

**ANALYSIS OPINI PUBLIK TERHADAP KENDARAAN LISTRIK
MENGUNAKAN ALGORITMA *NAÏVE BAYES CLASSIFIER*
(NBC) DENGAN EKSTRAKSI FITUR *TF-IDF***

Tugas Akhir
Untuk memenuhi sebagian persyaratan
mencapai derajat Sarjana S-1 Jurusan Teknik Elektro



Oleh:

**L. M. Ryas Amin Akbar
F1B 019 076**

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS MATARAM**

2023

Tugas akhir

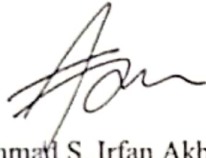
***ANALYSIS OPINI PUBLIK TERHADAP KENDARAAN LISTRIK
MENGUNAKAN ALGORITMA NAÏVE BAYES CLASSIFIER (NBC) DENGAN
EKSTRAKSI FITUR TF-IDF***

Oleh:

**L. M. Ryas Amin Akbar
F1B 019 076**

Telah diperiksa dan disetujui oleh Tim Pembimbing:

1. Pembimbing Utama



**L. Ahmad S. Irfan Akbar, ST., M.Eng
NIP: 198303102009121004**

Tanggal: 2/11/2023

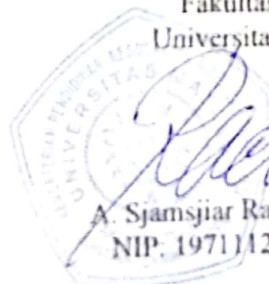
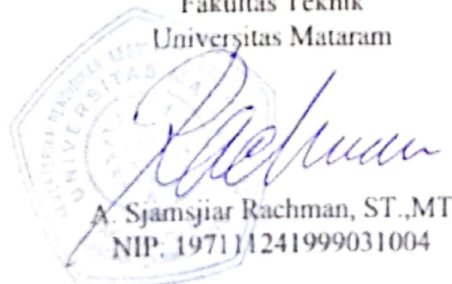
2. Pembimbing Pendamping



**Suthami Ariessaputra, ST., M.Eng
NIP: 198503272014041001**

Tanggal: 3/11/2023

Mengetahui
Ketua Jurusan Teknik Elektro
Fakultas Teknik
Universitas Mataram



**A. Sjamsjiar Rachman, ST., MT
NIP. 197111241999031004**

Tugas Akhir

**ANALYSIS OPINI PUBLIK TERHADAP KENDARAAN LISTRIK
MENGUNAKAN ALGORITMA NAÏVE BAYES CLASSIFIER (NBC)
DENGAN EKSTRAKSI FITUR TF-IDF**

Oleh:

**L. M. Ryas Amin Akbar
F1B 019 076**

Telah dipertahankan di depan Dewan Penguji
Pada Tanggal *2 November 2023*
dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1
Jurusan Teknik Elektro

Susunan Tim Penguji:

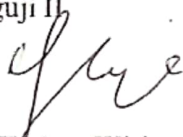
1. Penguji I



Cipta Ramadhani, ST., M.Eng.
NIP: 198506162019031008

Tanggal: *1 - 11 - 2023*


2. Penguji II



Giri Wahyu Wiriasto, ST., M.T.
NIP: 198209042010121001

Tanggal: *1 - 11 - 2023*

3. Penguji III



Dr. Ida Ayu Sri Adnyani, ST., M.Erg.
NIP: 197008231998022001

Tanggal: *31 - 10 - 2023*

Mataram,
Dekan Fakultas Teknik
Universitas Mataram



Muhammad Syamsu Iqbal, ST., MT., Ph.D.
NIP: 197202221999031002

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : L.M. Ryas Amin Akbar
NIM : F1B019076
Program Studi : Teknik Elektro
Perguruan Tinggi : Universitas Mataram

Dengan ini menyatakan bahwa tugas akhir yang telah saya buat dengan judul: “*Analysis Opini Publik Terhadap Kendaraan Listrik Menggunakan Algoritma Naïve Bayes Classifier (NBC) Dengan Ekstraksi Fitur Tf-Idf*” adalah asli (orsinil) atau tidak plagiat (menjiplak) dan belum pernah diterbitkan/dipublikasikan dimanapun dan dalam bentuk apapun.

Demikianlah surat pernyataan ini saya buat dengan sebenar-benarnya tanpa ada paksaan dari pihak manapun juga. Apabila dikemudian hari ternyata saya memberikan keterangan palsu dan atau ada pihak lain yang mengklaim bahwa tugas akhir yang telah saya buat adalah hasil karya milik seseorang atau badan tertentu, saya bersedia diproses baik secara pidana maupun perdata dan kelulusan saya dari Universitas Mataram dicabut/dibatalkan.

