Text_Clean'].apply(remove_white_M)

def remove_RT(text):
    return re.sub(r'^RT[\s]+',' ',text)
df['Text_Clean'] = df['Text_Clean'].apply(remove_RT)

#remove single char
def remove_single(text):
    return re.sub(r"\b[a-zA-Z]\b",' ',text)
df['Text_Clean'] = df['Text_Clean'].apply(lambda x: remove_single(x))
```

Gambar 4.19. *Script* membersihkan teks dalam kolom

Sama seperti script pada gambar 4.18, script pada gambar 4.19juga digunakan untuk membersihkan teks dalam kolom. Hal ini dilakukan untuk *cleaning* data sekali lagi agar tidak ada yang terlewat, sehingga dataset sesuai dengan yang diinginkan.

Pada gambar 4.19, fungsi 'remove_special' digunakan untuk menghilangkan karakter khusus seperti tab (\t), baris baru (\n), dan backslash (\). Selain itu, fungsi ini juga menghapus karakter non-ASCII, serta menghilangkan mention, link, dan hashtag dari teks. Fungsi 'remove_number' digunakan untuk enghilangkan angka dari teks, 'remove_punctuation' untuk menghilangkan tanda baca dari teks, 'remove_white_LT' untuk menghapus spasi di awal dan akhir teks,

'remove_white_M' untuk mengganti beberapa spasi berturut-turut dengan satu spasi, 'remove_RT' untuk menghilangkan awalan 'RT' (yang biasanya menunjukkan *retweet* di *Twitter*) dari teks, dan 'remove_single' untuk enghilangkan karakter tunggal yang berdiri sendiri. Secara keseluruhan, *script* ini bertujuan untuk melakukan pembersihan teks yang intensif pada kolom Clean dalam *DataFrame* 'df' dan hasilnya disimpan dalam kolom baru *Text_Clean*. Setiap langkah dalam proses pembersihan ini menangani jenis noise atau karakter yang berbeda dalam teks, sehingga hasil akhirnya adalah teks yang lebih bersih dan siap untuk analisis lebih lanjut.

b. Case Folding

Pada *case folding* setiap data teks akan diubah mnejadi huruf kecil, dan *script* yang digunakan adalah sebagai berikut:

```
#Proses case folding
df['Text_Clean'] = df['Text_Clean'].str.lower()
```

Gambar 4.20. *Script case folding* pada teks dalam kolom

Kode 'df['Text_Clean']' mengacu pada kolom Text_Clean dalam *DataFrame* 'df'. Kolom ini berisi teks yang telah dibersihkan melalui beberapa tahap pemrosesan sebelumnya. Untuk kode '.str' merupakan atibut yang digunakan untuk mengakses metode string pada setiap elemen dalam kolom *pandas series*. Ini memungkinkan operasi string diterapkan pada seluruh kolom, dan '.lower()' digunakan untuk mengubah semua huruf dalam string menjadi huruf kecil.

Dengan menjalankan script ini, semua teks dalam kolom '*Text_Clean*' akan diubah menjadi huruf kecil. Hal ini berguna untuk menghilangkan perbedaan antara

huruf besar dan kecil yang dapat mempengaruhi analisis teks, seperti perbandingan kata atau penghitungan frekuensi kata. Misalnya, kata "Hello" dan "hello" akan dianggap sama setelah diubah menjadi huruf kecil, sehingga analisis menjadi lebih konsisten.

```
df.drop_duplicates(subset = "Text_Clean", keep = 'first', inplace = True)
```

Gambar 4.21. Menghapus baris data yang mempunyai isi yang sama

Script pada gambar 4.21 merupakan *script* yang digunakan untuk menghapus baris yang mempunyai isi yang sama. Disini tidak semua baris akan dihapus, hanya beberapa baris dan satu baris pertama akan ditinggalkan atau tidak dihapus.

c. Proses Pelabelan / Anotasi data

Proses pelabelan menggunakan *library TextBlob* dapat dilakukan dengan menggunakan *script* dibawah ini:

```
from textblob import TextBlob

def Subjectivity(review):
    return TextBlob(review).sentiment.subjectivity

def Polarity(review):
    return TextBlob(review).sentiment.polarity

def analyze(score):
    if score < 0:
        return 'Negatif'
    elif score == 0:
        return 'Netral'
    else:
        return 'Positif'

df['Subjectivity'] = df['Text_Clean'].apply(Subjectvity)
df['Polarity'] = df['Text_Clean'].apply(Polarity)
df['LABTB'] = df['Polarity'].apply(analyze)
df
```

Gambar 4.22. *Script* proses anotasi data menggunakan *TextBlob*

Script di atas menggunakan library *TextBlob* untuk melakukan analisis sentimen pada teks dalam dataframe `df`.

- Pertama, terdapat fungsi `Subjectivity(review)` yang menggunakan *TextBlob* untuk menghitung tingkat subjektivitas dari teks yang diberikan (`review`). Fungsi ini mengembalikan nilai subjectivity menggunakan metode `sentiment.subjectivity` dari objek *TextBlob*.
- Kedua, terdapat fungsi `Polarity(review)` yang juga menggunakan *TextBlob* untuk menghitung polaritas teks yang diberikan (`review`). Fungsi ini mengembalikan nilai polaritas menggunakan metode `sentiment.polarity` dari objek *TextBlob*.
- Ketiga, terdapat fungsi `analyze(score)` yang digunakan untuk mengklasifikasikan nilai polaritas (`score`) menjadi kategori sentimen berdasarkan aturan berikut:
  - Jika nilai `score` kurang dari 0, maka teks diklasifikasikan sebagai 'Negatif'.
  - Jika nilai `score` sama dengan 0, maka teks diklasifikasikan sebagai 'Netral'.
  - Jika nilai `score` lebih besar dari 0, maka teks diklasifikasikan sebagai 'Positif'.

Setelah definisi fungsi-fungsi tersebut, dataframe `df` diperbarui dengan kolom-kolom baru:

- `Subjectivity`: Menyimpan hasil dari fungsi `Subjectvity` yang telah diterapkan pada kolom 'Text_Clean' dari dataframe `df`.

- `Polarity`: Menyimpan hasil dari fungsi `Polarity` yang telah diterapkan pada kolom 'Text_Clean' dari dataframe `df`.
- `LABTB`: Menyimpan hasil dari fungsi `analyze` yang diterapkan pada kolom 'Polarity' untuk mengklasifikasikan nilai polaritas menjadi kategori sentimen ('Negatif', 'Netral', atau 'Positif').

Dengan melakukan langkah-langkah ini, script tersebut menghasilkan analisis sentimen untuk setiap teks dalam dataframe `df`, yang kemudian dapat digunakan untuk pemahaman lebih lanjut tentang pola sentimen dalam data teks yang dianalisis.

Untuk Proses pelabelan menggunakan *library VADER* dapat dilakukan dengan menggunakan *script* dibawah ini:

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()

# Asumsikan 'analyser' sudah didefinisikan dan 'df' sudah dimuat
scores = [analyser.polarity_scores(x) for x in df['Text_Clean']]
print(scores)

# Tambahkan nilai compound, neg, neu, dan pos ke DataFrame
df['Compound_Score'] = [x['compound'] for x in scores]
df['Neg'] = [x['neg'] for x in scores]
df['Neu'] = [x['neu'] for x in scores]
df['Pos'] = [x['pos'] for x in scores]

# Menampilkan DataFrame dengan kolom baru
print(df.head())
```

Gambar 4.23. *Script* proses anotasi data menggunakan *VADER*

Script di atas bertujuan untuk melakukan analisis sentimen pada teks bersih dalam kolom `Text_Clean` dari DataFrame `df` menggunakan pustaka *VADER* Sentiment. Pertama, pustaka `SentimentIntensityAnalyzer` diimpor dari modul

VADER Sentiment. Sebuah objek analyzer, `analyser`, diciptakan untuk menganalisis sentimen teks.

Dengan asumsi bahwa `analyser` telah diinisialisasi dan DataFrame `df` telah dimuat, langkah berikutnya adalah menghitung skor sentimen untuk setiap teks dalam kolom `Text_Clean`. Ini dilakukan menggunakan list comprehension: `[analyser.polarity_scores(x) for x in df['Text_Clean']]`. Fungsi `polarity_scores` dari *VADER* menghasilkan skor sentimen dalam bentuk dictionary untuk setiap teks, termasuk nilai compound (keseluruhan sentimen), neg (sentimen negatif), neu (sentimen netral), dan pos (sentimen positif). Hasil analisis sentimen ini disimpan dalam variabel `scores`.

Selanjutnya, skrip menambahkan nilai compound, neg, neu, dan pos ke DataFrame `df` dengan mengakses nilai yang sesuai dari dictionary skor sentimen. Ini dilakukan melalui list comprehension yang mengekstrak masing-masing nilai dari dictionary dalam `scores` dan menambahkan kolom baru ke DataFrame: `df['Compound_Score']`, `df['Neg']`, `df['Neu']`, dan `df['Pos']`.

Terakhir, *Script* menampilkan lima baris pertama dari DataFrame yang telah diperbarui menggunakan `print(df.head())`. Ini membantu dalam memverifikasi bahwa kolom baru dengan skor sentimen telah ditambahkan dengan benar ke DataFrame. Dengan demikian, *Script* ini memungkinkan kita untuk melakukan analisis sentimen yang komprehensif terhadap teks bersih dalam dataset, memberikan wawasan yang lebih mendalam tentang sentimen yang terkandung dalam teks tersebut. Hal ini juga memastikan bahwa proses pengolahan data berjalan sesuai harapan sebelum melanjutkan ke langkah analisis lebih lanjut.

d. Tokenizing

Proses *tokenizing* dilakukan dengan menggunakan *script* berikut ini:

```
#NLTK word Wrapper
def word_TW(text):
    return word_tokenize(text)
df['Tweets'] = df['Text_Clean'].apply(lambda x: word_TW(x))
```

Gambar 4.24. *Script* proses *tokenizing*

script ini bertujuan untuk melakukan tokenisasi kata-kata (menggunakan word_tokenize dari nltk) pada teks yang sudah dibersihkan (Text_Clean), dan hasil tokenisasi disimpan dalam kolom *Tweets*. Tokenisasi ini memecah teks menjadi unit-unit terkecil yang disebut token atau kata-kata, yang berguna untuk analisis lebih lanjut seperti pemodelan bahasa atau pemrosesan teks lainnya.

e. Filtering

Proses *filtering* dilakukan dengan menggunakan *script* berikut ini:

Gambar 4.25. *Script* proses *filtering*

Script ini bertujuan untuk mengelola dan memperluas daftar stopwords (kata-kata umum yang biasanya diabaikan dalam pemrosesan teks) untuk bahasa Indonesia, serta menerapkannya untuk menghapus stopwords dari teks yang telah

ditokenisasi dalam *DataFrame* `df`. Cara kerja *script* ini adalah pertama, *script* ini mengimpor daftar *stopwords* bawaan dari NLTK untuk bahasa Indonesia. Kemudian, *script* ini menambahkan stopwords tambahan secara manual ke dalam daftar tersebut untuk memperluas cakupan kata-kata yang akan diabaikan. Selain itu, *script* diatas juga membaca *stopwords* tambahan dari file CSV (`stopwords-id.csv`), lalu menggabungkan dan menggunakan semua *stopwords* untuk memastikan tidak ada duplikasi, dan kemudian mengaplikasikan fungsi penghapusan stopwords pada kolom `*Tweets*` di *DataFrame* `df`.

Secara khusus, fungsi `stopwords_removal` yang didefinisikan dalam script ini digunakan untuk memfilter setiap token dalam teks, memastikan bahwa hanya kata-kata yang tidak ada dalam daftar stopwords yang akan disimpan. Fungsi ini diterapkan pada kolom `*Tweets*` yang berisi daftar token dari teks yang telah ditokenisasi sebelumnya. Dengan melakukan ini, teks dalam kolom `*Tweets*` dibersihkan dari kata-kata umum yang tidak memiliki banyak makna dalam analisis lebih lanjut.

Kode 'list_stopwords = stopwords.words('indonesian')' adalah mengambil daftar stopwords standar dalam bahasa Indonesia dari *library* nltk. Kode 'list_stopwords.extend(['jg','yg','dg','dgn','rt','ny','d','klo','kalo','biar','bikin','bilang','gak','gk','krn','nya','sih','si','tau','tdk','tuh','ya','jd','jgn','sdh','aja','n'])' berfungsi untuk menambahkan kata-kata yang sering digunakan tetapi tidak memiliki nilai penting dalam analisis teks (stopwords tambahan) ke daftar stopwords. Fungsi 'def stopwords_removal(text)' adalah untuk menerima teks yang berupa daftar token dan mengembalikan daftar token tanpa stopwords.

f. Stemming

Proses *Stemming* pada dataset dapat dilakukan dengan menggunakan *script* berikut ini:

```
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in df['Tweets']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ''

print(len(term_dict))
print("------------------------")

for term in term_dict:
    term_dict[term] = stemmed_wrapper(term)
    print (term,":",term_dict[term])

print(term_dict)
print("------------------------")

#apply stemmed term to dataframe
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

df['Tweets']= df['Tweets'].swifter.apply(get_stemmed_term)
print(df['Tweets'])
```

Gambar 4.26. *Script* proses *stemming* bahasa Indonesia

Proses *stemming* dati *script* diatas adalah pertama, fungsi 'stemmed_wrapper' didefinisikan untuk melakukan stemming pada sebuah kata menggunakan stemmer yang telah diinisialisasi sebelumnya (misalnya, 'PorterStemmer' dari NLTK). Fungsi ini menerima satu parameter 'term' dan mengembalikan bentuk dasar (stem) dari kata tersebut. Selanjutnya, sebuah dictionary 'term_dict' diinisialisasi untuk menyimpan kata-kata unik dari kolom '*Tweets*' beserta bentuk dasar (stem) masing-masing kata. Kemudian, dilakukan iterasi melalui setiap dokumen (daftar token) dalam kolom '*Tweets*' dari *DataFrame* 'df'. Pada iterasi ini, setiap kata yang unik ditambahkan ke 'term_dict'

dengan nilai awal berupa string kosong. Setelah proses ini selesai, dicetak jumlah kata unik yang ditemukan dan pembatas untuk memisahkan output.

Setelah itu dilakukan iterasi melalui setiap kata dalam 'term_dict' untuk menerapkan fungsi 'stemmed_wrapper' pada setiap kata. Hasil stemming dari setiap kata disimpan kembali di 'term_dict', dan dicetak kata asli beserta bentuk stemnya. Dicetak kembali seluruh isi term_dict dan pembatas untuk memisahkan output. Setelah itu, didefinisikan fungsi 'get_stemmed_term' yang menerima sebuah dokumen (daftar token) dan mengembalikan daftar token yang telah di-stemming dengan menggunakan 'term_dict'. Fungsi 'get_stemmed_term' kemudian diterapkan pada kolom '*Tweets*' dari *DataFrame* 'df' menggunakan metode 'apply' dari 'swifter', yang mempercepat proses aplikasi fungsi pada *DataFrame* yang besar. Hasil stemming dari setiap dokumen disimpan kembali ke kolom '*Tweets*'.

4.5.2. Anotasi Data

Proses pelabelan atau anotasi merupakan proses pemberian label sentimen terhadap setiap baris data. Untuk anotasi data menggunakan *library VADER* dapat dilakukan dengan menggunakan *script* berikut ini:

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()

scores = [analyser.polarity_scores(x) for x in df['Tweet']]
print(scores)

df['Compound_Score'] = [x['compound'] for x in scores]
df['Neg'] = [x['neg'] for x in scores]
df['Neu'] = [x['neu'] for x in scores]
df['Pos'] = [x['pos'] for x in scores]
```

Gambar 4.27. Proses anotasi data menggunakan *VADER Sentiment*

Untuk penjelasan dari *script* anotasi data pada gambar 4.18 menggunakan *VADER* dapat dilihat pada tabel 4.1 dibawah ini:

Tabel 4.1. Tabel penjelasan *script* anotasi data menggunakan *VADER Sentiment*

Langkah	Keterangan
Import *Library*	Memuat pustaka ***VADER*Sentiment** dan mengimpor **SentimentIntensityAnalyzer** untuk analisis sentimen.
Inisialisasi Analizer	Membuat objek **analyser** dari **SentimentIntensityAnalyzer()** untuk melakukan analisis sentimen.
Penghitungan Skor Sentimen	Mengiterasi setiap teks dalam kolom *'Tweet'* dari dataframe **df** dan menghitung skor sentimen menggunakan **analyser.polarity_scores(x)**, yang mengembalikan nilai skor sentimen dalam bentuk *dictionary* untuk setiap teks.
Penyimpanan Skor Sentimen	Menyimpan skor-skor sentimen yang dihasilkan ke dalam variabel **scores**, di mana setiap elemen dalam **scores** berisi *dictionary* dengan nilai *compound*, negatif (Neg), netral (Neu), dan positif (Pos) dari sentimen untuk setiap teks.
Penambahan Kolom pada DataFrame	Memperbarui dataframe **df** dengan menambahkan kolom-kolom baru: **Compound_Score** untuk nilai sentimen komposit (compound), Neg untuk nilai sentimen negatif, Neu untuk nilai sentimen netral, dan Pos untuk nilai sentimen positif. Setiap nilai diambil dari *dictionary* scores sesuai dengan jenis sentimen yang sesuai.

Script ini bertujuan untuk menerapkan analisis sentimen menggunakan *VADER* pada dataset tweet yang terdapat dalam dataframe `df`. Hasilnya adalah dataframe yang diperbarui dengan kolom-kolom baru yang menggambarkan berbagai aspek sentimen dari setiap teks *tweet*, yang dapat digunakan untuk analisis lebih lanjut dalam konteks penelitian sentimen pada media sosial.

. Untuk anotasi data menggunakan *library TextBlob* dapat dilakukan dengan menggunakan *script* berikut ini:

```
from textblob import TextBlob

def Subjectvity(review):
    return TextBlob(review).sentiment.subjectivity

def Polarity(review):
    return TextBlob(review).sentiment.polarity

def analyze(score):
    if score < 0:
        return 'Negatif'
    elif score == 0:
        return 'Netral'
    else:
        return 'Positif'

df['Subjectivity'] = df['Tweet'].apply(Subjectvity)
df['Polarity'] = df['Tweet'].apply(Polarity)
df['LABEL'] = df['Polarity'].apply(analyze)
```

Gambar 4.28. Proses anotasi data menggunakan *TextBlob*

Untuk penjelasan dari *script* anotasi data pada gambar 4.28 menggunakan *TextBlob* dapat dilihat pada tabel 4.2 dibawah ini:

Tabel 4.2. Tabel penjelasan *script* anotasi data menggunakan *TextBlob*

Langkah	Keterangan
Import *Library*	Mengimpor fungsi ***TextBlob*** dari *library* ***TextBlob*** untuk analisis sentimen berbasis aturan.
Definisi Fungsi Subjectvity	Fungsi **Subjectvity(review)** digunakan untuk menghitung tingkat subjektivitas dari sebuah *review* teks menggunakan *TextBlob*. Mengembalikan nilai *subjectivity* dari sentimen yang dianalisis.
Definisi Fungsi Polarity	Fungsi **Polarity(review)** digunakan untuk menghitung polaritas dari sebuah *review* teks menggunakan *TextBlob*. Mengembalikan nilai *polarity* dari sentimen yang dianalisis.
Definisi Fungsi analyze	**Fungsi analyze(score)** digunakan untuk mengkategorikan nilai polarity menjadi 'Negatif', 'Netral', atau 'Positif' berdasarkan skor yang diterima.
Penerapan Fungsi ke DataFrame	Menggunakan fungsi **apply()** pada kolom *'Tweet'* dari dataframe **df** untuk menghitung dan menyimpan nilai subjektivitas dalam kolom *'Subjectivity'*, nilai polaritas dalam

Tabel 4.2. (Lanjutan)

	kolom *'Polarity'*, dan kategori sentimen dalam kolom 'LABTB' berdasarkan fungsi **analyze**.

Script ini digunakan untuk menerapkan analisis sentimen menggunakan *library TextBlob* pada dataset *tweet* yang terdapat dalam dataframe 'df'. Hasilnya adalah *dataframe* yang diperbarui dengan kolom-kolom baru yang menggambarkan tingkat subjektivitas, polaritas, dan kategori sentimen (Negatif, Netral, atau Positif) dari setiap teks *tweet*, yang dapat digunakan untuk analisis lebih lanjut dalam konteks penelitian sentimen pada media sosial.

4.5.2.1 Anotasi data Bahasa Indonesia

a. Anotasi dataset bahasa Indonesia menggunakan *TextBlob*

Pada proses anotasi dataset bahasa Indonesia menggunakan *TextBlob*, dilakukan empat kali anotasi data, antara lain:

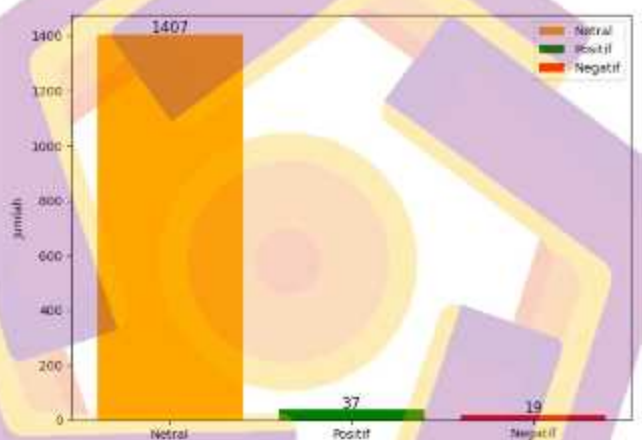
1. Anotasi ke – 1 dataset bahasa Indonesia menggunakan *TextBlob*

Anotasi data untuk dataset bahasa Indonesia yang pertama dilakukan adalah menganotasi data mentah yang belum di *cleaning*.

	Text_Clean	Subjectivity	Polarity	LABTB
62	RT @BukuERLANGGA: SahabatErlangga,\nErlangga S...	0.400000	0.200000	Positif
760	RT @educology_id: #SahabatEdu, Kurikulum Merde...	0.000000	0.000000	Netral
787	RT @studjellyca: [Catatan Sejarah Kelas 10]\n—...	1.000000	-0.750000	Negatif
794	@schfess Gaush keburuu, aku ga terlalu tau ya ...	0.000000	0.000000	Netral
807	RT @qanrz: hai i'm new #studytwt and i'm looki...	0.454545	0.170455	Positif
1199	RT @studjellyca: [Catatan Sejarah Kelas 10]\n—...	1.000000	-0.750000	Negatif

Gambar 4.29. Hasil anotasi data ke - 1 dataset indonesia menggunakan *TextBlob*

Pada gambar 4.29 data yang ditampilkan adalah enam data acak dari sentimen netral, positif dan negatif dari 1463 baris data yang dihasilkan menggunakan *library TextBlob*. Dapat dilihat ada variasi dalam tingkat subjektivitas dan polaritas terhadap kurikulum merdeka. *Tweet* dengan nilai polaritas negatif menunjukkan kritik atau ketidakpuasan, sementara *tweet* dengan nilai polaritas positif menunjukkan dukungan, sedangkan *tweet* dengan nilai netral cenderung tidak mengekspresikan sentimen yang kuat.



Gambar 4.30. Grafik hasil anotasi ke - 1 dataset Indonesia menggunakan *TextBlob*

Grafik pada gambar 4.30 merupakan hasil anotasi dataset yang pertama menggunakan *TextBlob*. Pada gambar tersebut terlihat bahwa jumlah sentimen yang dihasilkan adalah netral sebanyak 1407 atau 96,18%, positif sebanyak 37 atau 2,53% dan negatif sebanyak 19 atau 1,30%.

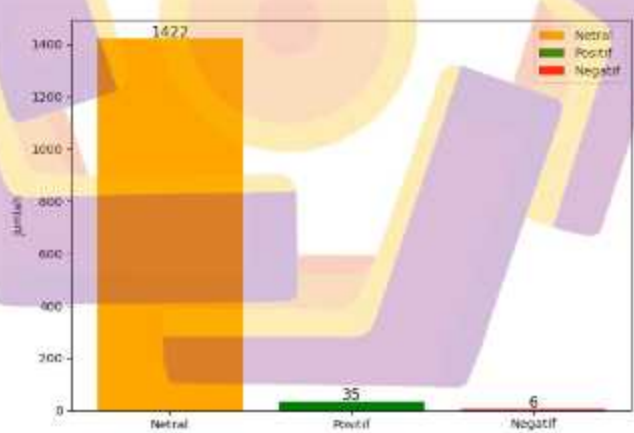
2. Anotasi ke – 2 dataset bahasa Indonesia menggunakan *TextBlob*

Anotasi data untuk dataset bahasa Indonesia yang kedua dilakukan dengan menganotasi data yang telah di *cleaning* namun tidak dilakukan *case folding*.

	Text_Clean	Subjectivity	Polarity	LABTB
62	RT @BukuERLANGGA: SahabatErlangga,\nErlangga S...	0.400000	0.200000	Positif
760	RT @educology_id: #SahabatEdu, Kurikulum Merde...	0.000000	0.000000	Netral
787	RT @studjellyca: [Catatan Sejarah Kelas 10]\n—...	1.000000	-0.750000	Negatif
794	@schfess Gaush keburuu, aku ga terlalu tau ya ...	0.000000	0.000000	Netral
807	RT @qanrz: hai i'm new #studytwt and i'm looki...	0.454545	0.170455	Positif
1199	RT @studjellyca: [Catatan Sejarah Kelas 10]\n—...	1.000000	-0.750000	Negatif

Gambar 4.31. Hasil anotasi ke - 2 dataset Indonesia menggunakan *TextBlob*

Dari gambar 4.31 tersebut, data yang ditampilkan adalah enam data acak dari sentimen netral, positif dan negatif dari 1463 baris data yang di anotasi. Sama seperti pada gambar 4.29 dapat dilihat variasi dalam tingkat subjektivitas dan polaritas terhadap kurikulum merdeka yang berbeda mulai dari nilai yang bersifat positif, netral hingga negatif.



Gambar 4.32. Grafik hasil anotasi ke - 2 dataset Indonesia menggunakan *TextBlob*

Dari grafik hasil anotasi dataset yang pertama pada gambar 4.32 menggunakan *TextBlob*, terlihat bahwa jumlah sentimen netral sebanyak 1422 atau

97,19%, positif sebanyak 35 atau 2,39% dan negatif sebanyak 6 atau 0,41%. Pada anotasi data yang ke – 2 terlihat adanya peningkatan dan penurunan disetiap sentimen. Hal tersebut dapat dilihat pada tabel ini:

Tabel 4.3. Tabel perbandingan perubahan nilai sentimen dari anotasi 1 dan 2 dataset Indonesia menggunakan *TextBlob*

Sentimen	Anotasi ke – 1	Anotasi ke - 2	Perubahan
Netral	96.18% / (1407)	97.19% / (1422)	Naik (+1.01%) (15)
Positif	2.53% / (37)	2.39% / (35)	Turun (-0.14%) (2)
Negatif	1.30% / (19)	0.41% / (6)	Turun (-0.89%) (13)

Dari tabel 4.3 dapat di lihat bahwa perubahan nilai Polarity dan Subjectivity dalam analisis sentimen antara anotasi ke-1 dan anotasi ke-2 menggambarkan perubahan dalam interpretasi emosional dan polaritas teks yang dibersihkan. Pada sentimen netral, terjadi kenaikan dari 96.18% menjadi 97.19%. Ini menunjukkan bahwa setelah proses pembersihan data pada anotasi ke-2, teks yang awalnya mungkin mengandung lebih banyak elemen non-informatif seperti angka atau emotikon telah dihapus, sehingga meningkatkan nilai netralitas teks yang dianalisis. Di sisi lain, sentimen positif menunjukkan penurunan dari 2.53% menjadi 2.39%, yang mengindikasikan bahwa setelah pembersihan data, nuansa positif dalam teks sedikit berkurang. Hal ini bisa disebabkan oleh penghapusan karakter yang mungkin memberikan sentimen positif yang lebih kuat sebelumnya. Sedangkan pada sentimen negatif, terjadi penurunan yang signifikan dari 1.30% menjadi 0.41%. Pengurangan ini menandakan bahwa setelah pembersihan data, teks yang awalnya cenderung mengekspresikan sentimen negatif, kemungkinan besar karena

penggunaan angka atau emotikon tertentu, telah mengalami penurunan yang besar dalam tingkat kenegatifannya. Dengan demikian, perubahan nilai Polarity dan Subjectivity ini mencerminkan bagaimana proses pembersihan data dapat mempengaruhi hasil analisis sentimen secara substansial dalam konteks penelitian skripsi ini.. Perubahan ini menunjukkan bahwa pembersihan data atau proses *cleaning* mempengaruhi hasil distribusi sentimen dengan lebih sedikit komentar yang dianggap negatif.

3. Anotasi ke – 3 dataset bahasa Indonesia menggunakan *TextBlob*

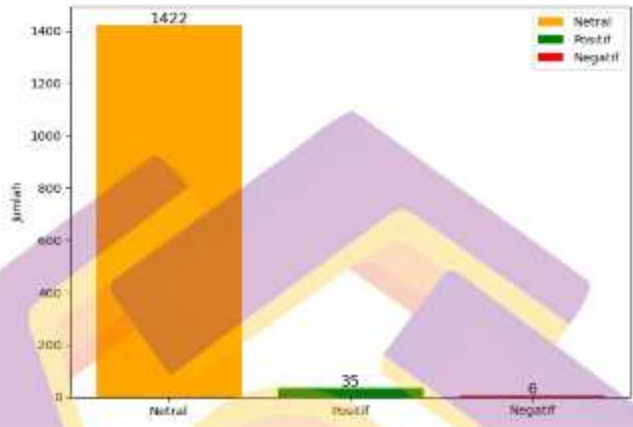
Anotasi data untuk dataset bahasa Indonesia yang ketiga dilakukan dengan menganotasi data yang telah di *spell checker, cleaning*, dan *case folding*.

	Text_Clean	Subjectivity	Polarity	LABTB
39	yg butuh lks sdsma terbaru tahun ajar bisa cek...	0.0	0.0	Netral
180	aku dulu gini minat mtk tp paling gasuka fisik...	0.0	0.0	Netral
278	publik speaking bikin ppt yg menarik dan gak b...	1.0	-1.0	Negatif
327	buat moots aku yg akt jangan jadi takut sma ya...	0.5	0.5	Positif
409	buat moots aku yg akt jangan jadi takut sma ya...	0.5	0.5	Positif
509	penjurusan in kurikulum merdeka is frustrating...	0.9	-0.4	Negatif

Gambar 4.33. Hasil anotasi ke - 3 dataset Indonesia menggunakan *TextBlob*

Dari gambar 4.33 tersebut, data yang ditampilkan adalah enam data acak dari sentimen netral, positif dan negatif dari 1463 baris data yang di anotasi. Sama seperti pada gambar 4.30 dan 4.32 dapat dilihat variasi dalam tingkat subjektivitas dan polaritas terhadap kurikulum merdeka yang berbeda mulai dari nilai yang bersifat positif, netral hingga negatif.

Untuk hasil anotasi yang ketiga menggunakan *library TextBlob* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.34. Grafik hasil anotasi ke - 3 dataset Indonesia menggunakan *TextBlob*

Dari grafik hasil anotasi dataset yang ke - 3 pada gambar 4.34 menggunakan *TextBlob*, terlihat bahwa jumlah sentimen netral sebanyak 1422 atau 97,19%, positif sebanyak 35 atau 2,39% dan negatif sebanyak 6 atau 0,41%. Pada anotasi data yang ke – 3 terlihat hasil dari anotasi yang dilakukan adalah sama dengan anotasi ke – 2 atau tidak ada peningkatan maupun penurunan pada hasil anotasi sentimen. Hal tersebut dapat dilihat pada tabel ini:

Tabel 4.4. Tabel perbandingan perubahan nilai sentimen dari anotasi 2 dan 3 dataset Indonesia menggunakan *TextBlob*

Sentimen	Anotasi ke – 2	Anotasi ke – 3	Perubahan
Netral	97.19% / (1422)	97.19% / (1422)	Sama (0%) (0)
Positif	2.39% / (35)	2.39% / (35)	Sama (0%) (0)
Negatif	0.41% / (6)	0.41% / (6)	Sama (0%) (0)

Dari tabel 4.4 dapat di lihat bahwa tidak ada perubahan pada nilai Polarity dan Subjectivity dalam analisis sentimen antara anotasi ke-2 dan anotasi ke-3, sehingga hasil anotasi ke – 3 sama dengan hasil anotasi ke - 2. Dari hasil anotasi ke – 3 dapat disimpulkan bahwa penggunaan *spell checker* dan *case folding* tidak berpengaruh pada anotasi data menggunakan *TextBlob*.

4. Anotasi ke – 4 dataset bahasa Indonesia menggunakan *TextBlob*

Anotasi data untuk dataset bahasa Indonesia yang ketiga dilakukan dengan menganotasi data yang telah di *spell checker*, *cleaning*, *case folding*, dan *Drop duplicate data*. Dengan adanya *drop duplicate data* atau penghapusan data yang sama dan meninggalkan satu data pertama, maka jumlah data pada dataset berkurang menjadi 667 baris data, sehingga akan mempengaruhi jumlah hasil sentimen baik positif, netral maupun negatif.

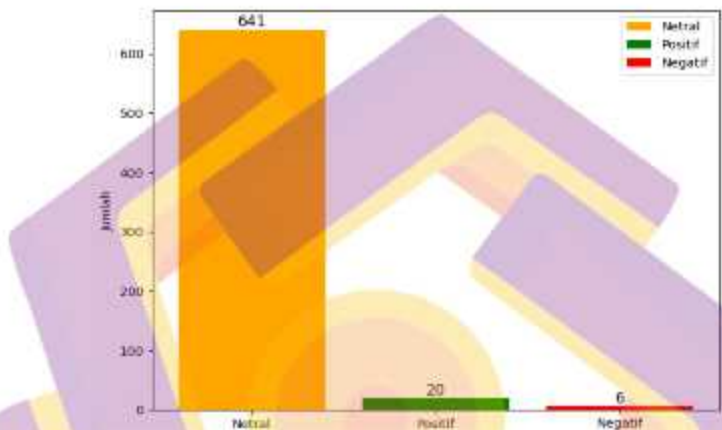
	Text_Clean	Subjectivity	Polarity	LABTB
38	sahabaterlangga erlangga straight point series...	0.4	0.2	Positif
161	publik speaking bikin ppt yg menarik dan gak b...	1.0	-1.0	Negatif
190	heyyy lagi angkatan huahauahau seumuran gksie ...	0.0	0.0	Netral
266	penjurusan in kurikulum merdeka is frustrating...	0.9	-0.4	Negatif
540	kurikulum merdeka makin tidak merdeka hahaha	0.4	0.2	Positif
556	kurmer itu kurikulum merdeka kahh ktnya sie sk...	0.0	0.0	Netral

Gambar 4.35. Hasil anotasi ke - 4 dataset Indonesia menggunakan *TextBlob*

Dari gambar 4.35 tersebut, data yang ditampilkan adalah enam data acak dari sentimen netral, positif dan negatif dari 667 baris data yang di anotasi setelah dilakukan *drop duplicate data*. Sama seperti pada anotasi 1, 2 dan 3, dapat dilihat

variasi dalam tingkat subjektivitas dan polaritas terhadap kurikulum merdeka yang berbeda mulai dari nilai yang bersifat positif, netral hingga negatif.

Untuk hasil anotasi yang keempat menggunakan *library TextBlob* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.36. Grafik hasil anotasi ke - 4 dataset Indonesia menggunakan *TextBlob*

Dari analisis grafik hasil anotasi dataset pertama yang ditunjukkan pada Gambar 4.36, terlihat bahwa distribusi sentimen dalam data yang dianalisis dengan menggunakan *TextBlob* menunjukkan bahwa sentimen netral mendominasi, dengan total mencapai 641 baris data atau 96,10% dari keseluruhan dataset. Di sisi lain, sentimen positif hanya tercatat sebanyak 20 baris data atau 3,00%, sedangkan sentimen negatif teridentifikasi pada 6 baris data atau 0,90%. Perbedaan hasil pada anotasi data yang keempat dibandingkan dengan hasil dari anotasi pertama, kedua, dan ketiga dapat dilihat jelas. Hal ini disebabkan oleh adanya perbedaan jumlah baris data pada dataset anotasi keempat yang tidak sama dengan dataset pada

anotasi pertama, kedua, dan ketiga. Hal tersebut dapat dilihat pada tabel dibawah ini:

Tabel 4.5. Tabel perbandingan perubahan nilai sentimen dari anotasi 1, 2, 3, dan 4 dataset Indonesia menggunakan *TextBlob*

Sentimen	**Anotasi ke – 1**	**Anotasi ke – 2**	**Anotasi ke – 3**	**Anotasi ke – 4**
Netral	96.18% / (1407)	97.19% / (1422)	97.19% / (1422)	96.10% / (641)
Positif	2.53% / (37)	2.39% / (35)	2.39% / (35)	3.00% / (20)
Negatif	1.30% / (19)	0.41% / (6)	0.41% / (6)	0.90% / (6)
Jumlah Data	1463	1463	1463	667

Dari tabel 4.5 dapat di lihat bahwa perbandingan hasil anotasi menggunakan *TextBlob* menunjukkan variasi dalam persentase sentimen antara empat anotasi. Pada anotasi pertama, kedua, dan ketiga, yang masing-masing memiliki jumlah data 1463, sentimen netral adalah 96.18% (1407), 97.19% (1422), dan 97.19% (1422) secara berurutan. Sentimen positif pada ketiga anotasi tersebut adalah 2.53% (37), 2.39% (35), dan 2.39% (35), sementara sentimen negatif adalah 1.30% (19), 0.41% (6), dan 0.41% (6). Namun, pada anotasi keempat, yang memiliki jumlah data 667 setelah dilakukan penghapusan duplikat, sentimen netral sedikit menurun menjadi 96.10% (641), sentimen positif meningkat menjadi 3.00% (20), dan sentimen negatif naik menjadi 0.90% (6). Perubahan ini mengindikasikan adanya pengaruh signifikan dari penghapusan data duplikat terhadap distribusi sentimen dalam analisis.

Tabel 4.6. Tabel perubahan nilai subjektivitas dan polaritas pada anotasi 1, 2, 3, dan 4 dataset Indonesia menggunakan *TextBlob*

Baris	Tweet	Subjectivity	Polarity	LABTB	Anotasi
216	SchApp! Guys kalian yang kelas 10 kurikulum merdeka spill dong mapel pilihan kalian buat kelas 11 sama jurusan buat kuliahnya :)	1.0	0.5	Positif	Anotasi ke - 1
216	SchApp Guys kalian yang kelas kurikulum merdeka spill dong mapel pilihan kalian buat kelas sama jurusan buat kuliahnya	0.0	0.0	Netral	Anotasi ke - 2
216	schapp guys kalian yang kelas kurikulum merdeka spill dong mapel pilihan kalian buat kelas sama jurusan buat kuliahnya	0.0	0.0	Netral	Anotasi ke – 3
125	schapp guys kalian yang kelas kurikulum merdeka spill dong mapel pilihan kalian buat kelas sama jurusan buat kuliahnya	0.0	0.0	Netral	Anotasi ke – 4

Pada tabel 4.6 menunjukkan variasi dalam nilai polaritas, subjektivitas, dan sentimen untuk setiap anotasi *tweet* yang sama menggunakan *TextBlob*. Pada anotasi ke-1, *tweet* memiliki subjektivitas yang tinggi dengan nilai 1.0 dan polaritas positif sebesar 0.5, menunjukkan evaluasi yang subjektif dan kecenderungan positif terhadap isi *tweet* tersebut. Anotasi ke-2, ke-3, dan ke-4 menunjukkan penurunan nilai subjektivitas dan polaritas menjadi 0.0, yang mengindikasikan bahwa *tweet* tersebut menjadi lebih objektif dan netral secara emosional dalam evaluasinya.

Perubahan nilai polaritas dan subjektivitas dapat menggambarkan bagaimana interpretasi terhadap konten berubah seiring dengan proses anotasi atau analisis yang lebih mendetail. Penurunan ke nilai netral dapat mengindikasikan bahwa konten tersebut lebih mendekati deskripsi faktual atau tidak mengandung penilaian emosional yang jelas.

Polaritas yang positif dan subjektivitas yang tinggi umumnya mewakili opini atau evaluasi pribadi yang kuat terhadap suatu topik, sementara polaritas netral dan subjektivitas rendah cenderung mencerminkan penjelasan objektif atau fakta yang tidak dibumbui dengan evaluasi pribadi.

b. Anotasi dataset bahasa Indonesia menggunakan *VADER*

Pada proses anotasi dataset bahasa Indonesia menggunakan *VADER*, dilakukan empat kali anotasi data, antara lain:

1. Anotasi ke – 1 dataset bahasa Indonesia menggunakan *VADER*

Anotasi data untuk dataset bahasa Indonesia yang pertama dilakukan adalah menganotasi data mentah yang belum di *cleaning*.

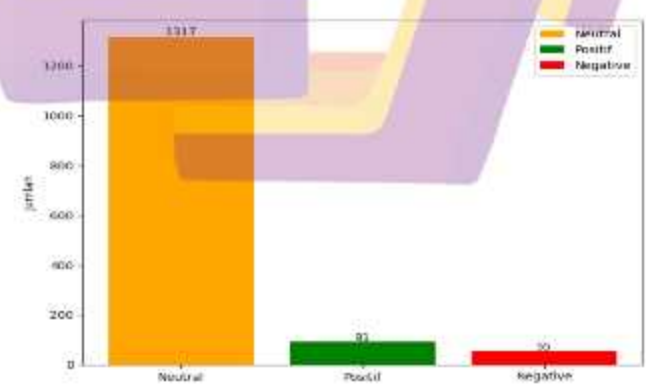
	Text_Clean	Compound_Score	Neg	Neu	Pos	LABVDR
90	Need joki buat modul 1 semester materi geograf...	-0.2732	0.104	0.896	0.000	Negatif
503	@schfess Gua klo ngerasain kurikulum merdeka k...	-0.4215	0.123	0.877	0.000	Negatif
718	RT @fairytars: ꒰ ˶ᵔ ᵕ ᵔ˶ ꒱ catatan materi ppkn kel...	0.6369	0.000	0.870	0.130	Positif
1077	RT @DEFINEGRADES: PDF Buku Paket SMA Kelas 10 ...	0.0000	0.000	1.000	0.000	Netral
1184	RT @st4rgazee: PDF Buku Paket SMA Kelas 10 Kur...	0.0000	0.000	1.000	0.000	Netral

Gambar 4.37. Hasil anotasi data ke - 1 dataset indonesia menggunakan *VADER*

Data pada tabel 4.37 merupakan 6 sampel data dari 1463 data yang telah dianotasi menggunakan *VADER*. Tabel tersebut menunjukkan bahwa hasil anotasi

diatas adalah hasil yang ditentukan oleh evaluasi sentimen berdasarkan *Compound Score* yang diberikan. Pada data tersebut, *tweet* pertama dan kedua mendapat nilai *Compound Score* negatif (-0.2732 dan -0.4215), dengan komposisi sentimen yang dominan pada aspek neutral (Neu) dan sedikit pada negatif (Neg). *Tweet* ketiga memiliki nilai *Compound Score* positif (0.6369), menandakan adanya sentimen positif dengan komposisi yang lebih tinggi pada aspek positif (Pos) dibandingkan aspek lainnya. Sedangkan *tweet* keempat dan kelima memiliki nilai *Compound Score* netral (0.0000), menunjukkan bahwa kedua *tweet* tersebut tidak mengandung evaluasi emosional yang signifikan dalam konteks analisis sentimen.

Pendekatan ke opini atau faktual dalam analisis ini tergantung pada nilai *Compound Score* dan komposisi nilai negatif, netral, dan positif. Nilai yang mendekati nol atau netral cenderung mencerminkan pendekatan yang lebih faktual atau deskriptif, sementara nilai positif atau negatif yang signifikan biasanya mencerminkan evaluasi atau opini pribadi terhadap topik yang dibahas dalam teks *tweet* tersebut.



Gambar 4.38. Grafik hasil anotasi ke - 1 dataset Indonesia menggunakan *VADER*

Grafik pada gambar 4.38 merupakan hasil anotasi dataset yang pertama menggunakan *VADER*. Pada gambar tersebut terlihat bahwa jumlah sentimen yang dihasilkan adalah netral sebanyak 1317 atau 90.05%, positif sebanyak 91 atau 6.22% dan negatif sebanyak 55 atau 3.75%.

2. Anotasi ke – 2 dataset bahasa Indonesia menggunakan *VADER*

Anotasi data untuk dataset bahasa Indonesia yang kedua dilakukan dengan menganotasi data yang telah di *cleaning* namun tidak dilakukan *case folding*.

	Text_Clean	Compound_Score	Neg	Neu	Pos	LABVDR
149	need joki tugas video pembelajaran berdiferens...	-0.2732	0.160	0.840	0.000	Negatif
473	Minggu berikutnya balik lg Yaallah pusing liat...	0.0000	0.000	1.000	0.000	Netral
672	Lits materi IPA kelas Kurikulum Merdeka enthus...	0.4404	0.000	0.805	0.195	Positif
863	Tapi kurikulum merdeka Gpp kan Aku share soon ya	0.2960	0.000	0.784	0.216	Positif
1169	joki tugas ulangan matematika soal type soalny...	-0.2732	0.139	0.861	0.000	Negatif
1204	PDF Buku Paket SMA Kelas Kurikulum Merdeka	0.0000	0.000	1.000	0.000	Netral

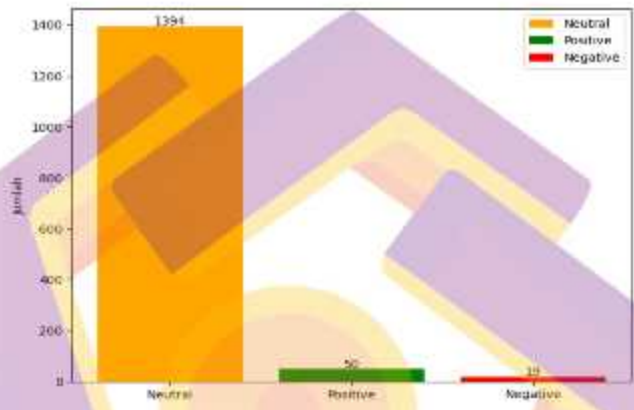
Gambar 4.39. Hasil anotasi ke - 2 dataset Indonesia menggunakan *VADER*

Dari Gambar 4.39 di atas, hasil analisis sentimen menggunakan library *VADER* menunjukkan nilai-nilai yang menggambarkan tingkat sentimen dari setiap teks. Compound Score menunjukkan sentimen keseluruhan: nilai positif berarti teks tersebut positif, nilai negatif berarti teks tersebut negatif, dan nilai netral berarti teks tidak memiliki sentimen yang kuat. Nilai negatif, netral, dan positif menunjukkan proporsi teks dalam masing-masing kategori sentimen.

Dari gambar 4.39, teks pertama (-0.2732) dan teks kelima (-0.2732) memiliki Compound Score negatif, menunjukkan sentimen negatif. Proporsi positif tertinggi terdapat pada teks keempat (0.2960) dan ketiga (0.4404), dengan nilai-

nilai yang mendekati 1 untuk sentimen positif. Teks kedua dan keenam memiliki nilai netral (1.000) yang menunjukkan ketiadaan sentimen yang kuat.

Untuk hasil anotasi yang kedua menggunakan *library VADER* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.40. Grafik hasil anotasi ke - 2 dataset Indonesia menggunakan *VADER*

Dari grafik hasil anotasi dataset yang pertama pada gambar 4.40 menggunakan *VADER*, terlihat bahwa jumlah sentimen netral sebanyak 1394 atau 95.22%, positif sebanyak 50 atau 3.42% dan negatif sebanyak 19 atau 1.30%. Pada anotasi data yang ke – 2 terlihat adanya peningkatan dan penurunan disetiap sentimen. Hal tersebut dapat dilihat pada tabel dibawah ini:

Tabel 4.7. Tabel perbandingan perubahan nilai sentimen dari anotasi 1 dan 2 dataset Indonesia menggunakan *VADER*

Sentimen	Anotasi ke – 1	Anotasi ke - 2	Perubahan
Netral	90.05% / (1317)	95.22% / (1394)	Naik (+5.17%) (77)
Positif	6.22% / (91)	3.42% / (50)	Turun (-2.80%) (41)
Negatif	3.75% / (55)	1.30% / (19)	Turun (-2.45%) (36)

Hasil anotasi menggunakan *VADER* pada Tabel 4.7 menunjukkan perubahan sentimen antara dua anotasi berbeda. Pada anotasi pertama, menggunakan dataset mentah, 90.05% teks netral (1317 teks), 6.22% teks positif (91 teks), dan 3.75% teks negatif (55 teks). Setelah anotasi kedua dengan dataset yang telah di-clean tanpa casefolding, teks netral meningkat menjadi 95.22% (1394 teks), sementara teks positif turun menjadi 3.42% (50 teks), dan teks negatif turun menjadi 1.30% (19 teks). Perubahan ini menunjukkan bahwa proses cleaning data dapat mempengaruhi hasil analisis sentimen.

3. Anotasi ke – 3 dataset bahasa Indonesia menggunakan *VADER*

Anotasi data untuk dataset bahasa Indonesia yang ketiga dilakukan dengan menganotasi data yang telah di *spell checker, cleaning*, dan *case folding*.

	Text_Clean	Compound_Score	Neg	Neu	Pos	LABVDR
7	list materi amp catatan kelas kurikulum merd...	0.0000	0.000	1.000	0.000	Netral
472	setuju aku juga kurikulum merdeka nder stress ...	-0.4215	0.167	0.833	0.000	Negatif
745	tk dan bangsa lancang kuning riau binaan fkdb ...	-0.5106	0.155	0.845	0.000	Negatif
863	tapi kurikulum merdeka gpp kan aku share soon yaa	0.2960	0.000	0.784	0.216	Positif
884	lits materi ipa kelas kurikulum merdeka enthus...	0.4404	0.000	0.805	0.195	Positif

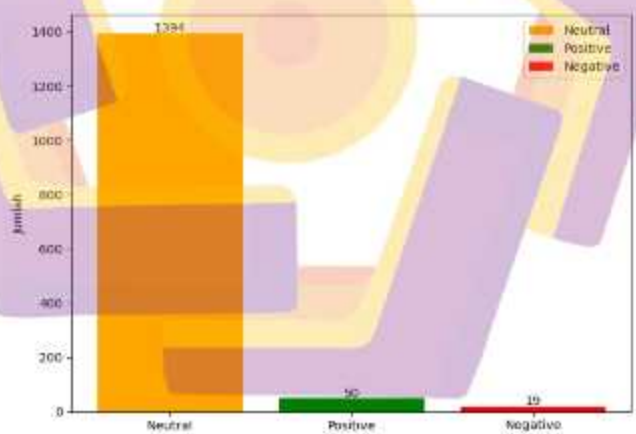
Gambar 4.41. Hasil anotasi ke - 3 dataset Indonesia menggunakan *VADER*

Dari gambar 4.41, data yang ditampilkan adalah enam sampel data acak dari sentimen netral, positif dan negatif dari 1463 baris data yang di anotasi. Dalam tabel tersebut, terlihat hasil anotasi menggunakan *library VADER,* sebagai contoh, pada baris kedua terdapat teks yang menyatakan "setuju aku juga kurikulum merdeka nder stress ...", yang diberi nilai compound score -0.4215. Skor negatif ini menunjukkan adanya sentimen yang cenderung negatif dalam teks tersebut. Penilaian ini mungkin dipengaruhi oleh kata-kata seperti "*stress*" yang secara

umum memiliki konotasi negatif dalam konteks diskusi mengenai kurikulum merdeka.

Di sisi lain, pada baris kelima terdapat teks yang mengandung frasa "lits materi ipa kelas kurikulum merdeka enthus...", dengan nilai compound score 0.4404. Skor positif ini mengindikasikan bahwa teks tersebut mengandung sentimen yang lebih positif. Kemungkinan, nilai positif ini dipengaruhi oleh kata-kata seperti "enthusiastic" yang memberikan nuansa positif dan dukungan terhadap kurikulum merdeka. Untuk hasil sentimen dari anotasi ke – 3, data sentimen yang dihasilkan sama dengan hasil anotasi data pada anotasi ke 2.

Untuk hasil anotasi yang ketiga menggunakan *library VADER* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.42. Grafik hasil anotasi ke - 3 dataset Indonesia menggunakan *VADER*

Dari grafik hasil anotasi dataset yang pertama pada gambar 4.42 menggunakan *VADER*, terlihat bahwa jumlah sentimen netral, positif dan negatif

tetap sama dengan hasil anotasi pada anotasi ke - 2. Hal tersebut dapat dilihat pada tabel ini:

Tabel 4.8. Tabel perbandingan perubahan nilai sentimen dari anotasi 2 dan 3 dataset Indonesia menggunakan *VADER*

Sentimen	Anotasi ke – 2	Anotasi ke – 3	Perubahan
Netral	95.22% / (1394)	95.22% / (1394)	Sama (0%) (0)
Positif	3.42% / (50)	3.42% / (50)	Sama (0%) (0)
Negatif	1.30% / (19)	1.30% / (19)	Sama (0%) (0)

Dari tabel 4.8 dapat di lihat bahwa tidak ada perubahan pada nilai *compound* dalam analisis sentimen antara anotasi ke-2 dan anotasi ke-3. Dari hasil anotasi ke – 3 dapat disimpulkan bahwa penggunaan *spell checker* dan *case folding* tidak berpengaruh pada anotasi data menggunakan *VADER*.

4. Anotasi ke – 4 dataset bahasa Indonesia menggunakan *VADER*

Anotasi data untuk dataset bahasa Indonesia yang ketiga dilakukan dengan menganotasi data yang telah di *spell checker*, *cleaning*, *case folding*, dan *Drop duplicate data*. Dengan adanya *drop duplicate data* atau penghapusan data yang sama dan meninggalkan satu data pertama, maka jumlah data pada dataset berkurang menjadi 667 baris data, sehingga akan mempengaruhi jumlah hasil sentimen baik positif, netral maupun negatif.

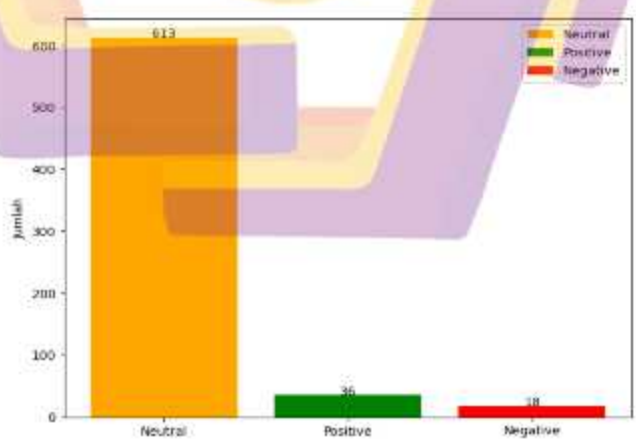
	Text_Clean	Compound_Score	Neg	Neu	Pos	LABVDR
161	publik speaking bikin ppt yg menarik dan gak b...	-0.3182	0.126	0.874	0.000	Negatif
312	aku sebagai anak smk yg sekolah nerapin kuriku...	0.0000	0.000	1.000	0.000	Netral
355	padahal kemaren lg hectic sama kurikulum merde...	0.0000	0.000	1.000	0.000	Netral
453	siapa yang pake kurikulum merdekaaaaaaaaaaaaaa...	0.0772	0.000	0.915	0.085	Positif
550	haha hihi taunya udah ga sekelas lagi ini semw...	0.4588	0.000	0.812	0.188	Positif

Gambar 4.43. Hasil anotasi ke - 4 dataset Indonesia menggunakan *VADER*

Gambar 4.43 menggambarkan hasil anotasi sentimen menggunakan *library VADER* dengan menampilkan hanya 6 sampel teks. Pada baris pertama terdapat teks yang menyatakan "publik speaking bikin ppt yg menarik dan gak b...", dengan nilai compound score -0.3182, menunjukkan sentimen negatif. Ini mungkin disebabkan oleh kata-kata yang mengandung kecemasan atau ketidaknyamanan terkait dengan persiapan publik speaking yang diungkapkan dalam teks.

Di sisi lain, terdapat juga teks yang diberi nilai compound score 0.4588 pada baris kelima, yang menandakan sentimen positif. Teks tersebut mengandung frasa seperti "haha hihi taunya udah ga sekelas lagi ini semw...", yang tampaknya mengandung rasa humor dan nostalgia positif terhadap pengalaman sekolah. Sentimen positif ini tercermin dari kata-kata seperti "haha" dan "hihi" yang menambahkan nuansa keceriaan.

Untuk hasil anotasi yang keempat menggunakan *library VADER* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.44. Grafik hasil anotasi ke - 4 dataset Indonesia menggunakan *VADER*

Dari grafik hasil anotasi dataset yang pertama pada gambar 4.44 menggunakan *VADER*, terlihat bahwa jumlah sentimen netral sebanyak 613 atau 91.96%, positif sebanyak 36 atau 5.40% dan negatif sebanyak 18 atau 2.70%. Hasil pada anotasi data yang ke – 4 berbeda dengan hasil dari anotasi 1, 2, dan 3, karena jumlah baris data pada dataset anotasi ke – 4 berbeda. Hal tersebut dapat dilihat pada tabel ini:

Tabel 4.9. Tabel perbandingan perubahan nilai sentimen dari anotasi 1, 2, 3, dan 4 dataset Indonesia menggunakan *VADER*

Sentimen	**Anotasi ke – 1**	**Anotasi ke – 2**	**Anotasi ke – 3**	**Anotasi ke – 4**
Netral	90.05% / (1317)	95.22% / (1394)	95.22% / (1394)	91.96% / (613)
Positif	6.22% / (91)	3.42% / (50)	3.42% / (50)	5.40% / (36)
Negatif	3.75% / (55)	1.30% / (19)	1.30% / (19)	2.70% / (18)
Jumlah Data	1463	1463	1463	667

Tabel 4.9 menunjukkan hasil anotasi sentimen menggunakan metode *VADER* untuk empat kali penilaian berbeda. Pada anotasi pertama, sekitar 90.05% dari total data dinyatakan sebagai netral, sementara 6.22% memiliki sentimen positif dan 3.75% menunjukkan sentimen negatif. Pada anotasi kedua dan ketiga, persentase sentimen netral meningkat menjadi 95.22%, sedangkan persentase sentimen positif dan negatif menurun menjadi sekitar 3.42% dan 1.30% secara berturut-turut. Anotasi keempat menunjukkan perubahan, di mana sentimen netral tetap dominan dengan 91.96%, sementara sentimen positif dan negatif berada pada 5.40% dan 2.70%. Total data yang dianotasi dalam tabel ini adalah 1463 pada setiap

anotasi, kecuali untuk anotasi keempat yang berjumlah 667 data. Perubahan persentase dalam setiap anotasi mencerminkan variasi dalam penilaian sentimen berdasarkan konten teks yang dianalisis menggunakan.

Tabel 4.10. Tabel perubahan nilai *compound* menggunakan 1 sampel dari anotasi 1, 2, 3, dan 4 pada dataset Indonesia menggunakan *VADER*

Baris	Tweet	Compound Score	Neg	Neu	Pos	LAB VDR	Anotasi
715	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt 😩😩😩 TAPI PLS GWEH GA MAU KURIKULUM 2013 YG IPA IPS PLSS NA... https://t.co/uJhUsN1Uwb	-0.633	0.13	0.51	0.06	Negatif	Anotasi ke - 1
715	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt TAPI PLS GWEH GA MAU KURIKULUM YG IPA IPS PLSS NA	0.2577	0.0	0.90	0.09	Positif	Anotasi ke - 2
715	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt tapi pls gweh ga mau kurikulum yg ipa ips plss na	0.0772	0.0	0.93	0.06	Positif	Anotasi ke - 3
353	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt tapi pls gweh ga mau kurikulum yg ipa ips plss na	0.0772	0.0	0.93	0.06	Positif	Anotasi ke - 4

Tabel 4.10 menampilkan hasil analisis sentimen dari sebuah *tweet* yang telah dianotasi menggunakan *VADER*. *Tweet* ini berkaitan dengan kurikulum Merdeka dan menyatakan kelelahan siswa yang mengikuti kurikulum tersebut. Setiap baris dalam tabel mencerminkan hasil analisis sentimen berdasarkan variasi kecil dari teks yang sama, dengan hasil yang berbeda dalam skor komposit (Compound_Score), nilai negatif (Neg), nilai netral (Neu), dan nilai positif (Pos).

Pada baris pertama, *tweet* yang dianalisis menyertakan emotikon yang menunjukkan kelelahan dan kesedihan, serta tautan ke konten eksternal. Hasil analisis menunjukkan Compound_Score sebesar -0.633, dengan skor negatif (Neg) 0.192, netral (Neu) 0.745, dan positif (Pos) 0.063. Anotasi ini dikategorikan sebagai negatif. Emotikon yang digunakan serta struktur kalimat yang lebih emosional mungkin berkontribusi terhadap skor negatif yang lebih tinggi.

Pada baris kedua, emotikon dan tautan telah dihilangkan dari *tweet*, menghasilkan Compound_Score yang jauh lebih positif yaitu 0.2577, dengan nilai negatif 0.0, netral 0.903, dan positif 0.097. Anotasi ini dikategorikan sebagai positif. Penghilangan elemen-elemen emosional yang kuat dan fokus pada teks murni kemungkinan besar mempengaruhi perubahan skor menjadi lebih positif.

Pada baris ketiga dan keempat, *tweet* tersebut diulang dengan teks yang hampir identik dan tidak menyertakan emotikon atau tautan. Kedua hasil analisis ini menunjukkan Compound_Score sebesar 0.0772, nilai negatif 0.0, nilai netral 0.936, dan nilai positif 0.064, yang juga dikategorikan sebagai positif. Tidak ada perubahan isi kalimat *tweet* walau sudah dilakukan spell checker dan case folding. Perbedaan ini dibandingkan dengan baris kedua lebih kecil dan mungkin disebabkan oleh variasi minor dalam pemrosesan teks oleh *VADER*, namun secara umum, hasil ini menunjukkan sentimen yang lebih netral-positif.

Analisis perubahan nilai Compound, serta nilai Neg, Neu, dan Pos, menunjukkan bahwa elemen seperti emotikon dan tautan memiliki dampak signifikan terhadap hasil sentimen. Emotikon yang menunjukkan emosi negatif kuat dapat meningkatkan skor negatif dan menurunkan skor positif, menghasilkan

keseluruhan sentimen yang lebih negatif. Sebaliknya, teks yang bersih dari elemen emosional cenderung menghasilkan sentimen yang lebih netral atau bahkan positif. Ini menunjukkan pentingnya mempertimbangkan konteks penuh *tweet*, termasuk simbol-simbol emosional, dalam analisis sentimen.

c. Analasisis perbandingan hasil Anotasi bahasa Indonesia menggunakan *TextBlob* dan *VADER*

Perbandingan hasil analisis dari *library TextBlob* dan *VADER* terhadap dataset bahasa Indonesia dilakukan dengan mengambil satu sampel atau satu baris data yang sama dari setiap anotasi yang telah dilakukan. Sampel tersebut akan menjadi bahan analisis dengan tujuan agar dapat melihat perubahan pola polaritas dan subjektivitas pada *TextBlob* dan perubahan pola *compound score* dari *VADER*.

Tabel 4.11. Tabel perbandingan hasil anotasi dataset indonesia pada 1 sampel baris data menggunakan *VADER* dan *TextBlob*

Baris	Tweet	Compound Score	LABVDR	Subjectivity	Polarity	LABTB	Anotasi
715	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt 😡😡😡 TAPI PLS GWEH GA MAU KURIKULUM 2013 YG IPA IPS PLSS NA...	-0.633	Negatif	0.0	0.0	Netral	Anotasi ke - 1
715	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt TAPI PLS GWEH GA MAU KURIKULUM YG IPA IPS PLSS NA	0.2577	Positif	0.0	0.0	Netral	Anotasi ke - 2
715	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt	0.0772	Positif	0.0	0.0	Netral	Anotasi ke - 3

Tabel 4.11. (Lanjutan)

	tapi pls gweh ga mau kurikulum yg ipa ips plss na						
353	gue sbg anak kurikulum merdeka mengaku aslinya capek bgt tapi pls gweh ga mau kurikulum yg ipa ips plss na	0.0772	Positif	0.0	0.0	Netral	Anotasi ke - 4

Tabel 4.11 menggambarkan anotasi sentimen pada beberapa *tweet* yang memiliki isi hampir sama, namun dengan variasi yang sedikit berbeda dalam penulisan dan konten. Analisis ini melibatkan dua alat analisis sentimen: *VADER* dan *TextBlob*, yang memberikan hasil berbeda pada sentimen dan komponen-komponen emosional lainnya.

Pada *tweet* pertama, sentimen negatif mendominasi dengan skor *compound* -0.633, menunjukkan adanya emosi yang sangat negatif. *VADER* mengidentifikasi 19.2% dari teks ini bersifat negatif, 74.5% netral, dan hanya 6.3% yang positif. Anotasi sentimen ini adalah "Negatif" dan *TextBlob* menilai sentimen sebagai "Netral". Subjektivitas dan polaritas *TextBlob* keduanya berada di 0.0, yang menunjukkan bahwa teks ini dinilai sangat objektif dan tidak memiliki kecenderungan emosional yang kuat.

Selain itu pada *tweet* kedua, meskipun konten hampir sama, perubahan dalam penulisan dan penghapusan emoji menyebabkan skor *compound* naik menjadi 0.2577, menunjukkan perubahan ke arah sentimen positif. Tidak ada komponen negatif yang teridentifikasi, 90.3% dari teks bersifat netral, dan 9.7% bersifat positif. Meskipun LABVDR menilai sentimen sebagai "Positif", *TextBlob* masih menilai sentimen ini sebagai "Netral" dengan subjektivitas dan polaritas tetap

di 0.0. Hal ini menyoroti bahwa perubahan kecil dalam teks dapat mempengaruhi interpretasi sentimen, terutama dalam analisis otomatis.

Tweet ketiga dan keempat memiliki skor *compound* yang sama yaitu 0.0772, dengan hasil analisis *VADER* yang menunjukkan 93.6% netral dan 6.4% positif. Anotasi sentimen pada kedua *tweet* ini adalah "Positif", meskipun nilai positif yang ditemukan cukup rendah. *TextBlob* tetap memberikan hasil "Netral" dengan subjektivitas dan polaritas di 0.0. Hal ini menunjukkan bahwa meskipun ada perubahan kecil dalam teks, hasil analisis *VADER* dan *TextBlob* cenderung konsisten dalam menilai sentimen sebagai netral dengan sedikit kecenderungan positif.

Dari analisa perubahan nilai compound dan polaritas atau subjektivitas dalam anotasi sentimen pada dataset Indonesia dapat memberikan wawasan tentang bagaimana perbedaan kecil dalam teks dapat mempengaruhi hasil analisis. *VADER* cenderung lebih sensitif terhadap perubahan kecil dalam teks dan dapat menunjukkan pergeseran yang signifikan dalam skor sentimen, sementara *TextBlob* menunjukkan stabilitas lebih besar dalam analisis subjektivitas dan polaritas. Ini mengindikasikan bahwa *VADER* mungkin lebih cocok untuk menangkap nuansa emosional yang halus dalam teks bahasa indonesia, sementara *TextBlob* dapat digunakan untuk analisis yang lebih stabil dan objektif. Perbedaan ini sangat penting untuk dipertimbangkan ketika memilih alat analisis sentimen yang tepat sesuai kebutuhan spesifik. Oleh karena itu, pemilihan model yang digunakan harus disesuaikan dengan konteks dan tujuan analisis sentimen yang ingin dicapai.

4.5.2.1 Anotasi data Bahasa Inggris

a. Anotasi dataset bahasa inggris menggunakan *TextBlob*

Pada proses anotasi dataset bahasa inggris dataset yang digunakan adalah dataset yang telah ditranslate dari data mentah bahasa Indonesia menggunakan *google translate*. Proses anotasi pertama menggunakan *TextBlob*, dilakukan empat kali anotasi data, antara lain:

1. Anotasi ke – 1 dataset bahasa inggris menggunakan *TextBlob*

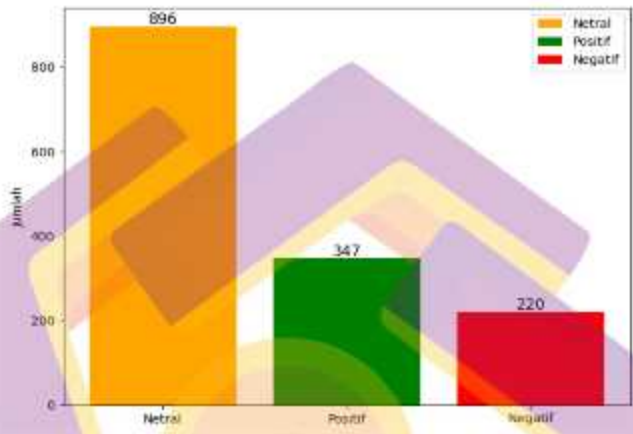
Anotasi data untuk dataset bahasa inggris yang pertama dilakukan adalah menganotasi data mentah yang belum di *cleaning*.

	Tweet	Subjectivity	Polarity	LABTB
0	KISI PAT Mathematics Class 4 Semester 2 Indepe...	0.5625	0.00	Netral
1	I want to ask high school seniors who are alre...	0.3325	0.08	Positif
2	RT @DEFINEGRADES	0.0000	0.00	Netral
3	RT @serinshi: List of materials & notes fo...	0.0000	0.00	Netral
4	I'm really tired of the independent curriculum 😔	0.4125	-0.20	Negatif
...	...	...	...	...
1458	RT @k___ale: list of class 10 materials (indep...	0.1250	0.00	Netral
1459	Work title and art performance implementing th...	0.0000	0.00	Netral
1460	RT @k___ale	0.0000	0.00	Netral
1461	Class 10 High School Independent Curriculum Bo...	0.3325	0.08	Positif
1462	Independent Curriculum Book for Grade 8 Middle...	0.0625	0.00	Netral

Gambar 4.45. Hasil anotasi data ke - 1 dataset inggris menggunakan *TextBlob*

Gambar 4.45 merupakan gambar hasil anotasi data ke – 1, dimana anotasi tersebut menghasilkan sentimen netral, positif dan negatif dari 1463 baris data yang menggunakan *library TextBlob*. Dapat dilihat ada variasi dalam tingkat subjektivitas dan polaritas terhadap kurikulum merdeka. *Tweet* dengan nilai

polaritas negatif menunjukkan kritik atau ketidakpuasan, sementara *tweet* dengan nilai polaritas positif menunjukkan dukungan, sedangkan *tweet* dengan nilai netral cenderung tidak mengekspresikan sentimen yang kuat.



Gambar 4.46. Grafik hasil anotasi ke - 1 dataset inggris menggunakan *TextBlob*

Grafik pada gambar 4.46 merupakan hasil anotasi dataset yang pertama menggunakan *TextBlob*. Pada gambar tersebut terlihat bahwa jumlah sentimen yang dihasilkan adalah netral sebanyak 896 atau 61.24%, positif sebanyak 347 atau 23.72% dan negatif sebanyak 220 atau 15.04%. Pada anotasi data menggunakan bahasa inggris, hasil pelabelan data menggunakan *library TextBlob* lebih varitif dibandingkan dengan bahasa Indonesia, karena kamus *TextBlob* lebih mendukung data berbahasa inggris dari pada bahasa lainnya.

2. Anotasi ke – 2 dataset bahasa inggris menggunakan *TextBlob*

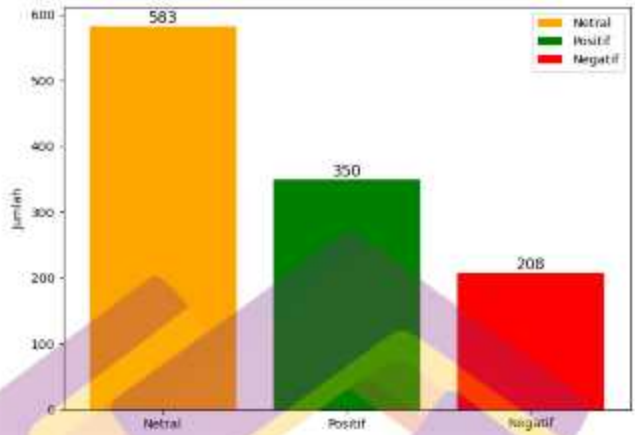
Anotasi data untuk dataset bahasa inggris yang kedua dilakukan dengan menganotasi data yang telah di *cleaning* namun tidak dilakukan *case folding*.

	Text_Clean	Subjectivity	Polarity	LABTB
0	KISI PAT Mathematics Class Semester Independen...	0.5625	0.00	Netral
1	want to ask high school seniors who are alre...	0.3325	0.08	Positif
3	List of materials amp notes for class	0.0000	0.00	Netral
4	Im really tired of the independent curriculum	0.4125	-0.20	Negatif
5	list of class materials independent curriculum...	0.1250	0.00	Netral
...	...	...	...	...
1457	Just read any independent curriculum	0.1250	0.00	Netral
1458	list of class materials independent curriculum...	0.1250	0.00	Netral
1459	Work title and art performance implementing th...	0.0000	0.00	Netral
1461	Class High School Independent Curriculum Book	0.3325	0.08	Positif
1462	Independent Curriculum Book for Grade Middle S...	0.0625	0.00	Netral

Gambar 4.47. Hasil anotasi ke - 2 dataset Indonesia menggunakan *TextBlob*

Hasil dari anotasi kedua terhadap dataset bahasa Inggris dapat dilihat pada Gambar 4.47. Anotasi ini menghasilkan pengelompokkan data berdasarkan sentimen yang bersifat netral, positif, dan negatif. Pada proses anotasi kedua, terjadi penurunan jumlah data yang signifikan sebagai akibat dari proses pembersihan data (cleaning). Sebelumnya, dataset bahasa Inggris berisi 1463 data, namun setelah proses pembersihan, jumlah data berkurang menjadi 1141. Proses pembersihan ini termasuk menghapus emotikon yang ada dalam setiap baris data. Oleh karena itu, pada dataset yang telah dibersihkan, tidak lagi ditemukan emotikon, yang menunjukkan bahwa langkah-langkah pembersihan data yang dilakukan telah berhasil menghilangkan elemen-elemen tersebut dari dataset.

Untuk hasil anotasi yang kedua menggunakan *library TextBlob* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.48. Grafik hasil anotasi ke - 2 dataset inggris menggunakan *TextBlob*

Dari grafik hasil anotasi dataset yang pertama pada gambar 4.48 menggunakan *TextBlob*, terlihat bahwa jumlah sentimen netral menurun menjadi 1141, dimana netral sebanyak 583 atau 51.53%, positif sebanyak 350 atau 30.24% dan negatif sebanyak 208 atau 18.23%. Pada anotasi data yang ke – 2 terlihat adanya peningkatan dan penurunan disetiap sentimen. Hal tersebut dapat dilihat pada tabel ini:

Tabel 4.12. Tabel perbandingan perubahan nilai sentimen dari anotasi 1 dan 2 dataset Indonesia menggunakan *TextBlob*

Sentimen	Anotasi ke – 1	Anotasi ke - 2	Perubahan
Netral	61.24% / (896)	51.53% / (583)	Naik (-9,71%) (313)
Positif	23.72% / (347)	30.24% / (350)	Turun (+6,52%) (3)
Negatif	15.04% / (220)	18.23% / (208)	Turun (+3,19%) (-12)
Jumlah data	1463	1141	Turun (322)

Dalam Tabel 4.12 yang membandingkan perubahan nilai sentimen dari dua anotasi dataset Indonesia menggunakan *TextBlob*, terlihat pergeseran yang signifikan pada distribusi sentimen antara anotasi pertama dan kedua. Sentimen netral mengalami penurunan sebesar 9,71% dari 61,24% (896 data) pada anotasi pertama menjadi 51,53% (583 data) pada anotasi kedua, menunjukkan adanya perubahan dalam persepsi atau interpretasi netralitas data. Sentimen positif meningkat sebesar 6,52% dari 23,72% (347 data) pada anotasi pertama menjadi 30,24% (350 data) pada anotasi kedua, yang menunjukkan adanya peningkatan dalam jumlah data yang dianggap memiliki konotasi positif. Sebaliknya, sentimen negatif juga mengalami peningkatan sebesar 3,19% dari 15,04% (220 data) menjadi 18,23% (208 data), mencerminkan adanya peningkatan jumlah data yang dianggap memiliki konotasi negatif. Secara keseluruhan, jumlah data yang dianotasi menurun dari 1463 pada anotasi pertama menjadi 1141 pada anotasi kedua, dengan total penurunan sebesar 322 data. Perubahan ini disebabkan dari proses *cleaning* data, dimana saat proses pembersihan data, terdapat baris kosong sehingga baris tersebut dihapus karena anak mengganggu proses anotasi.

3. Anotasi ke – 3 bahasa inggris menggunakan *TextBlob*

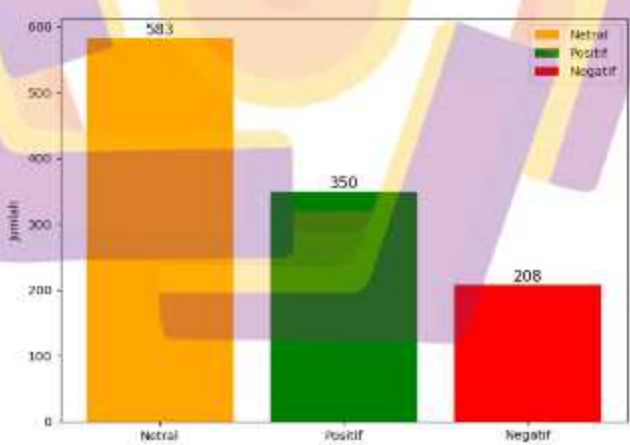
Anotasi data untuk dataset bahasa Indonesia yang ketiga dilakukan dengan menganotasi data yang telah di *cleaning* dan *case folding*.

	Text_Clean	Subjectivity	Polarity	LABTB
0	kisi pat mathematics class semester independen...	0.5625	0.00	Netral
1	want to ask high school seniors who are alre...	0.3325	0.08	Positif
3	list of materials amp notes for class	0.0000	0.00	Netral
4	im really tired of the independent curriculum	0.4125	-0.20	Negatif
5	list of class materials independent curriculum...	0.1250	0.00	Netral
...	...	...	...	...

Gambar 4.49. Hasil anotasi ke - 3 dataset inggris menggunakan *TextBlob*

Dari gambar 4.49 diatas, data yang ditampilkan adalah hasil anotasi ke – 4. Dapat dilihat setelah dilakukan proses pembersihan data dan mengubah data menjadi huruf kecil, maka dapat dilihat bahwa setiap baris data menjadi bersih dan setiap huruf menjadi huruf kecil semua.

Untuk hasil anotasi yang ketiga menggunakan *library TextBlob* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.50. Grafik hasil anotasi ke - 3 dataset inggris menggunakan *TextBlob*

Dari grafik hasil anotasi dataset yang pertama pada gambar 4.50 menggunakan *TextBlob*, terlihat bahwa tidak ada perubahan jumlah sentimen dari anotasi ke - 2. Hal tersebut dapat dilihat pada tabel ini:

Tabel 4.13. Tabel perbandingan perubahan nilai sentimen dari anotasi 2 dan 3 dataset Indonesia menggunakan *TextBlob*

Sentimen	Anotasi ke – 2	Anotasi ke – 3	Perubahan
Netral	51.53% / (583)	51.53% / (583)	Sama (0%) (0)
Positif	30.24% / (350)	30.24% / (350)	Sama (0%) (0)
Negatif	18.23% / (208)	18.23% / (208)	Sama (0%) (0)

Dari tabel 4.13 dapat di lihat bahwa tidak ada perubahan pada nilai Polarity dan Subjectivity dalam analisis sentimen antara anotasi ke-2 dan anotasi ke-3, sehingga hasil anotasi ke – 3 sama dengan hasil anotasi ke - 2. Dari hasil anotasi ke – 3 dapat disimpulkan bahwa penggunaan *case folding* tidak berpengaruh pada anotasi data menggunakan *TextBlob*.

4. Anotasi ke – 4 dataset bahasa inggris menggunakan *TextBlob*

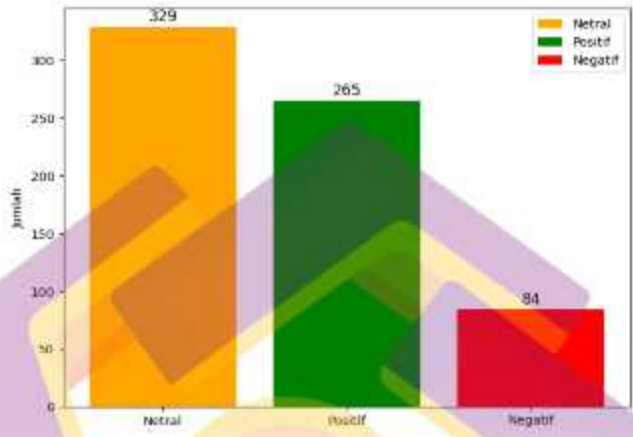
Anotasi data untuk dataset bahasa Indonesia yang ketiga dilakukan dengan menganotasi data yang telah di *cleaning*, *case folding*, dan *Drop duplicate data*. Dengan adanya *drop duplicate data* atau penghapusan data yang sama dan meninggalkan satu data pertama, maka jumlah data pada dataset berkurang menjadi 678 baris data, sehingga akan mempengaruhi jumlah hasil sentimen baik positif, netral maupun negatif.

	Text_Clean	Subjectivity	Polarity	LABTB
0	kisi pat mathematics class semester independen...	0.5625	0.00	Netral
1	want to ask high school seniors who are alre...	0.3325	0.08	Positif
3	list of materials amp notes for class	0.0000	0.00	Netral
4	im really tired of the independent curriculum	0.4125	-0.20	Negatif
5	list of class materials independent curriculum...	0.1250	0.00	Netral
...	...	...	...	...
1456	definitely physics the same as geo but only ...	0.4375	0.00	Netral
1457	just read any independent curriculum	0.1250	0.00	Netral
1459	work title and art performance implementing th...	0.0000	0.00	Netral
1461	class high school independent curriculum book	0.3325	0.08	Positif
1462	independent curriculum book for grade middle s...	0.0625	0.00	Netral

Gambar 4.51. Hasil anotasi ke - 4 dataset inggris menggunakan *TextBlob*

Dari Gambar 4.51 tersebut, tampak bahwa data yang ditampilkan merupakan hasil anotasi keempat, yang mencakup data sentimen netral, positif, dan negatif dari total 678 baris data setelah proses penghapusan duplikasi (drop duplicate). Proses anotasi pada data keempat ini dilakukan tanpa mengubah makna kalimat dalam setiap baris data. Seperti halnya pada anotasi sebelumnya, yaitu anotasi pertama, kedua, dan ketiga, variasi dalam tingkat subjektivitas dan polaritas terhadap kurikulum merdeka tetap terlihat. Hal ini terlihat dari nilai-nilai sentimen yang bervariasi, mulai dari yang bersifat positif, netral, hingga negatif. Dengan kata lain, meskipun proses anotasi kali ini dilakukan dengan menjaga keutuhan makna kalimat, tetap terlihat perbedaan dalam bagaimana berbagai baris data merespons dan menilai kurikulum merdeka, mencerminkan berbagai perspektif dan opini yang ada. Hal ini menunjukkan bahwa opini terkait kurikulum merdeka masih beragam dan dipengaruhi oleh sudut pandang masing-masing responden.

Untuk hasil anotasi yang keempat menggunakan *library TextBlob* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.52. Grafik hasil anotasi ke - 4 dataset inggris menggunakan *TextBlob*

Dari grafik hasil anotasi dataset pertama yang ditunjukkan pada Gambar 4.52 menggunakan *TextBlob*, dapat dilihat bahwa total jumlah data yang dianalisis adalah sebanyak 678 baris. Dari jumlah tersebut, sentimen netral mendominasi dengan 329 baris atau 48,53% dari keseluruhan data. Sentimen positif juga cukup signifikan, tercatat sebanyak 265 baris atau 39,09%, sementara sentimen negatif lebih sedikit, dengan 84 baris atau 12,38%. Perbedaan hasil pada anotasi data yang keempat dibandingkan dengan anotasi pertama, kedua, dan ketiga terletak pada jumlah baris data yang berbeda. Hal ini mempengaruhi distribusi sentimen yang tercatat. Perbedaan ini dapat lebih jelas dipahami melalui tabel dibawah ini yang menyajikan rincian jumlah baris data dalam setiap anotasi, yang menunjukkan variasi antara dataset anotasi keempat dan dataset pada anotasi sebelumnya.

Tabel 4.14. Tabel perbandingan perubahan nilai sentimen dari anotasi 1, 2, 3, dan 4 dataset inggris menggunakan *TextBlob*

Sentimen	**Anotasi ke – 1**	**Anotasi ke – 2**	**Anotasi ke – 3**	**Anotasi ke – 4**
Netral	61.24% / (896)	51.53% / (583)	51.53% / (583)	48.53% / (329)
Positif	23.72% / (347)	30.24% / (350)	30.24% / (350)	39.09% / (265)
Negatif	15.04% / (220)	18.23% / (208)	18.23% / (208)	12.38%/ (84)
Jumlah Data	1463	1141	1141	678

Dari data pada tabel 4.14, terlihat adanya pola perubahan signifikan dalam distribusi sentimen dari anotasi pertama hingga anotasi keempat. Pada awalnya, sentimen netral mendominasi dengan persentase sebesar 61.24% dari total 1463 data pada anotasi pertama. Namun, persentase ini menunjukkan penurunan konsisten pada anotasi berikutnya, berkurang menjadi 51.53% pada anotasi kedua dan ketiga, dan akhirnya turun menjadi 48.53% pada anotasi keempat dengan total data yang juga berkurang menjadi 678. Penurunan ini mungkin mencerminkan peningkatan pemahaman atau perubahan perspektif terhadap data seiring dengan berjalannya waktu atau perubahan metodologi anotasi.

Sebaliknya, sentimen positif menunjukkan tren peningkatan. Dari awalnya 23.72% pada anotasi pertama, persentase ini meningkat menjadi 30.24% pada anotasi kedua dan ketiga, dan akhirnya mencapai 39.09% pada anotasi keempat. Ini mungkin menunjukkan adanya pergeseran dalam penilaian atau interpretasi yang lebih optimis terhadap data yang sama pada periode yang berbeda. Sentimen negatif

mengalami fluktuasi yang lebih kompleks. Meskipun mengalami peningkatan dari 15.04% pada anotasi pertama menjadi 18.23% pada anotasi kedua dan ketiga, persentase ini akhirnya turun menjadi 12.38% pada anotasi keempat. Ini menunjukkan adanya variasi dalam interpretasi negatif, yang bisa jadi disebabkan oleh perubahan konteks atau kriteria penilaian yang diterapkan dari *TextBlob*.

Secara keseluruhan, data menunjukkan adanya dinamika yang signifikan dalam distribusi sentimen dari anotasi 1 sampai 4. Perubahan ini dapat diakibatkan oleh beberapa faktor, diantaranya perubahan dalam metode anotasi, perubahan jumlah data, atau penghapusan data yang sama. Mengingat jumlah data yang juga menurun dari anotasi pertama hingga keempat, penting untuk mempertimbangkan bahwa variasi dalam jumlah data dapat mempengaruhi distribusi sentimen yang diamati.

Tabel 4.15. Tabel perubahan nilai subjektivitas dan polaritas menggunakan 1 sampel pada anotasi 1 dan 2 dataset inggris menggunakan *TextBlob*

Baris	Tweet	Subjectivity	Polarity	LABTB	Anotasi
790	RT @studjellyca: [Grade 10 History Notes] —Free Curriculum Forced Planting (Cultuurstelsel) https://t.co/AL1dSaEQLE	1.0	-0.75	Negatif	Anotasi ke - 1
790	Grade History Notes Free Curriculum Forced Planting Cultuurstelsel	0.5	0.04999999	Positif	Anotasi ke - 2
790	grade history notes free curriculum forced planting cultuurstelsel	0.5	0.04999999	Positif	Anotasi ke – 3
790	grade history notes free curriculum forced planting cultuurstelsel	0.5	0.04999999	Positif	Anotasi ke – 4

Hasil anotasi pada tabel 4.15 menunjukkan transformasi nilai subjektivitas dan polaritas dari anotasi pertama hingga anotasi keempat pada dataset bahasa Inggris menggunakan *library TextBlob*. Pada anotasi pertama, yang merupakan data mentah, tweet ini diberi subjektivitas maksimum (1.0) dan polaritas yang sangat negatif (-0.75), dengan label sentimen "Negatif". Namun, pada anotasi kedua, yang telah menghapus emotikon dari teks, subjektivitas menurun menjadi 0.5 dan polaritas mengalami perubahan menjadi positif (0.05), yang seiring dengan perubahan label sentimen menjadi "Positif".

Anotasi ketiga melibatkan proses case folding, yang menghasilkan perubahan minor dalam subjektivitas dan polaritas yang tetap pada 0.5 dan 0.05 secara berturut-turut, dengan label sentimen yang juga tetap "Positif". Pada anotasi keempat, yang mencakup langkah-langkah dari anotasi kedua dan ketiga serta penghapusan data duplikat, subjektivitas dan polaritas tetap konsisten pada nilai 0.5 dan 0.05. Hal ini menunjukkan bahwa transformasi data lebih lanjut, seperti case folding dan penghapusan duplikat, tidak mengubah substansial nilai subjektivitas dan polaritas yang telah disesuaikan pada langkah sebelumnya. Namun untuk penghapusan duplikat data berpengaruh terhadap jumlah data yang akan di anotasi, sehingga ada perbedaan jumlah sentimen dari anotasi 1, 2, dan 3 terhadap jumlah sentimen pada anotasi data ke – 4.

Analisis ini menunjukkan bahwa proses anotasi data, terutama dalam analisis sentimen dengan library seperti *TextBlob*, sangat dipengaruhi oleh pemrosesan awal seperti penghapusan emotikon dan case folding. Meski

transformasi ini bisa mengubah label sentimen dari negatif menjadi positif, nilai subjektivitas dan polaritas tweet tetap stabil setelah pemrosesan awal.

b. Anotasi dataset bahasa Indonesia menggunakan *VADER*

Pada proses anotasi dataset bahasa inggris menggunakan *VADER*, dilakukan empat kali anotasi data, antara lain:

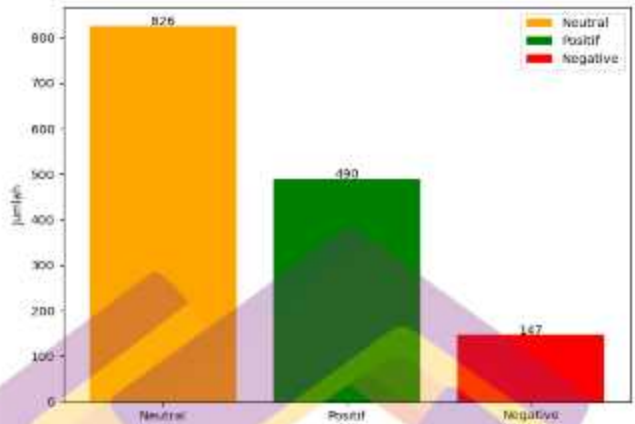
1. Anotasi ke – 1 dataset bahasa inggris menggunakan *VADER*

Anotasi data untuk dataset bahasa inggris yang pertama dilakukan adalah menganotasi data mentah yang belum di *cleaning*.

	Tweet	Compound_Score	Neg	Neu	Pos	LABVDR
0	KISI PAT Mathematics Class 4 Semester 2 indepe...	0.0000	0.000	1.000	0.000	Netral
1	I want to ask high school seniors who are alre...	0.5818	0.000	0.825	0.175	Positif
2	RT @DEFINEGRADES:	0.0000	0.000	1.000	0.000	Netral
3	RT @sarikushi: List of materials & notes fo...	0.0000	0.000	1.000	0.000	Netral
4	I'm really tired of the independent curriculum 😭	-0.4391	0.278	0.609	0.113	Negatif
...	...	...	...	...	...	...
1458	RT @k___ale: list of class 10 materials (indep...	0.0000	0.000	1.000	0.000	Netral
1459	Work title and art performance implementing th...	0.4939	0.000	0.887	0.113	Positif

Gambar 4.53. Hasil anotasi data ke - 1 dataset inggris menggunakan *VADER*

Data pada tabel 4.53 merupakan hasil anotasi data pertama dari 1463 baris data yang telah dianotasi menggunakan *VADER*. Proses anotasi yang pertama menggunakan data mentah yang belum di lakukan *cleaning* data. Dari tabel tersebut terlihat adanya variasi nilai sentimen yang bersifat netral, positif dan negatif. Untuk variasi dari jumlah masing-masing sentimen yang dihasilkan dapat dilihat dari grafik dibawah ini:



Gambar 4.54. Grafik hasil anotasi ke - 1 dataset inggris menggunakan *VADER*

Dari grafik pada gambar 4.54 terlihat bahwa hasil anotasi dataset bahasa inggris yang pertama menggunakan *VADER* menghasilkan jumlah sentimen netral sebanyak 826 atau 56.49%, positif sebanyak 490 atau 33.47%, dan negatif sebanyak 147 atau 10.04%. Hasil ini juga dipengaruhi oleh adanya emotikon dalam setiap baris data, dimana *VADER* merupakan anator yang dapat membaca emotikon sebagai sebuah emosi.

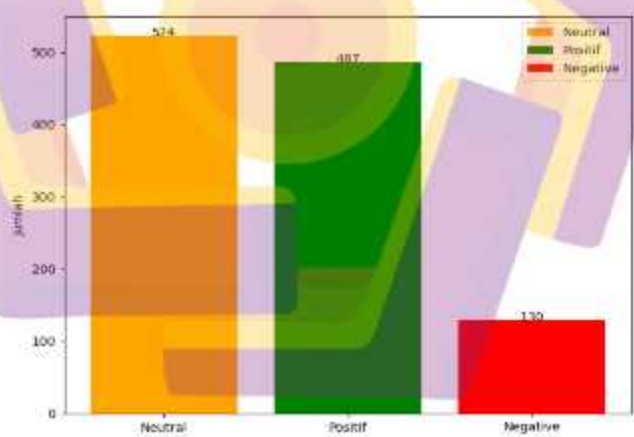
2. Anotasi ke – 2 dataset bahasa inggris menggunakan *VADER*

Anotasi data untuk dataset bahasa inggris yang kedua dilakukan dengan menganotasi data yang telah di *cleaning* namun tidak dilakukan *case folding*.

Text_Clean	Compound_Score	Neg	Neu	Pos	LABVDR
KISI PAT Mathematics Class Semester Independen...	0.0000	0.000	1.000	0.000	Netral
want to ask high school seniors who are alre...	0.3818	0.000	0.816	0.184	Positif
List of materials amp notes for class	0.0000	0.000	1.000	0.000	Netral
Im really tired of the independent curriculum	-0.4927	0.347	0.653	0.000	Negatif
list of class materials independent curriculum...	0.0000	0.000	1.000	0.000	Netral

Gambar 4.55. Hasil anotasi ke - 2 dataset inggris menggunakan *VADER*

Dari Gambar 4.55 di atas, tampak hasil anotasi kedua dari dataset bahasa Inggris menggunakan *VADER*. Dalam proses ini, beberapa perubahan telah diterapkan pada setiap baris kalimat, di antaranya adalah penghapusan emotikon, retweet, serta angka. Meskipun adanya perubahan format tersebut, makna dari setiap baris data tetap tidak berubah. Selain itu, jumlah data juga mengalami pengurangan karena selama proses pembersihan data, baris-baris kosong ditemukan dan dihapus. Penghapusan baris kosong ini dilakukan untuk memastikan bahwa proses analisis anotasi dapat dilanjutkan dengan data yang bersih dan konsisten. Hasil dari anotasi kedua yang menggunakan library *VADER* dapat dilihat lebih lanjut pada grafik yang disajikan di bawah ini, yang menggambarkan distribusi sentimen dari data yang telah dianalisis.



Gambar 4.56. Grafik hasil anotasi ke - 2 dataset inggris menggunakan *VADER*

Dari grafik hasil anotasi dataset yang kedua pada gambar 4.56 menggunakan *VADER*, terlihat bahwa jumlah data mengalami penurunan karena adanya baris kosong yang dihapus setelah proses pembersihan data. Dari grafik diatas dapat

dilihat bahwa hasil anotasi ke – 2 menghasilkan sentimen netral sebanyak 524 atau 45.92%, positif sebanyak 487 atau 42.71%, dan negatif sebanyak 130 atau 11.37%. Hal tersebut dapat dilihat pada tabel berikut ini:

Tabel 4.16. Tabel perbandingan perubahan nilai sentimen dari anotasi 1 dan 2 dataset inggris menggunakan *VADER*

Sentimen	Anotasi ke – 1	Anotasi ke - 2	Perubahan
Netral	56.49% / (826)	45.92% / (524)	Turun (-10,57%) (304)
Positif	33.47% / (490)	42.71% / (487)	Naik (+9,24%) (-3)
Negatif	10.04% / (147)	11.37% / (130)	Naik (+1,33%) (-17)
Jumlah data	1463	1141	Turun (322)

Data dari Tabel 4.16 memperlihatkan perubahan nilai sentimen antara dua anotasi pada dataset bahasa Inggris menggunakan metode *VADER*. Pada anotasi pertama, persentase sentimen netral adalah 56.49%, yang kemudian turun menjadi 45.92% pada anotasi kedua. Hal ini menunjukkan penurunan signifikan sebesar 10.57% dalam kategori sentimen netral, menghasilkan perbedaan absolut sebesar 304 data. Sebaliknya, sentimen positif mengalami kenaikan dari 33.47% pada anotasi pertama menjadi 42.71% pada anotasi kedua, menunjukkan kenaikan sebesar 9.24%, dengan perbedaan hanya 3 data. Sentimen negatif juga mengalami peningkatan dari 10.04% menjadi 11.37%, naik sebesar 1.33% atau 17 data.

Perubahan ini menggambarkan dinamika dalam analisis sentimen yang dipengaruhi oleh metodologi anotasi dan algoritma seperti *VADER*. Penurunan signifikan dalam sentimen netral dapat mengindikasikan pergeseran dalam persepsi atau penilaian terhadap data, sedangkan peningkatan dalam sentimen positif dan negatif bisa mencerminkan sensitivitas yang lebih baik dari algoritma terhadap

variasi dalam ekspresi sentimen dalam teks. Penurunan jumlah total data yang dianotasi dari 1463 menjadi 1141 juga menunjukkan perubahan dalam skala atau fokus analisis, yang perlu dipertimbangkan dalam interpretasi akhir terhadap hasil analisis sentimen.

3. Anotasi ke – 3 dataset bahasa inggris menggunakan *VADER*

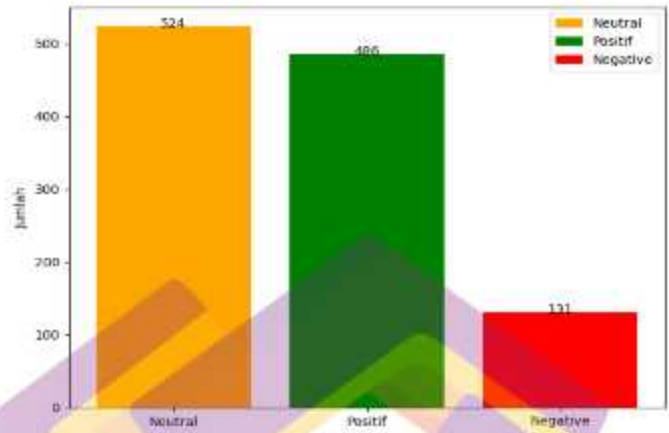
Anotasi data untuk dataset bahasa Indonesia yang ketiga dilakukan dengan menganotasi data yang telah di *cleaning* dan *case folding*.

Text_Clean	Compound_Score	Neg	Neu	Pos	LABVDR
kbsi pat mathematics class semester independen...	0.0000	0.000	1.000	0.000	Netral
want to ask high school seniors who are alre...	0.3818	0.000	0.816	0.184	Positif
list of materials amp notes for class	0.0000	0.000	1.000	0.000	Netral
im really tired of the independent curriculum	-0.4027	0.347	0.653	0.000	Negatif
list of class materials independent curriculum...	0.0000	0.000	1.000	0.000	Netral
...	...	...	...	...	...
just read any independent curriculum	0.0000	0.000	1.000	0.000	Netral
list of class materials independent curriculum...	0.0000	0.000	1.000	0.000	Netral
work title and art performance implementing th...	0.4939	0.000	0.873	0.127	Positif
class high school independent curriculum book	0.0000	0.000	1.000	0.000	Netral
independent curriculum book for grade middle s...	0.0000	0.000	1.000	0.000	Netral

Gambar 4.57. Hasil anotasi ke - 3 dataset inggris menggunakan *VADER*

Gambar 4.57 merupakan hasil anoatasi data yang ketiga, sebelum proses anotasi yang pertama dilakukan adalah *cleaning* dan *case folding*. Untuk hasil sentimen anotasi ke – 3 mengalami sedikit perubahan dari anotasi ke – 2..

Untuk hasil anotasi yang ketiga menggunakan *library VADER* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.58. Grafik hasil anotasi ke - 3 dataset inggris menggunakan *VADER*

Dari grafik hasil anotasi dataset yang pertama pada gambar 4.58 menggunakan *VADER*, terlihat bahwa jumlah sentimen netral tetap, namun ada perubahan kecil pada sentimen positif dan negatif, dimana ada penurunan di sentimen positif sebanyak 1 baris data menjadi negatif, sehingga jumlah data sentimen positif menjadi 486 dan negatif menjadi 131. Hal tersebut dapat dilihat pada tabel ini:

Tabel 4.17. Tabel perbandingan perubahan nilai sentimen dari anotasi 2 dan 3 dataset inggris menggunakan *VADER*

Sentimen	Anotasi ke – 2	Anotasi ke – 3	Perubahan
Netral	45.92% / (524)	45.92% / (524)	Naik (0%) (0)
Positif	42.71% / (487)	42.64% / (486)	Turun (-0,07%) (-1)
Negatif	11.37% / (130)	11.43% / (131)	Naik (+0,06) (1)

Tabel 4.17 menunjukkan perubahan nilai sentimen dari anotasi ke-2 dan ke-3 menggunakan algoritma *VADER* pada dataset bahasa Inggris. Dalam konteks ini,

perubahan-perubahan tersebut memberikan wawasan tentang dinamika dalam analisis sentimen menggunakan metode yang berbeda.

Pertama, untuk sentimen netral, terlihat bahwa persentase tetap stabil pada 45.92% dari total data pada kedua anotasi. Meskipun persentase tetap, hal ini mungkin menunjukkan konsistensi dalam penilaian netral terhadap *tweet* yang dianalisis. Kedua, untuk sentimen positif, terjadi penurunan yang kecil dari 42.71% pada anotasi ke-2 menjadi 42.64% pada anotasi ke-3. Meskipun penurunan ini hanya sebesar 0.07%, hal ini menunjukkan adanya variasi dalam penilaian positif terhadap *tweet* yang mungkin disebabkan oleh sensitivitas algoritma atau variasi dalam interpretasi teks.

Ketiga, untuk sentimen negatif, terlihat kenaikan dari 11.37% pada anotasi ke-2 menjadi 11.43% pada anotasi ke-3. Meskipun kenaikan ini juga kecil (0.06%), hal ini menunjukkan pergeseran kecil dalam penilaian negatif terhadap *tweet* yang dianalisis.

Dari hasil tersebut, dapat disimpulkan bahwa *cleaning* dan *case folding* mempunyai pengaruh terhadap pergeseran data. Walaupun kecil, akan tetapi hasil tersebut mempengaruhi hasil anotasi dan yang mungkin akan mempengaruhi hasil dari model mesin yang akan dibangun nantinya.

4. Anotasi ke – 4 dataset bahasa inggris menggunakan *VADER*

Anotasi data untuk dataset bahasa inggris yang keempat dilakukan dengan menganotasi data yang telah di *cleaning*, *case folding*, dan *Drop duplicate data*. Dengan adanya *drop duplicate data* atau penghapusan data yang sama dan meninggalkan satu data pertama, maka jumlah data pada dataset berkurang menjadi

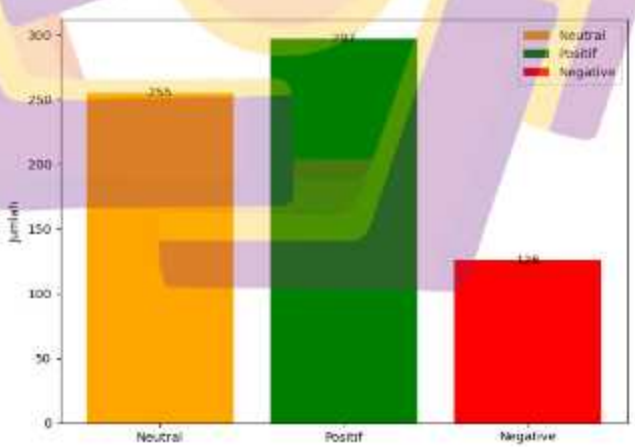
678 baris data, sehingga akan mempengaruhi jumlah hasil sentimen baik positif, netral maupun negatif.

Text_Clean	Compound_Score	Neg	Neu	Pos	LABVDR
kisi pat mathematics class semester independen...	0.0000	0.000	1.000	0.000	Netral
want to ask high school seniors who are alre...	0.3818	0.000	0.816	0.184	Positif
list of materials amp notes for class	0.0000	0.000	1.000	0.000	Netral
im really tired of the independent curriculum	-0.4927	0.347	0.653	0.000	Negatif
list of class materials independent curriculum...	0.0000	0.000	1.000	0.000	Netral
...	...	...	...	...	...
definitely physics the same as gest but only...	0.8249	0.000	0.758	0.242	Positif
just read any independent curriculum	0.0000	0.000	1.000	0.000	Netral

Gambar 4.59. Hasil anotasi ke - 4 dataset Indonesia menggunakan *VADER*

Dari gambar 4.59 diatas dapat dilihat bahwa data hasil anotasi keempat menghasilkan pengelompokan sentimen yang variatif yaitu sentimen netral, positif dan negatif dari 678 baris data yang telah dilakukan *drop duplicate data*.

Untuk hasil anotasi yang keempat menggunakan *library VADER* dapat dilihat pada gambar grafik dibawah ini:



Gambar 4.60. Grafik hasil anotasi ke - 4 dataset inggris menggunakan *VADER*

Dapat dilihat pada Gambar 4.60 bahwa hasil anotasi pada tahap keempat menunjukkan perbedaan yang signifikan dibandingkan dengan hasil dari anotasi pertama, kedua, dan ketiga. Pada anotasi keempat, tampak bahwa sentimen positif mendominasi lebih banyak dibandingkan dengan sentimen netral dan negatif. Perubahan ini kemungkinan disebabkan oleh proses cleaning data, case folding, dan penghapusan data duplikat (drop duplicate). Dalam proses ini, data duplikat yang mungkin paling banyak terdapat pada kategori sentimen netral telah dihapus, yang berakibat pada penurunan signifikan pada proporsi sentimen netral dalam dataset. Untuk perbedaan ini, dapat dilihat pada tabel dibawah ini:

Tabel 4.18. Tabel perbandingan perubahan nilai sentimen dari anotasi 1, 2, 3, dan 4 dataset inggris menggunakan *VADER*

Sentimen	**Anotasi ke – 1**	**Anotasi ke – 2**	**Anotasi ke – 3**	**Anotasi ke – 4**
Netral	56.49% / (826)	45.92% / (524)	45.92% / (524)	37.61% / (255)
Positif	33.47% / (490)	42.71% / (487)	42.64% / (486)	43.78% / (297)
Negatif	10.04% / (147)	11.37% / (130)	11.43% / (131)	18.61% / (126)
Jumlah Data	1463	1141	1141	678

Tabel 4.18 menggambarkan perubahan nilai sentimen dari empat tahap anotasi yang berbeda menggunakan *VADER* pada dataset bahasa Inggris. Pertama, terlihat adanya penurunan yang signifikan dalam persentase sentimen netral dari 56.49% pada anotasi pertama menjadi 37.61% pada anotasi keempat. Penurunan ini menandai pergeseran yang tajam dalam cara *tweet* yang dianalisis dinilai netral,

yang mungkin disebabkan oleh proses *cleaning*, *case folding*, dan *drop duplicate data*.

Kedua, terdapat kenaikan yang stabil dalam sentimen positif, dari 33.47% pada anotasi pertama menjadi 43.78% pada anotasi keempat. Peningkatan ini menunjukkan bahwa algoritma *VADER* mampu mengenali lebih banyak *tweet* yang mengandung sentimen positif dalam dataset atas perubahan dalam cara anotasi dilakukan.

Ketiga, meskipun sentimen negatif tetap relatif stabil dari anotasi kedua hingga ketiga, terdapat lonjakan signifikan pada anotasi keempat. Persentase sentimen negatif meningkat dari 11,37% pada anotasi ketiga menjadi 18,61% pada anotasi keempat. Peningkatan yang mencolok ini dapat mengindikasikan adanya perubahan substansial dalam dataset antara ketiga dan keempat, yang kemungkinan menyebabkan lebih banyak tweet dinilai sebagai negatif pada tahap anotasi terakhir. Perubahan ini mungkin mencerminkan variasi dalam jenis atau konten tweet yang dianalisis, yang mengarah pada penilaian sentimen yang lebih negatif pada data terbaru.

Secara keseluruhan, perubahan-perubahan ini menyoroti pentingnya proses anotasi yang konsisten dan sensitif terhadap dinamika dalam data teks. Interpretasi sentimen yang akurat memerlukan pemahaman yang mendalam terhadap faktor-faktor yang dapat mempengaruhi penilaian sentimen, seperti penerapan proses *cleaning*, *case folding*, dan *Drop duplicate data* atau perubahan dalam teknik anotasi yang diterapkan. Dengan demikian, tabel ini memberikan gambaran yang komprehensif tentang kompleksitas dalam analisis sentimen berbasis teks dan

perubahan yang dapat terjadi seiring dengan perubahan proses analisis dalam anotasi.

Tabel 4.19. Tabel perubahan nilai *compound* menggunakan 1 sampel dari anotasi 1, 2, 3, dan 4 pada dataset inggris menggunakan *VADER*

Baris	Tweet	Compound Score	Neg	Neu	Pos	LAB VDR	Anotasi
715	As an independent curriculum student, I admit that I'm really tired 🙂😑😡 BUT PLS GWEH I...	-0.5462	0.226	0.668	0.1	Negatif	Anotasi ke - 1
715	As an independent curriculum student admit that Im really tired BUT PLS GWEH DONT WAN...	0.0263	0.15	0.68	0.15	Positif	Anotasi ke - 2
715	as an independent curriculum student admit that im really tired but pls gweh dont want...	-0.0474	0.13	0.74	0.12	Negatif	Anotasi ke - 3
715	as an independent curriculum student admit that im really tired but pls gweh dont want...	-0.0474	0.13	0.74	0.12	Negatif	Anotasi ke - 4

Tabel 4.19 menampilkan perubahan nilai *Compound Score* dan komponen-komponen sentimen (Negatif, Netral, Positif) dari empat tahap anotasi yang berbeda menggunakan algoritma *VADER* pada dataset bahasa Inggris. Analisis ini memberikan wawasan tentang bagaimana interpretasi sentimen terhadap sebuah *tweet* dapat bervariasi tergantung pada proses anotasi yang dilakukan.

Pertama, *tweet* pada Anotasi ke-1 dengan *Compound Score* -0.5462, yang diklasifikasikan sebagai "Negatif" oleh algoritma *VADER*. Nilai Negatif (0.226) cukup tinggi dibandingkan dengan Neu (0.668) dan Pos (0.106), menunjukkan bahwa *tweet* ini mengandung banyak kata-kata dengan sentimen negatif. *Compound Score* negatif mengindikasikan bahwa *tweet* ini secara keseluruhan dianggap memiliki sentimen negatif yang dominan.

Kedua, pada Anotasi ke-2, *tweet* yang sama memiliki *Compound Score* 0.0263, diklasifikasikan sebagai "Positif". Nilai Positif (0.157) lebih tinggi dibandingkan dengan Negatif (0.154) dan Neu (0.689). Ini menunjukkan bahwa setelah proses anotasi yang mungkin mencakup penghapusan emotikon dan case folding, interpretasi *tweet* berubah menjadi lebih positif. *Compound Score* yang rendah (meskipun positif) menandakan bahwa *tweet* ini mungkin masih netral secara keseluruhan, tetapi memiliki sedikit kecenderungan positif.

Ketiga, pada Anotasi ke-3 dan ke-4, *tweet* yang sama memiliki *Compound Score* -0.0474, dengan klasifikasi "Negatif". Nilai Negatif, Neu, dan Pos hampir identik di kedua anotasi ini (0.134, 0.74, 0.126), menunjukkan bahwa interpretasi sentimen terhadap *tweet* ini konsisten sebagai negatif. *Compound Score* yang mendekati nol menandakan bahwa *tweet* ini cenderung netral, tetapi dengan sedikit kecenderungan negatif yang lebih kuat pada anotasi ini.

Secara keseluruhan, perubahan dalam nilai *Compound Score* dari anotasi ke anotasi mencerminkan perubahan dalam interpretasi sentimen *tweet* yang terjadi seiring dengan proses anotasi yang berbeda. Komponen-komponen sentimen (Negatif, Netral, Positif) dan perbandingan antara komponen membantu menggambarkan bagaimana sebuah *tweet* dapat dinilai berdasarkan kata-kata dengan sentimen yang dominan. Nilai *Compound Score* yang mendekati nol menunjukkan bahwa *tweet* mungkin memiliki campuran sentimen atau cenderung netral, sedangkan nilai yang jauh dari nol menunjukkan kecenderungan yang lebih kuat ke arah positif atau negatif. Interpretasi ini menjadi landasan penting dalam

memahami apakah sebuah teks menyampaikan fakta, opini, atau bahkan sentimen emosional dari pengguna.

c. Analasisis perbandingan hasil Anotasi bahasa Indonesia menggunakan *TextBlob* dan *VADER*

Perbandingan hasil analisis dari *library TextBlob* dan *VADER* terhadap dataset bahasa Indonesia dilakukan dengan mengambil satu sampel atau satu baris data yang sama dari setiap anotasi yang telah dilakukan. Sampel tersebut akan menjadi bahan analisis dengan tujuan agar dapat melihat perubahan pola polaritas dan subjektivitas pada *TextBlob* dan perubahan pola *compound score* dari *VADER*.

Tabel 4.20. Tabel perbandingan hasil anotasi dataset inggris pada 1 sampel baris data menggunakan *VADER* dan *TextBlob*

Baris	Tweet	Compound Score	LAB VDR	Subjectivity	Polarity	LABTB	Anotasi
715	As an independent curriculum student, I admit that I'm really tired 😢😐😫 BUT PLS GWEH I DON'T WANT THE 2013 CURRICULUM YG IPA IPS PLSS NA... https	-0.5462	Negatif	0.4125	-0.2	Negatif	Anotasi ke - 1
715	As an independent curriculum student admit that Im really tired BUT PLS GWEH DONT WANT THE CURRICULUM	0.0263	Positif	0.4125	-0.2	Negatif	Anotasi ke - 2

Tabel 4.20. (Lanjutan)

	YG IPA IPS PLSS NA https						
715	as an independent curriculum student admit that im really tired but pls gweh dont want the curriculum yg ipa ips plss na https	-0.0474	Negatif	0.4125	-0.2	Negatif	Anotasi ke - 3
353	as an independent curriculum student admit that im really tired but pls gweh dont want the curriculum yg ipa ips plss na https	-0.0474	Negatif	0.4125	-0.2	Negatif	Anotasi ke - 4

Dalam tabel 4.20, terlihat perbandingan hasil anotasi antara algoritma *VADER* dan *TextBlob* pada dataset bahasa Inggris untuk beberapa tweet yang sama. *VADER* menggunakan pendekatan lexicon-based yang memperhitungkan intensitas kata-kata dalam menentukan sentimen, sementara *TextBlob* mengandalkan analisis berbasis aturan untuk menentukan subjektivitas dan polaritas teks.

Pertama, pada tweet dengan indeks 715, *VADER* menghasilkan nilai Compound Score -0.5462, menandakan sentimen negatif yang dominan. Nilai komponen sentimen Negatif, Netral, dan Positif yang spesifik (0.226, 0.668, 0.106) juga mencerminkan penilaian yang jelas terhadap sentimen tersebut. Di sisi lain, *TextBlob* memberi nilai subjektivitas 0.4125 dan polaritas -0.2, yang menunjukkan bahwa teks tersebut subjektif namun dengan kecenderungan negatif yang kurang kuat dibandingkan dengan *VADER*. Perbedaan ini mungkin disebabkan oleh cara

masing-masing algoritma menafsirkan kata-kata tertentu dan intensitasnya dalam konteks sentimen.

Kedua, pada tweet dengan anotasi ke-2, *VADER* menunjukkan nilai Compound Score 0.0263 yang menunjukkan sentimen positif, dengan komponen sentimen yang mendukung (Negatif: 0.154, Netral: 0.689, Positif: 0.157). Namun, *TextBlob* tetap menilai teks ini sebagai negatif dengan nilai subjektivitas dan polaritas yang sama (0.4125 dan -0.2). Perbedaan ini menggambarkan bahwa *VADER* lebih sensitif terhadap perbedaan dalam intensitas kata-kata positif dalam teks, sementara *TextBlob* lebih mendasarkan penilaiannya pada aturan gramatikal dan leksikal.

Ketiga, pada tweet dengan anotasi ke-3 dan ke-4, *VADER* dan *TextBlob* konsisten dalam menilai tweet tersebut sebagai negatif, meskipun dengan perbedaan minor dalam nilai Compound Score dan nilai subjektivitas/polaritas. Ini menunjukkan konsistensi dalam interpretasi teks yang lebih dominan dalam sentimen negatif oleh kedua algoritma, meskipun pendekatannya dalam menentukan sentimen bisa berbeda.

Secara keseluruhan, perbandingan antara *VADER* dan *TextBlob* dalam tabel ini menyoroti perbedaan pendekatan dalam analisis sentimen. *VADER* cenderung lebih baik dalam mengenali intensitas sentimen berdasarkan kata-kata tertentu, sementara *TextBlob* lebih fokus pada aturan linguistik untuk menentukan polaritas dan subjektivitas teks. Memahami perbedaan ini penting untuk memilih algoritma yang sesuai dengan kebutuhan analisis sentimen berdasarkan konteks dan tujuan spesifik.

4.5.3. Validasi hasil Anotasi menggunakan *Naive Bayes* dan SVM

4.5.3.1 Validasi menggunakan *Naive Bayes*

Pada analisis sentimen menggunakan *Naive Bayes* pada dataset bahasa Indonesia yang telah di *preprocessing*, dapat dilihat pada *script* dibawah ini:

```
# proses TI - IDF
tf = TfidfVectorizer()
data = tf.fit_transform(df['Tweets'].astype('U'))
```

Gambar 4.61. Proses pembobotan TF-IDF

Fungsi dari *script* diatas menggunakan `TfidfVectorizer` dari *library* `scikit-learn` untuk mengubah teks dalam kolom `*Tweets*` dari *DataFrame* `df` menjadi representasi vektor berdasarkan nilai TF-IDF (Term Frequency-Inverse Document Frequency).

Untuk pembagian data uji dan latih menggunakan *script* dibawah ini:

```
#splitting data
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(x_tfidf,df['LABVDR'],
                                                    test_size =0.2)
```

Gambar 4.62. Proses *spitting* data uji dan data latih (NB)

Fungsi di atas digunakan untuk membagi dataset menjadi data pelatihan (training) dan data pengujian (testing) dengan menggunakan fungsi 'train_test_split' dari *library* 'scikit-learn'.

Pertama, *library* 'scikit-learn' diimpor, khususnya modul 'train_test_split' dari 'sklearn.model_selection', yang merupakan fungsi yang digunakan untuk membagi dataset. Selanjutnya, fungsi 'train_test_split' digunakan untuk membagi

data. Fungsi ini menerima beberapa parameter: data fitur (x_tfidf), data target (df['LABVDR']), dan parameter test_size yang menentukan proporsi data yang akan digunakan sebagai data pengujian. Dalam hal ini, 'test_size=0.2' berarti 20% dari data akan digunakan sebagai data pengujian, sedangkan 80% sisanya akan digunakan sebagai data pelatihan. Parameter pertama, 'x_tfidf', adalah data fitur yang telah ditransformasikan menjadi representasi TF-IDF. Parameter kedua, df['LABVDR'], adalah data target yang berisi label sentimen.

Dengan membagi dataset menjadi data latih dan data uji, maka model dapat dilatih pada data latih dan menguji kinerjanya pada data uji yang terpisah, yang membantu dalam mengevaluasi seberapa baik model akan bekerja pada data yang belum pernah dilihat sebelumnya.

Klasifikasi *Naive Bayes* menggunakan *script* dibawah ini:

```
#klasifikasi
klas = MultinomialNB().fit(X_train, y_train)
predicted = klas.predict(X_test)
print("MultinomialNB Accuracy: ", accuracy_score(y_test,predicted))
print("MultinomialNB Precision: ", precision_score(y_test,predicted, average='weighted', pos_label="Positif"))
print("MultinomialNB Recall: ", recall_score(y_test,predicted, average='weighted', pos_label="Positif"))
print("MultinomialNB f1_score: ", f1_score(y_test,predicted, average='weighted', pos_label="Positif"))

print(f'Confusion matrix:\n{confusion_matrix(y_test,predicted)}')
print('\n')
print(classification_report(y_test,predicted,zero_division=0))
```

Gambar 4.63. Klasifikasi *Naive Bayes*

Fungsi ini menggunakan model klasifikasi *Naive Bayes* Multinomial (MultinomialNB) untuk melakukan prediksi dan evaluasi kinerja pada data pengujian (`X_test`, `y_test`). Model dilatih dengan data pelatihan (`X_train`, `y_train`) dan digunakan untuk memprediksi data pengujian, dengan hasil prediksi disimpan dalam variabel `predicted`.

Setelahnya, fungsi mencetak metrik evaluasi seperti akurasi (`accuracy_score`), presisi (`precision_score`), recall (`recall_score`), dan F1-score (`f1_score`). Akurasi mengukur ketepatan keseluruhan model, presisi mengukur ketepatan prediksi positif, recall mengukur kemampuan model dalam mendeteksi kelas positif, dan F1-score adalah gabungan dari presisi dan recall. Semua metrik ini dihitung dengan bobot seimbang untuk kelas positif ("Positif").

Fungsi juga mencetak matriks kebingungan (`confusion_matrix`), yang menunjukkan jumlah prediksi benar dan salah untuk setiap kelas, memberikan wawasan detail tentang kinerja model. Terakhir, fungsi mencetak laporan klasifikasi (`classification_report`), yang merangkum presisi, recall, dan F1-score untuk setiap kelas, serta menangani kasus nol untuk menghindari pembagian dengan nol, sehingga membantu evaluasi kinerja model secara menyeluruh.

Pada klasifikasi NB menggunakan dataset hasil anotasi bahasa Indonesia dilakukan dengan menggunakan perbandingan data latih dan data uji sebesar 80:20, 75:25 dan 70:30, sedangkan data hasil klasifikasi model *Naive Bayes* akan disajika dalam bentuk tabel yang diurutkan tidak berdasarkan percobaan, tapi berdasarkan nilai tertinggi ke nilai terendah.

1. Hasil Klasifikasi *Naive Bayes* menggunakan Dataset Bahasa Indonesia

a. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 1 bahasa indonesia

Hasil klasifikasi menggunakan anotasi ke – 1 dapat dilihat pada tabel 4.21 dibawah ini:

Tabel 4.21. Tabel hasil klasifikasi model Naïve Bayes menggunakan Anotasi ke – 1 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
3	NB	Bahasa indonesia	*TextBblob*	x	70%	30%	97,07%
1	NB	Bahasa indonesia	*TextBblob*	x	80%	20%	96,58%
2	NB	Bahasa indonesia	*TextBblob*	x	75%	25%	95,08%
5	NB	Bahasa indonesia	*VADER*	x	75%	25%	92,62%
6	NB	Bahasa indonesia	*VADER*	x	70%	30%	92,48%
4	NB	Bahasa indonesia	*VADER*	x	80%	20%	91,12%

Data hasil klasifikasi model *Naive Bayes* pada Tabel 4.21 disajikan dengan urutan yang tidak mengikuti urutan percobaan, melainkan diurutkan dari nilai tertinggi hingga terendah. Pada tabel tersebut, terlihat bahwa percobaan ke-3 dari hasil anotasi pertama menggunakan library *TextBlob* menunjukkan hasil akurasi model *Naive Bayes* yang terbaik. Akurasi yang dicapai dalam percobaan ini adalah 97,07%, yang merupakan hasil dari pembagian data latih dan data uji dengan rasio 70:30. Ini menandakan bahwa model *Naive Bayes* pada percobaan tersebut mampu mengklasifikasikan data dengan sangat baik dibandingkan dengan percobaan lainnya yang ada dalam tabel.

b. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 2 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 2 dapat dilihat pada tabel 4.22 dibawah ini:

Tabel 4.22. Tabel hasil klasifikasi model Naïve Bayes menggunakan Anotasi ke – 2 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	NB	Bahasa indonesia	*TextBblob*	x	75%	25%	98,36%
3	NB	Bahasa indonesia	*TextBblob*	x	70%	30%	97,72%
1	NB	Bahasa indonesia	*TextBblob*	x	80%	20%	97,28%
5	NB	Bahasa indonesia	*VADER*	x	75%	25%	97,26%
4	NB	Bahasa indonesia	*VADER*	x	80%	20%	97,09%
6	NB	Bahasa indonesia	*VADER*	x	70%	30%	95,80%

Pada Tabel 4.22 di atas, dapat dilihat bahwa hasil akurasi tertinggi dari model *Naive Bayes* tercatat pada percobaan kedua, yang diambil dari hasil anotasi kedua menggunakan library *TextBlob*. Model ini menunjukkan performa yang sangat baik dengan tingkat akurasi mencapai 98,36%. Akurasi ini diperoleh dari pembagian data latih dan data uji dengan rasio 75:25. Rasio ini berarti bahwa 75% dari data digunakan untuk melatih model, sementara 25% sisanya digunakan untuk menguji model, sehingga memberikan gambaran yang akurat tentang kemampuan model dalam mengklasifikasikan data dengan benar. Hasil ini menunjukkan

efektivitas penggunaan *TextBlob* dalam analisis sentimen dan ketepatan model *Naive Bayes* dalam mengolah data yang tersedia.

c. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 3 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 3 dapat dilihat pada tabel 4.23 dibawah ini:

Tabel 4.23. Tabel hasil klasifikasi model Naïve Bayes menggunakan Anotasi ke – 3 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	NB	Bahasa indonesia	*TextBblob*	x	80%	20%	98,03%
3	NB	Bahasa indonesia	*TextBblob*	x	70%	30%	97,49%
4	NB	Bahasa indonesia	*VADER*	x	80%	20%	96,92%
6	NB	Bahasa indonesia	*VADER*	x	70%	30%	96,12%
5	NB	Bahasa indonesia	*VADER*	x	75%	25%	95,08%
2	NB	Bahasa indonesia	*TextBblob*	x	75%	25%	92,26%

Pada Tabel 4.23 di atas, terlihat bahwa hasil akurasi terbaik untuk model *Naive Bayes* tercatat pada percobaan pertama dari hasil anotasi ketiga yang menggunakan library *TextBlob*. Model ini mencapai akurasi sebesar 98,03% berdasarkan persentase data latih dan data uji yang digunakan, yaitu 80:20. Angka ini menunjukkan kinerja model yang sangat baik dalam mengklasifikasikan sentimen dengan menggunakan data yang telah dianotasi.

d. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 4 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 4 dapat dilihat pada tabel 4.24 dibawah ini:

Tabel 4.24. Tabel hasil klasifikasi model Naïve Bayes menggunakan Anotasi ke – 4 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	NB	Bahasa indonesia	*TextBblob*	√	75%	25%	97,00%
3	NB	Bahasa indonesia	*TextBblob*	√	70%	30%	95,52%
1	NB	Bahasa indonesia	*TextBblob*	√	80%	20%	94,77%
6	NB	Bahasa indonesia	*VADER*	√	70%	30%	93,03%
4	NB	Bahasa indonesia	*VADER*	√	80%	20%	90,29%
5	NB	Bahasa indonesia	*VADER*	√	75%	25%	89,82%

Pada tabel 4.24 diatas terlihat bahwa hasil akurasi model *Naive Bayes* terbaik berada pada percobaan ke 2 dari hasil anotasi ke – 4 menggunakan *library TextBlob* dengan hasil akurasi sebesar 97,00% dari presentase data latih dan data uji yaitu 75:25. Selain itu anotasi data yang keempat menggunakan *drop duplicate* data, sehingga data yang di gunakan dalam klasifikasi berbeda jumlahnya dengan dataset pada anotasi 1 sampai 3.

e. Analisa perbandingan hasil klasifikasi model *Naive Bayes*.

Perbandingan hasil klasifikasi *Naive Bayes* dari anotasi 1 sampai 4 dapat dilihat pada tabel 4.25 dibawah ini:

Tabel 4.25. Tabel Analisa perbandingan hasil klasifikasi model *Naive Bayes* pada dataset bahasa indonesia

Anotasi	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	NB	Bahasa indonesia	*TextBblob*	x	75%	25%	98,36%
3	NB	Bahasa indonesia	*TextBblob*	x	80%	20%	98,03%
1	NB	Bahasa indonesia	*TextBblob*	x	70%	30%	97,07%
4	NB	Bahasa indonesia	*TextBblob*	√	75%	25%	97,00%

Tabel di atas menampilkan hasil evaluasi kinerja model *Naive Bayes* (NB) yang diterapkan pada dataset teks berbahasa Indonesia menggunakan lexicon *TextBlob* dan *VADER*, dengan berbagai skenario pengaturan data dan penghapusan duplikat. Dari hasil klasifikasi dengan berbagai skenario, nilai tertinggi selalu berada pada *library TextBlob*, sehingga pada tabel tidak ada hasil klasifikasi dari *VADER*. Setiap anotasi mencerminkan bagaimana variasi dalam pembagian data pelatihan dan pengujian serta penghapusan duplikat mempengaruhi akurasi model. Dalam skenario pertama (anotasi 2), model dilatih dengan menggunakan 75% data sebagai data pelatihan dan 25% sebagai data pengujian, menghasilkan akurasi tertinggi sebesar 98,36%. Ini menunjukkan bahwa proporsi ini memberikan

keseimbangan optimal antara jumlah data yang cukup untuk pelatihan dan validasi melalui pengujian.

Skenario kedua (anotasi 3) menunjukkan bahwa proporsi data pelatihan 80% dan data pengujian 20% menghasilkan akurasi model sedikit menurun menjadi 98,03%, dibandingkan dengan anotasi 1. Penurunan ini mengindikasikan bahwa proporsi data serta sentimen dalam data pengujian mempengaruhi kemampuan model. Pada skenario ketiga (anotasi 1), di mana 70% data digunakan untuk pelatihan dan 30% untuk pengujian, akurasi model turun lebih jauh menjadi 97,07%. Meskipun jumlah sentimen pada anotasi 2 dan 3 sama, perbedaan terletak pada proses case folding, yang mempengaruhi hasil klasifikasi.

Skenario keempat (anotasi 4) menggabungkan penghapusan duplikat data dengan proporsi pembagian data 75% untuk pelatihan dan 25% untuk pengujian. Hasilnya adalah akurasi sebesar 97,00%, yang lebih rendah dibandingkan dengan skenario tanpa penghapusan duplikat. Ini menunjukkan bahwa meskipun penghapusan duplikat dapat membantu dalam mengurangi redundansi dan mungkin mengurangi *overfitting*, dalam kasus ini, penghapusan duplikat mengurangi variasi dalam data pelatihan, yang pada gilirannya mengurangi akurasi model.

Dari hasil yang tertera pada tabel diatas dapat dikatakan bahwa *TextBlob* mempunyai pengaruh terhadap bahasa indoensia. *TextBlob*, yang pada dasarnya dirancang untuk analisis teks berbahasa Inggris, masih dapat digunakan untuk teks berbahasa Indonesia, meskipun hasilnya mungkin tidak seakurat jika digunakan pada teks berbahasa Inggris. Hasil akurasi yang tinggi dalam tabel menunjukkan bahwa *TextBlob* masih mampu memberikan hasil yang cukup baik dalam skenario

ini. Namun, dapat pada dasarnya *TextBlob* mungkin tidak mengoptimalkan semua nuansa dan kekhasan bahasa Indonesia. Oleh karena itu, meskipun *TextBlob* dapat digunakan, hasilnya harus selalu divalidasi dan dibandingkan dengan alat analisis teks yang dirancang khusus untuk bahasa Indonesia.

2. Hasil Klasifikasi *Naive Bayes* menggunakan Dataset Bahasa Inggris

Untuk klasifikasi *Naive Bayes* menggunakan dataset bahasa inggris, nilai C yang di gunakan adalah "0.01, 0.05, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5".

f. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 1 bahasa inggris

Hasil klasifikasi menggunakan anotasi ke – 1 dapat dilihat pada tabel 4.26 dibawah ini:

Tabel 4.26. Tabel hasil klasifikasi model *Naive Bayes* menggunakan Anotasi ke – 1 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	NB	Bahasa inggris	*TextBblob*	x	75%	25%	83,67%
3	NB	Bahasa inggris	*TextBblob*	x	70%	30%	83,55%
1	NB	Bahasa inggris	*TextBblob*	x	80%	20%	80,54%
4	NB	Bahasa inggris	*VADER*	x	80%	20%	78,42%
6	NB	Bahasa inggris	*VADER*	x	70%	30%	76,07%
5	NB	Bahasa inggris	*VADER*	x	75%	25%	73,77%

Data hasil klasifikasi model *Naive Bayes* pada tabel 4.26 diurutkan tidak berdasarkan percobaan, tapi berdasarkan nilai tertinggi ke nilai terendah. pada tabel 4.26 diatas terlihat bahwa hasil akurasi model *Naive Bayes* terbaik berada pada percobaan ke 2 dari hasil anotasi ke – 1 menggunakan *library TextBlob* dengan hasil akurasi sebesar 83,67% dari presentase data latih dan data uji yaitu 75:25.

g. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 2 bahasa inggris.

Hasil klasifikasi menggunakan anotasi ke – 2 dapat dilihat pada tabel 4.27 dibawah ini:

Tabel 4.27. Tabel hasil klasifikasi model *Naive Bayes* menggunakan Anotasi ke – 2 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	NB	Bahasa inggris	*TextBblob*	x	75%	25%	81,11%
1	NB	Bahasa inggris	*TextBblob*	x	80%	20%	79,03%
4	NB	Bahasa inggris	*VADER*	x	80%	20%	78,60%
3	NB	Bahasa inggris	*TextBblob*	x	70%	30%	77,55%
5	NB	Bahasa inggris	*VADER*	x	75%	25%	76,92%
6	NB	Bahasa inggris	*VADER*	x	70%	30%	73,76%

Pada tabel 4.27 diatas terlihat bahwa hasil akurasi model *Naive Bayes* terbaik berada pada percobaan ke 2 dari hasil anotasi ke – 2 menggunakan *library*

TextBlob dengan hasil akurasi sebesar 81,11% dari presentase data latih dan data uji yaitu 72:25.

h. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 3 bahasa inggris.

Hasil klasifikasi menggunakan anotasi ke – 3 dapat dilihat pada tabel 4.28 dibawah ini:

Tabel 4.28. Tabel hasil klasifikasi model Naïve Bayes menggunakan Anotasi ke – 3 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	NB	Bahasa inggris	*TextBblob*	x	75%	25%	82,51%
1	NB	Bahasa inggris	*TextBblob*	x	80%	20%	80,78%
3	NB	Bahasa inggris	*TextBblob*	x	70%	30%	78,71%
6	NB	Bahasa inggris	*VADER*	x	70%	30%	76,09%
4	NB	Bahasa inggris	*VADER*	x	80%	20%	72,92%
5	NB	Bahasa inggris	*VADER*	x	75%	25%	72,37%

Pada tabel 4.28 diatas terlihat bahwa hasil akurasi model *Naive Bayes* terbaik berada pada percobaan ke 2 dari hasil anotasi ke – 3 menggunakan *library TextBlob* dengan hasil akurasi sebesar 82,51% dari presentase data latih dan data uji yaitu 72:25.

i. Hasil klasifikasi *Naive Bayes* menggunakan dataset hasil anotasi ke – 4 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 4 dapat dilihat pada tabel 4.29 dibawah ini:

Tabel 4.29. Tabel hasil klasifikasi model Naïve Bayes menggunakan Anotasi ke – 4 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	NB	Bahasa inggris	*TextBblob*	√	80%	20%	70,58%
2	NB	Bahasa inggris	*TextBblob*	√	75%	25%	66,47%
3	NB	Bahasa inggris	*TextBblob*	√	70%	30%	65,68%
4	NB	Bahasa inggris	*VADER*	√	80%	20%	58,82%
5	NB	Bahasa inggris	*VADER*	√	75%	25%	58,82%
6	NB	Bahasa inggris	*VADER*	√	70%	30%	53,92%

Pada tabel 4.29 diatas terlihat bahwa hasil akurasi model *Naive Bayes* terbaik berada pada percobaan ke 2 dari hasil anotasi ke – 4 menggunakan *library TextBlob* dengan hasil akurasi sebesar 66,47% dari presentase data latih dan data uji yaitu 75:25. Selain itu anotasi data yang keempat menggunakan *drop duplicate* data, sehingga data yang di gunakan dalam klasifikasi berbeda jumlahnya dengan dataset pada anotasi 1 sampai 3. Hasil klasifikasi *Naive Bayes* menggunakan dataset dari anotasi ke – 4 bahasa inggris merupakan hasil akurasi *Naive Bayes* terkecil dibandingkan akurasi menggunakan anotasi lainnya yang berbahasa inggris.

j. Analisa perbandingan hasil klasifikasi model *Naive Bayes*.

Perbandingan hasil klasifikasi *Naive Bayes* dari anotasi 1 sampai 4 dapat dilihat pada tabel 4.30 dibawah ini:

Tabel 4.30. Tabel Analisa perbandingan hasil klasifikasi model *Naive Bayes* pada dataset bahasa inggris

Anotasi	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	NB	Bahasa inggris	*TextBblob*	x	75%	25%	83,67%
3	NB	Bahasa inggris	*TextBblob*	x	75%	25%	82,51%
2	NB	Bahasa inggris	*TextBblob*	x	75%	25%	81,11%
4	NB	Bahasa inggris	*TextBblob*	√	80%	20%	70,58%

Tabel 4.30 memperlihatkan hasil analisis perbandingan klasifikasi model *Naive Bayes* yang diterapkan pada dataset bahasa Inggris menggunakan lexicon *TextBlob* dan *VADER*. Empat skenario anotasi yang berbeda dievaluasi dengan mempertimbangkan penggunaan lexicon, penghapusan duplikat data, dan pembagian data untuk pelatihan dan pengujian. Dari percobaan-percobaan pada anotasi data menggunakan *TextBlob* dan *VADER* dan saat diimplementasikan pada klasifikasi model *Naive Bayes*, setiap percobaan nilai tertinggi selalu berada pada nilai akurasi *TextBlob*.

Pertama, pada skenario anotasi 1, model *Naive Bayes* dilatih dengan dataset yang tidak mengalami penghapusan duplikat dan menggunakan pembagian data 75% untuk pelatihan dan 25% untuk pengujian. Hasilnya, model mencapai akurasi

sebesar 83,67%, yang merupakan nilai tertinggi di antara keempat skenario. Ini menunjukkan bahwa penggunaan lexicon *TextBlob* pada dataset bahasa Inggris tanpa penghapusan duplikat dapat memberikan kinerja yang cukup baik.

Kedua, skenario anotasi 3 menunjukkan model *Naive Bayes* dengan konfigurasi serupa seperti anotasi 1, tetapi dengan hasil sedikit lebih rendah pada akurasi sebesar 82,51%. Penurunan ini mungkin disebabkan oleh variasi dalam data pelatihan dan pengujian, meskipun faktor lain seperti karakteristik data atau pembagian yang berbeda juga dapat memengaruhi hasil.

Ketiga, skenario anotasi 2 menunjukkan akurasi yang lebih rendah lagi, yakni 81,11%. Meskipun menggunakan konfigurasi yang sama dengan skenario 1 dan 3, variasi dalam data atau faktor lain yang tidak diketahui bisa menjadi penyebab penurunan ini.

Keempat, skenario anotasi 4 menunjukkan kinerja yang jauh lebih rendah dengan akurasi hanya 66,47%. Perbedaan utamanya adalah penggunaan penghapusan duplikat data sebelum pelatihan model. Hasil ini menunjukkan bahwa penghapusan duplikat dapat memengaruhi secara signifikan kinerja model, dengan potensi mengurangi akurasi secara drastis dalam kasus ini.

4.5.3.2 Validasi menggunakan Support Vector Machine (SVM)

Untuk klasifikasi menggunakan SVM menggunakan *script-script* dibawah ini:

```
x = df['Tweets']
y = df['LABTB']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

Gambar 4.64. Proses *spitting* data uji dan data latih (SVM)

Script diatas bertujuan untuk mengatur data dari dua kolom dalam *DataFrame* `df`: `*Tweets*` dan `LABTB`. Variabel `x` dan `y` menampung nilai dari kolom `*Tweets*` dan `LABTB` secara berturut-turut. Selanjutnya, fungsi `train_test_split` dari *library* scikit-learn digunakan untuk membagi data menjadi empat subset yang berbeda: `x_train`, `x_test`, `y_train`, dan `y_test`. Proses ini membagi data menjadi dua bagian: data latih (train) dan data uji (test), dengan ukuran data uji sebesar 20% dari total data. Data latih akan digunakan untuk melatih model, sedangkan data uji akan digunakan untuk menguji kinerja model yang telah dilatih.

Selanjutnya adalah prose mengubah data teks menjadi vektor angka menggunakan *script* dberikut:

```
vectorizer = CountVectorizer()
vectorizer.fit(x_train)
```

```
▾ CountVectorizer
CountVectorizer()
```

Gambar 4.65. Mengubah data teks menjadi vektor angka

Fungsi `CountVectorizer()` digunakan untuk mengubah kumpulan dokumen teks menjadi representasi vektor angka. Saat `CountVectorizer()` dipanggil, akan dibuat sebuah objek yang bisa menangani tokenisasi, menghitung frekuensi kata, dan membangun vektor fitur berdasarkan kumpulan teks yang

diberikan. Pada contoh ini, `vectorizer.fit(x_train)` digunakan untuk melatih atau menyesuaikan `CountVectorizer()` dengan data latih (`x_train`). Proses ini akan mengumpulkan seluruh kata yang muncul dalam data latih untuk kemudian dihitung frekuensinya dan dijadikan sebagai fitur dalam model vektor. Dengan menggunakan `CountVectorizer()`, setiap kata dalam dataset akan diubah menjadi fitur dengan nilai yang merepresentasikan berapa kali kata tersebut muncul dalam setiap dokumen teks, yang nantinya bisa digunakan sebagai input untuk model machine learning seperti klasifikasi teks atau analisis lainnya.

Klasifikasi SVM menggunakan *script* berikut ini:

```
for c in [0.01, 0.05, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5]:
    svm = LinearSVC(C=c)
    svm.fit(x_train, y_train)
    print('akurasi untuk c = %s: %s' %(c, accuracy_score(y_test, svm.predict(x_test))))
```

Gambar 4.66. Klasifikasi model *Support Vector Machine* (SVM)

Script di atas menggunakan pendekatan iteratif untuk melatih model SVM dengan variasi parameter C yang berbeda. Pada setiap iterasi, model LinearSVC dibuat dengan nilai C tertentu dari daftar yang telah ditentukan ([0.01, 0.05, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5]). Model ini kemudian dilatih menggunakan data pelatihan (x_train dan y_train). Setelah pelatihan selesai, akurasi model diukur menggunakan data uji (x_test dan y_test) dengan membandingkan prediksi model terhadap label yang sebenarnya menggunakan metrik `accuracy_score`.

Pada klasifikasi SVM menggunakan dataset hasil anotasi bahasa Indonesia dilakukan dengan menggunakan perbandingan data latih dan data uji sebesar 80:20, 75:25 dan 70:30, sedangkan data hasil klasifikasi model *Naive Bayes* akan disajikan

dalam bentuk tabel yang diurutkan tidak berdasarkan percobaan, tapi berdasarkan nilai tertinggi ke nilai terendah.

1. Hasil Klasifikasi SVM menggunakan Dataset Bahasa Indonesia

a. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 1 bahasa indonesia

Hasil klasifikasi menggunakan anotasi ke – 1 dapat dilihat pada tabel 4.31 dibawah ini:

Tabel 4. 31. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 1 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	SVM	Bahasa indonesia	*TextBblob*	x	80%	20%	97,61%
2	SVM	Bahasa indonesia	*TextBblob*	x	75%	25%	96,99%
3	SVM	Bahasa indonesia	*TextBblob*	x	70%	30%	96,12%
5	SVM	Bahasa indonesia	*VADER*	x	75%	25%	93,16%
4	SVM	Bahasa indonesia	*VADER*	x	80%	20%	92,83%
6	SVM	Bahasa indonesia	*VADER*	x	70%	30%	89,06%

Data hasil klasifikasi model *Naive Bayes* pada tabel 4. 31 diurutkan tidak berdasarkan percobaan, tapi berdasarkan nilai tertinggi ke nilai terendah. pada tabel 4.31 diatas terlihat bahwa hasil akurasi model SVM terbaik berada pada percobaan

ke 1 dari hasil anotasi ke – 1 menggunakan *library TextBlob* dengan hasil akurasi sebesar 97,61% dari presentase data latih dan data uji yaitu 80:20.

b. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 2 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 2 dapat dilihat pada tabel 4.32 dibawah ini:

Tabel 4.32. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 2 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
3	SVM	Bahasa indonesia	*TextBblob*	x	70%	30%	97,94%
1	SVM	Bahasa indonesia	*TextBblob*	x	80%	20%	97,61%
2	SVM	Bahasa indonesia	*TextBblob*	x	75%	25%	97,54%
5	SVM	Bahasa indonesia	*VADER*	x	75%	25%	95,90%
4	SVM	Bahasa indonesia	*VADER*	x	80%	20%	95,52%
6	SVM	Bahasa indonesia	*VADER*	x	70%	30%	95,67%

Pada Tabel 4.32 di atas, terlihat bahwa hasil akurasi terbaik untuk model Support Vector Machine (SVM) ditemukan pada percobaan ketiga dari hasil anotasi kedua, yang menggunakan library *TextBlob*. Dalam percobaan ini, model mencapai akurasi sebesar 97,94%, berdasarkan pembagian data latih dan data uji dengan rasio 70:30. Angka akurasi ini menunjukkan performa optimal dari model dalam

memproses dan mengklasifikasikan data, dengan hasil yang konsisten dan sangat baik..

c. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 3 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 3 dapat dilihat pada tabel 4.33 dibawah ini:

Tabel 4.33. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 3 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
3	SVM	Bahasa indonesia	*TextBblob*	x	70%	30%	97,94%
2	SVM	Bahasa indonesia	*TextBblob*	x	75%	25%	97,26%
1	SVM	Bahasa indonesia	*TextBblob*	x	80%	20%	96,24%
4	SVM	Bahasa indonesia	*VADER*	x	80%	20%	96,24%
5	SVM	Bahasa indonesia	*VADER*	x	75%	25%	96,99%
6	SVM	Bahasa indonesia	*VADER*	x	70%	30%	96,35%

Pada tabel 4.33 diatas terlihat bahwa hasil akurasi model SVM terbaik berada pada percobaan ke 3 dari hasil anotasi ke – 3 menggunakan *library TextBlob* dengan hasil akurasi sebesar 97,94% dari presentase data latih dan data uji yaitu 70:30.

d. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 4 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 4 dapat dilihat pada tabel 3.34 dibawah ini:

Tabel 3.34. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 4 dataset bahasa indonesia

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	SVM	Bahasa indonesia	*TextBblob*	√	80%	20%	96,26%
2	SVM	Bahasa indonesia	*TextBblob*	√	75%	25%	94,61 %
6	SVM	Bahasa indonesia	*VADER*	√	70%	30%	94,02%
3	SVM	Bahasa indonesia	*TextBblob*	√	70%	30%	93,03%
5	SVM	Bahasa indonesia	*VADER*	√	75%	25%	91,61%
4	SVM	Bahasa indonesia	*VADER*	√	80%	20%	90,29%

Pada tabel 4.34 diatas terlihat bahwa hasil akurasi model SVM terbaik berada pada percobaan ke 1 dari hasil anotasi ke – 4 menggunakan *library TextBlob* dengan hasil akurasi sebesar 96,26% dari presentase data latih dan data uji yaitu 80:20. Selain itu anotasi data yang keempat menggunakan *drop duplicate* data, sehingga data yang di gunakan dalam klasifikasi berbeda jumlahnya dengan dataset pada anotasi 1 sampai 3. Hal ini menunjukkan bahwa penghapusan data duplikat dapat berpengaruh pada hasil akurasi model.

e. Analisa perbandingan hasil klasifikasi model SVM.

Perbandingan hasil klasifikasi SVM dari anotasi 1 sampai 4 dapat dilihat pada tabel 3.35 dibawah ini:

Tabel 4.35. Tabel Analisa perbandingan hasil klasifikasi model SVM pada dataset bahasa indonesia

Anotasi	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	SVM	Bahasa indonesia	*TextBblob*	x	70%	30%	97,94%
3	SVM	Bahasa indonesia	*TextBblob*	x	70%	30%	97,94%
1	SVM	Bahasa indonesia	*TextBblob*	x	80%	20%	97,61%
4	SVM	Bahasa indonesia	*TextBblob*	√	80%	20%	96,26%

Tabel 4.35 menyajikan hasil analisis perbandingan klasifikasi model Support Vector Machine (SVM) pada dataset bahasa Indonesia, dengan menggunakan lexicon *TextBlob* dan *VADER*. Dari percobaan-percobaan pada anotasi data menggunakan *TextBlob* dan *VADER* dan saat diimplementasikan pada klasifikasi model SVM, setiap percobaan nilai tertinggi selalu berada pada nilai akurasi *TextBlob*. Terdapat empat skenario berbeda yang dievaluasi berdasarkan pengaruh dari penghapusan data duplikat serta pembagian data untuk pelatihan dan pengujian.

Pertama, skenario anotasi ke-2 dan ke-3 menggunakan pembagian data 70% untuk pelatihan dan 30% untuk pengujian, yang menghasilkan tingkat akurasi yang identik sebesar 97,94%. Kedua skenario ini menunjukkan konsistensi dalam

performa model SVM dengan pengaturan data yang serupa. Kedua, skenario anotasi ke-1 menggunakan pembagian data 80% untuk pelatihan dan 20% untuk pengujian, dengan akurasi sebesar 97,61%. Meskipun menggunakan proporsi data yang berbeda, model tetap menunjukkan hasil yang stabil dan tinggi dalam klasifikasi.

Skenario ketiga, yaitu anotasi keempat yang melibatkan penghapusan data duplikat, menghasilkan akurasi yang sedikit lebih rendah yaitu 96,26%. Penghapusan data duplikat mempengaruhi performa model dengan menurunkan tingkat akurasi, hal ini menunjukkan bahwa keberadaan data yang beragam dan unik sangat penting dalam proses pelatihan model. Data yang beragam memungkinkan model untuk belajar dari berbagai contoh dan pola, yang pada akhirnya dapat meningkatkan akurasi dan generalisasi model dalam menganalisis data baru. Oleh karena itu, meskipun penghapusan data duplikat dapat membantu mengurangi redundansi, penting untuk mempertimbangkan dampaknya terhadap kualitas dan keberagaman data pelatihan.

Penting untuk dicatat bahwa tabel ini hanya memuat data dari lexicon *TextBlob* karena hasil dari *VADER* tidak memenuhi kriteria tertinggi. Penggunaan lexicon *TextBlob* dalam konteks ini menunjukkan efektivitasnya dalam menerjemahkan dan mengklasifikasikan teks berbahasa Indonesia, meskipun alternatif seperti *VADER* atau lexicon bahasa Indonesia lainnya dapat menjadi pilihan tambahan untuk validasi lebih lanjut atau peningkatan performa analisis sentimen.

2. Hasil Klasifikasi SVM menggunakan Dataset Bahasa Inggris

Untuk klasifikasi *Naive Bayes* menggunakan dataset bahasa inggris, nilai C yang di gunakan adalah "0.01, 0.05, 0.25, 0.5, 0.75, 1, 1.5, 2, 2.5, 3, 3.5".

f. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 1 bahasa inggris

Hasil klasifikasi menggunakan anotasi ke – 1 dapat dilihat pada tabel 3.36 dibawah ini:

Tabel 4.36. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 1 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	SVM	Bahasa inggris	*TextBblob*	x	80%	20%	88,73%
2	SVM	Bahasa inggris	*TextBblob*	x	75%	25%	86,33%
3	SVM	Bahasa inggris	*TextBblob*	x	70%	30%	84,05%
4	SVM	Bahasa inggris	*VADER*	x	80%	20%	81,91%
5	SVM	Bahasa inggris	*VADER*	x	75%	25%	83,33%
6	SVM	Bahasa inggris	*VADER*	x	70%	30%	81,32 %

Data hasil klasifikasi model SVM pada tabel 4.36 diurutkan tidak berdasarkan percobaan, tapi berdasarkan nilai tertinggi ke nilai terendah. pada tabel 4.36 diatas terlihat bahwa hasil akurasi model *Naive Bayes* terbaik berada pada percobaan ke 1 dari hasil anotasi ke – 1 menggunakan *library TextBlob* dengan hasil akurasi sebesar 88,73% dari presentase data latih dan data uji yaitu 80:20.

g. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 2 bahasa inggris.

Hasil klasifikasi menggunakan anotasi ke – 2 dapat dilihat pada tabel 3.37 dibawah ini:

Tabel 4.37. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 2 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	SVM	Bahasa inggris	*TextBblob*	x	75%	25%	86,71%
3	SVM	Bahasa inggris	*TextBblob*	x	70%	30%	82,50%
1	SVM	Bahasa inggris	*TextBblob*	x	80%	20%	82,09%
6	SVM	Bahasa inggris	*VADER*	x	70%	30%	79,30%
4	SVM	Bahasa inggris	*VADER*	x	80%	20%	79,03%
5	SVM	Bahasa inggris	*VADER*	x	75%	25%	76,92%

Pada Tabel 4.37 di atas, dapat dilihat bahwa hasil akurasi terbaik untuk model *Naive Bayes* tercatat pada percobaan kedua, yang menggunakan hasil anotasi kedua dengan library *TextBlob*. Pada percobaan ini, model menunjukkan akurasi sebesar 86,71%. Ini menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengklasifikasikan data. Prosentase data yang digunakan dalam pelatihan dan pengujian adalah 72% untuk data latih dan 25% untuk data uji. Rasio ini menunjukkan bahwa sebagian besar data digunakan untuk melatih model, sementara sebagian kecil digunakan untuk menguji kinerja model, yang berkontribusi pada akurasi yang tinggi tersebut..

h. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 3 bahasa inggris.

Hasil klasifikasi menggunakan anotasi ke – 3 dapat dilihat pada tabel 3.38 dibawah ini:

Tabel 4.38. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 3 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
2	SVM	Bahasa inggris	*TextBblob*	x	75%	25%	83,91%
1	SVM	Bahasa inggris	*TextBblob*	x	80%	20%	83,40%
3	SVM	Bahasa inggris	*TextBblob*	x	70%	30%	77,25%
6	SVM	Bahasa inggris	*VADER*	x	70%	30%	85,13%
4	SVM	Bahasa inggris	*VADER*	x	80%	20%	84,71%
5	SVM	Bahasa inggris	*VADER*	x	75%	25%	84,26%

Pada Tabel 4.38 di atas, terlihat bahwa hasil akurasi terbaik untuk model *Naive Bayes* tercatat pada percobaan kedua, yang menggunakan hasil anotasi ketiga dan diterapkan dengan library *TextBlob*. Model ini mencapai akurasi sebesar 83,91%. Akurasi ini diperoleh dari pembagian data latih dan data uji yang terdiri dari 72% data latih dan 25% data uji, menunjukkan performa yang signifikan dalam mengklasifikasikan sentimen dengan menggunakan metode tersebut.

i. Hasil klasifikasi SVM menggunakan dataset hasil anotasi ke – 4 bahasa Indonesia.

Hasil klasifikasi menggunakan anotasi ke – 4 dapat dilihat pada tabel 4.39 dibawah ini:

Tabel 4.39. Tabel hasil klasifikasi model SVM menggunakan Anotasi ke – 4 dataset bahasa inggris

Try	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	SVM	Bahasa inggris	*TextBblob*	√	80%	20%	76,47%
3	SVM	Bahasa inggris	*TextBblob*	√	70%	30%	74,01%
5	SVM	Bahasa inggris	*VADER*	√	75%	25%	71,76%
2	SVM	Bahasa inggris	*TextBblob*	√	75%	25%	71,17%
6	SVM	Bahasa inggris	*VADER*	√	70%	30%	70,58%
4	SVM	Bahasa inggris	*VADER*	√	80%	20%	69,85%

Pada tabel 4.39 diatas terlihat bahwa hasil akurasi model *Naive Bayes* terbaik berada pada percobaan ke 2 dari hasil anotasi ke – 4 menggunakan *library TextBlob* dengan hasil akurasi sebesar 66,47% dari presentase data latih dan data uji yaitu 75:25. Selain itu anotasi data yang keempat menggunakan *drop duplicate* data, sehingga data yang di gunakan dalam klasifikasi berbeda jumlahnya dengan dataset pada anotasi 1 sampai 3. Hasil klasifikasi *Naive Bayes* dari anotasi ke – 4 bahasa inggris merupakan hasil akurasi *Naive Bayes* terkecil.

j. Analisa perbandingan hasil klasifikasi model SVM.

Perbandingan hasil klasifikasi SVM dari anotasi 1 sampai 4 dapat dilihat pada tabel 4.40 dibawah ini:

Tabel 4.40. Tabel Analisa perbandingan hasil klasifikasi model SVM pada dataset bahasa inggris

Anotasi	Model	Dataset	*Lexicon*	*Drop duplicate*	*Splitting Data*		*Model Accuracy*
					Train	*Test*	
1	SVM	Bahasa inggris	*TextBblob*	x	80%	20%	88,73%
3	SVM	Bahasa inggris	*TextBblob*	x	75%	25%	83,91%
2	SVM	Bahasa inggris	*TextBblob*	x	75%	25%	86,71%
4	SVM	Bahasa inggris	*TextBblob*	√	80%	20%	76,47%

Tabel 4.40 menampilkan hasil analisis perbandingan klasifikasi model SVM (Support Vector Machine) yang dilakukan pada dataset berbahasa Inggris. Penggunaan *lexicon TextBlob* dalam semua skenario menunjukkan variasi dalam akurasi model, yang dipengaruhi oleh pengaturan seperti pembagian data untuk pelatihan dan pengujian serta keberadaan data duplikat.

Pada skenario pertama (anotasi 1), model dilatih dengan menggunakan 80% data untuk pelatihan dan 20% untuk pengujian, menghasilkan akurasi tertinggi sebesar 88,73%. Skenario kedua (anotasi 3) menggunakan pembagian data 75% untuk pelatihan dan 25% untuk pengujian, dengan akurasi sedikit menurun menjadi 83,91%. Skenario ketiga (anotasi 2) juga menggunakan pembagian data 75% untuk pelatihan dan 25% untuk pengujian, dengan akurasi sebesar 86,71%. Pada skenario

terakhir (anotasi 4) yang melibatkan penghapusan data duplikat, dengan pembagian data 80% untuk pelatihan dan 20% untuk pengujian, akurasi menurun drastis menjadi 76,47%.

Penggunaan *lexicon TextBlob* dalam tabel ini menyoroti pentingnya proporsi yang tepat dalam pembagian data untuk pelatihan dan pengujian, serta pengaruh penghapusan data duplikat terhadap akurasi model. Meskipun nilai akurasi pada semua skenario terlihat beragam, keseluruhan menunjukkan bahwa skenario dengan pembagian data 80% untuk pelatihan dan 20% untuk pengujian memberikan performa yang lebih baik dibandingkan dengan skenario lainnya. Penilaian berdasarkan *lexicon VADER* tidak disertakan dalam tabel ini karena nilai akhir dari *VADER* tidak mencapai kategori tertinggi yang diharapkan, sehingga hanya data dari *lexicon TextBlob* yang disajikan.

4.5.4. Perbandingan hasil validasi menggunakan *Naive Bayes* dan *Support Vector Machine*

Hasil akurasi dari kedua model yang telah dibangun akan dibandingkan untuk mengetahui hasil dari model mana yang merupakan hasil terbaik. Namun untuk model dengan menggunakan dataset bahasa Indonesia tidak dapat dianggap sebagai salah satu nilai terbaik, karena dalam kamus *lexicon VADER* dan *TextBlob* untuk bahasa Indonesia masih minim, sehingga proses anotasi pada bahasa Indonesia menggunakan *lexicon VADER* dan *TextBlob* masih belum bisa dianggap sebagai salah satu proses anotasi terbaik.

Disini tabel perbandingan akurasi dari model dengan menggunakan dataset bahasa Indonesia dan bahasa inggris akan dipisahkan terlebih dahulu, lalu dari kedua tabel tersebut akan ditarik akurasi terbaik lalu dibandingkan kembali agar memperoleh hasil akhir yang terbaik.

Dibawah ini merupakan tabel perbandingan dari akurasi model menggunakan dataset bahasa Indonesia.

Tabel 4.41. Tabel Perbandingan Akurasi Model Support Vector Machine (SVM) dan *Naive Bayes* (NB) pada bahasa indonesia

Anotasi	**Model**	**Dataset**	***Lexicon***	***Drop Duplicate***	***Splitting Data***		***Model Accuracy***
					Train	***Test***	
2	NB	Bahasa indonesia	*TextBblob*	x	75%	25%	98,36%
2	SVM	Bahasa indonesia	*TextBblob*	x	70%	30%	97,94%
3	SVM	Bahasa indonesia	*TextBblob*	x	70%	30%	97,94%

Tabel 4.41 menyajikan perbandingan akurasi model Support Vector Machine (SVM) dan *Naive Bayes* (NB) yang dilatih dengan dataset teks berbahasa Indonesia menggunakan lexicon dari *TextBlob*. Terdapat dua jenis anotasi yang dibandingkan dalam tabel ini, yaitu NB dan SVM. Model SVM diuji dalam dua skenario pembagian data yang berbeda: 70% data untuk pelatihan dan 30% untuk pengujian. Hasilnya menunjukkan akurasi yang sama, yaitu 97,94% untuk kedua skenario ini.

Sementara itu, model NB diuji dalam satu skenario dengan pembagian data 75% untuk pelatihan dan 25% untuk pengujian, dan menghasilkan akurasi tertinggi

sebesar 98,36%. Meskipun model SVM juga menunjukkan kinerja yang sangat baik dengan akurasi yang mendekati nilai NB, perbedaan dalam skenario pembagian data menunjukkan bahwa pengaturan pembagian data dapat mempengaruhi performa model secara signifikan.

Penggunaan *lexicon TextBlob* dalam eksperimen ini menunjukkan bahwa lexicon tersebut cukup efektif dalam mendukung akurasi model NB dalam menganalisis sentimen teks berbahasa Indonesia dibandingkan dengan *lexicon VADER*.

Tabel 4.42. Tabel Perbandingan Akurasi Model Support Vector Machine (SVM) dan *Naive Bayes* (NB) bahasa inggris

Anotasi	**Model**	**Dataset**	***Lexicon***	***Drop Duplicate***	***Splitting Data***		***Model Accuracy***
					Train	***Test***	
1	SVM	Bahasa inggris	*TextBblob*	x	80%	20%	88,73%
1	NB	Bahasa inggris	*TextBblob*	x	75%	25%	83,67%

Tabel 4.42 menunjukkan perbandingan akurasi antara model Support Vector Machine (SVM) dan *Naive Bayes* (NB) yang dilatih menggunakan dataset bahasa Inggris dengan lexicon *TextBlob*. Hasil menunjukkan bahwa model SVM memiliki akurasi lebih tinggi dibandingkan dengan model NB. Model NB, yang dilatih dengan pembagian data 75% untuk pelatihan dan 25% untuk pengujian, mencapai akurasi sebesar 83,67%. Sebaliknya, model SVM, yang dilatih dengan pembagian data 80% untuk pelatihan dan 20% untuk pengujian, menghasilkan akurasi yang lebih tinggi yaitu 88,73%.

Kesimpulan dari tabel ini adalah bahwa model SVM menunjukkan performa yang lebih baik dibandingkan dengan model NB dalam menganalisis dataset bahasa Inggris menggunakan lexicon *TextBlob*. Perbedaan dalam skenario pembagian data juga mengindikasikan bahwa model SVM lebih mampu memanfaatkan proporsi data pelatihan yang lebih besar untuk meningkatkan akurasinya. Oleh karena itu, dalam konteks penggunaan *TextBlob* untuk analisis sentimen pada teks berbahasa Inggris, model SVM dapat dianggap lebih efektif dibandingkan dengan model NB.

BAB V

PENUTUP

5.1 Kesimpulan

Dari penelitian yang telah dilakukan, peneliti dapat menarik beberapa kesimpulan sebagai berikut:

a. Dari analisis anotasi data yang dilakukan sebelum dan sesudah proses *cleaning*, *Casefolding*, dan *drop duplicate data* ditemukan bahwa proses-proses tersebut mempunyai pengaruh terhadap perubahan nilai *compound, subjektivitas* dan *polaritas*. Perubahan nilai *compound, subjektivitas* dan *polaritas* sangat mempengaruhi hasil anotasi data, dimana hasil anotasi data pada setiap skenario berubah-ubah dan hasil anotasi di stiap skenario juga mempengaruhi hasil validasi sentimen menggunakan SVM dan *Naive Bayes*.

b. Dari analisis yang dilakukan terhadap hasil klasifikasi menggunakan model *Naive Bayes* dan Support Vector Machine (SVM), ditemukan bahwa model SVM memberikan performa terbaik dibandingkan *Naive Bayes*. Percobaan dilakukan dengan menggunakan dua jenis dataset, yaitu bahasa Indonesia dan bahasa Inggris yang merupakan hasil terjemahan dari dataset bahasa Indonesia. Selain itu, digunakan dua jenis lexicon untuk anotasi pelabelan, yaitu *VADER* dan *TextBlob*. Untuk dataset bahasa Indonesia, baik *Naive Bayes* maupun SVM menunjukkan akurasi tertinggi sebesar 98,36%, terutama saat menggunakan lexicon *TextBlob* dan tanpa penghapusan data duplikat. Namun, akurasi ini tidak dapat dianggap optimal karena keterbatasan dari *lexicon VADER* maupun

TextBlob dalam mendukung bahasa Indonesia. Untuk dataset bahasa Inggris, SVM dengan lexicon *TextBlob* dan tanpa penghapusan data duplikat mencapai akurasi tertinggi sebesar 88,73%, dengan nilai C terbaik sebesar 0,25 dan pembagian data latih dan uji 80:20.

c. Secara keseluruhan, penelitian ini menghasilkan nilai akurasi diatas nilai kelayakan yaitu 80% dan hasil tersebut lebih tinggi dibandingkan dengan hasil rata-rata dari *state of the art researce*.

5.2 Saran

Setelah melakukan penelitian ini, peneliti merumuskan beberapa saran sebagai peningkatan penelitian selanjutnya, antara lain:

a. Untuk teks berbahasa Indonesia, perlu mengembangkan atau memperbarui lexicon yang lebih komprehensif dan relevan untuk meningkatkan akurasi anotasi. Menggabungkan lexicon dari berbagai sumber atau membuat lexicon khusus untuk domain tertentu bisa membantu.

b. Untuk penelitian selanjutnya dapat mengkombinasikan metode anotasi otomatis dengan anotasi manual untuk meningkatkan akurasi. Anotasi manual oleh penutur asli dapat membantu mengidentifikasi kesalahan atau bias dalam hasil otomatis dan menyesuaikan nilai compound dan polarity dengan lebih tepat.

c. Selain SVM dan *Naive Bayes*, penelitian lebih lanjut dapat mengeksplorasi model lain seperti Random Forest atau deep learning models untuk melihat apakah ada peningkatan signifikan dalam akurasi.

DAFTAR PUSTAKA

Abimanyu, D., Budianita, E., Pandu Cynthia, E., Yanto, F., Studi Teknik Informatika, P., & Sains Dan Teknologi, F. (2022). Analisis Sentimen Akun Twitter Apex Legends Menggunakan *VADER*. *Jurnal Nasional Komputasi Dan Teknologi Informasi*, *5*(3). https://techno.kompas.com

Abiola, O., Abayomi-Alli, A., Tale, O. A., Misra, S., & Abayomi-Alli, O. (2023). Sentiment analysis of COVID-19 *tweets* from selected hashtags in Nigeria using *VADER* and Text Blob analyser. *Journal of Electrical Systems and Information Technology*, *10*(1). https://doi.org/10.1186/s43067-023-00070-9

Afrillia, Y., Rosnita, L., & Siska, D. (2022). Analisis Sentimen Ciutan Twitter Terkait Penerapan Permendikbudristek Nomor 30 Tahun 2021 Menggunakan *TextBlob* dan Support Vector Machine. *G-Tech: Jurnal Teknologi Terapan*, *6*(2). https://doi.org/10.33379/gtech.v6i2.1778

Alenzi, B. M., Khan, M. B., Hasanat, M. H. A., Saudagar, A. K. J., Alkhathami, M., & Altameem, A. (2022). Automatic Annotation Performance of *TextBlob* and *VADER* on Covid Vaccination Dataset. *Intelligent Automation and Soft Computing*, *34*(2). https://doi.org/10.32604/iasc.2022.025861

Al-Shabi, M. A. (2020). Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining. *International Journal of Computer Science and Network Security*, *20*(1).

Ardiansah, I., & Maharani, A. (2020). Optimalisasi Instagram Sebagai Media Marketing. In *CV. Cendekian Press*.

Arispe, M. C. A., Capucao, J. N. B., Relucio, F. S., & Maligat, D. E. , Jr. (2019). Teachers' sentiments to Bikol MTB-MLE: Using sentiment analysis and text mining techniques. *International Journal of Research Studies in Education*, *8*(4). https://doi.org/10.5861/ijrse.2019.4906

Ashari, H., Afrianto, D., & Al Faruq, H. A. (2020). Perbandingan Kinerja Algoritma Multinomial Naive Bayes (MNB), Multivariate Bernoulli dan Rocchio Algorithm Dalam Klasifikasi Konten Berita Hoax Berbahasa Indonesia Pada Media Sosial. *Doctoral Dissertation, Universitas Muhammadiyah Jember*.

Aslam, N., Rustam, F., Lee, E., Washington, P. B., & Ashraf, I. (2022). Sentiment Analysis and Emotion Detection on Cryptocurrency Related *Tweets* Using Ensemble LSTM-GRU Model. *IEEE Access*, *10*. https://doi.org/10.1109/ACCESS.2022.3165621

Aulia, G. N., & Patriya, E. (2019). IMPLEMENTASI LEXICON BASED DAN NAIVE BAYES PADA ANALISIS SENTIMEN PENGGUNA TWITTER TOPIK PEMILIHAN PRESIDEN 2019. *Jurnal Ilmiah Informatika Komputer*, *24*(2). https://doi.org/10.35760/ik.2019.v24i2.2369

Ayu Muthia, D. (2017). ANALISIS SENTIMEN PADA REVIEW RESTORAN DENGAN TEKS BAHASA INDONESIA MENGUNAKAN ALGORITMA NAIVE BAYES. *JURNALILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER*, *2*(2), 39–45. www.zomato.com

Azhar, R., Surahman, A., & Juliane, C. (2022). Analisis Sentimen Terhadap Cryptocurrency Berbasis Python *TextBlob* Menggunakan Algoritma Naïve Bayes. In *Jurnal Sains Komputer & Informatika (J-SAKTI* (Vol. 6, Issue 1).

Baita, A., & Cahyono, N. (2021). Analisis Sentimen Mengenai Vaksin Sinovac Menggunakan Algoritma Support Vector Machine (SVM) dan K-Nearest NEighbor (KNN). *Information System Journal (INFOS)*, *4*(2), 42–46.

Barai, M. K. (2021, October 20). *Sentiment Analysis with TextBlob and VADER in Python*. https://www.analyticsvidhya.com/blog/2021/10/sentiment-analysis-with-*TextBlob*-and-*VADER*/

Dewi, S., & Arianto, D. B. (2023). Twitter Sentiment Analysis Towards Qatar as Host of the 2022 World Cup Using *TextBlob*. *Journal of Social Research*, *2*(2). https://doi.org/10.55324/josr.v2i2.615

Hendy Evan, F., & Sigit Purnomo, Y. W. (2014). Pembangunan Perangkat Lunak Peringkas Dokumen dari Banyak Sumber Menggunakan Sentence Scoring dengan Metode TF-IDF. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Yogyakarta*, G17–G22.

Pratiwi, N. I., & Widodo. (2017). Klasifikasi Dokumen Karya Akhir Mahasiswa Menggunakan NaÃ¯ve Bayes Classifier (NBC) Berdasarkan Abstrak Karya Akhir Di Jurusan Teknik Elektro Universitas Negeri Jakarta. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, *1*(1), 31–38. https://doi.org/10.21009/pinter.1.1.5

Irfan, M., Dewi, P. S., Zulfikar, W. B., Slamet, C., & Taufik, I. (2022). Sentiment Analysis as Assessment of the COVID-19 Social Assistance Pollemic using Random Forest Algorithm. *Proceeding of 2022 8th International Conference on Wireless and Telematics, ICWT 2022*. https://doi.org/10.1109/ICWT55831.2022.9935483

Law Insider. (2023). *Results Data Definition | Law Insider*. https://www.lawinsider.com/dictionary/results-data

Lestari, N. A., Akhriza, M., Yuniar, E., Ppkia, S., Paramita, P., Laksda, J., Sucipto, A., & Timur, J. (2020). Metode Naive Bayes Classifier dengan *TextBlob* untuk analisis Sentimen Terhadap Pelayanan Indihome dan First Media. *Seminar Nasional Teknologi Informasi Dan Komunikasi STI&K (SeNTIK)*, *4*(1). https://t.co/Ws2wOyU5kz

Mas Diyasa, I. G. S., Marini Mandenni, N. M. I., Fachrurrozi, M. I., Pradika, S. I., Nur Manab, K. R., & Sasmita, N. R. (2021). Twitter Sentiment Analysis as an Evaluation and Service Base On Python *TextBlob*. *IOP Conference Series: Materials Science and Engineering*, *1125*(1). https://doi.org/10.1088/1757-899x/1125/1/012034

Nugroho, A. S., Witarto, A. B., & Handoko, D. (2003). *Support Vector Machine-Teori dan Aplikasinya dalam Bioinformatika 1*. http://asnugroho.net

Nur Syahirah Wan Min, W., Zareen Zulkarnain, N., & Teknologi Maklumat Dan Komunikasi, F. (2020). Comparative Evaluation of Lexicons in Performing Sentiment Analysis. In *JOURNAL OF ADVANCED COMPUTING TECHNOLOGY AND APPLICATION (JACTA)* (Vol. 2, Issue 1).

Prihatini, P. M. (2016). Implementasi Ekstraksi Fitur Pada Pengolahan Dokumen Berbahasa Indonesia. *Jurnal Matrix*, *6*(3).

Ren, R., Wu, D. D., & Wu, D. D. (2019). Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Systems Journal*, *13*(1). https://doi.org/10.1109/JSYST.2018.2794462

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, *60*(5). https://doi.org/10.1108/00220410410560582

Rolly Intan, & Andrew Defeng. (2006). HARD: SUBJECT-BASED SEARCH ENGINE MENGGUNAKAN TF-IDF DAN JACCARD'S COEFFICIENT. *Jurnal Teknik Industri*, *8*(1).

Su, S. (2022). Sentimental Analysis Applied on Movie Reviews. *Journal of Education, Humanities and Social Sciences*, *3*. https://doi.org/10.54097/ehss.v3i.1685

Suanpang, P., Jamjuntr, P., & Kaewyong, P. (2021). Sentiment Analysis With a *TextBlob* Package Implications for Tourism. *Journal of Management Information and Decision Sciences*, *24*(6).