Dibuat di : Mataram
Pada tanggal : 6 November 2023
Yang menyatakan



L.M. Ryas Amin Akbar

PRAKATA

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Pengasih dan Penyayang atas segala berkat, bimbingan, dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan Tugas Akhir dengan judul ”*Analysis Opini Publik Terhadap Kendaraan Listrik Menggunakan Algoritma Naïve Bayes Classifier (NBC) Dengan Ekstraksi Fitur TF-IDF*”.

Tugas Akhir ini dilaksanakan di Laboratorium Komputer, Jurusan Teknik Elektro Universitas Mataram. Tujuan dari Tugas Akhir ini adalah pertama, untuk menilai persepsi masyarakat melalui analisis opini tentang kendaraan listrik berdasarkan teks komentar dengan *Naïve Bayes Classifier*; kedua, mengevaluasi performa model klasifikasi analisis yang dibangun menggunakan algoritma *Naïve Bayes Classifier* dengan ekstraksi fitur *Tf-Idf*; ketiga, membangun sistem aplikasi analisis opini berbasis web untuk klasifikasi, visualisasi, dan evaluasi data teks komentar. Tugas akhir ini juga merupakan salah satu persyaratan kelulusan guna mencapai gelar kesarjanaan di Jurusan Teknik Elektro, Fakultas Teknik Universitas Mataram.

Akhir kata, penulis berharap hasil dari penelitian ini dapat memberikan manfaat bagi seluruh masyarakat. Semoga penelitian ini dapat menjadi pijakan untuk penelitian lebih lanjut di bidang yang sama, dan berkontribusi dalam upaya memajukan ilmu pengetahuan dan teknologi.

Mataram, 31 Oktober 2023



Penulis

UCAPAN TERIMAKASIH

Tugas Akhir ini dapat diselesaikan berkat bimbingan, dukungan dan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih setulus–tulusnya terutama kepada:

1. Keluarga penulis, terutama kedua orang tua yaitu Ibu Siti Juwaeriah dan Bapak L. Muh. Isnaini serta ketujuh saudara-saudari penulis yaitu L. M. Kurnia Rizki, Bq. Putri Ajeng Lestari, Lalu Sahid Ramdani Mubaroq, Bq. Nada Latifa, Bq. Cahaya Nurul Baroroh, Lalu M. Bahtiar Hamzah dan L. M. Akbar Maulana.
2. Bapak A. Sjamsjiar Rachman, ST., MT. selaku Ketua Jurusan Teknik Elektro Universitas Mataram.
3. Bapak Lalu A. Syamsul Irfan A., S.T., M.Eng. dan Bapak Suthami Ariessaputra, ST., M.Eng. selaku Dosen Pembimbing yang telah memberikan bimbingan dan arahan selama penyusunan Tugas akhir ini dilakukan sehingga dapat diselesaikan.
4. Ibu Dra. Syamsinas Jafar, M. Hum. Dosen Jurusan Bahasa Indonesia FKIP Universitas Mataram yang telah membantu untuk menjadi pakar dalam pelabelan data pada penelitian Tugas Akhir ini.
5. Bapak Giri Wahyu Wiriasto, S.T., M.T., Bapak Cipta Ramadhani, ST., M.Eng., dan Ibu Dr. Ida Ayu Sri Adnyani, S.T., M.Erg. selaku Tim Dosen Penguji yang telah memberikan saran, kritik dan arahan untuk meningkatkan kualitas Tugas Akhir ini.
6. Jajaran Dosen Teknik Elektro Universitas Mataram yang telah banyak memberikan ilmu yang bermanfaat selama masa perkuliahan.
7. Teman-teman seperjuangan Teknik Elektro 2019 yang telah banyak membantu dan memotivasi penulis dalam menyelesaikan Tugas Akhir ini.
8. Rifa Utami Aripin yang selalu memberi dukungan dan semangat dalam penyelesaian Tugas Akhir ini.
9. Semua pihak lain yang tidak dapat disebutkan satu-persatu yang telah membantu baik dalam bentuk saran, kritik dan motivasi.

Semoga Allah Subhanahu Wa Ta'ala memberikan imbalan yang setimpal atas bantuan yang telah diberikan kepada penulis.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	iv
PRAKATA.....	v
UCAPAN TERIMA KASIH	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xii
ABSTRAK.....	xiii
<i>ABSTRACT</i>	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan Laporan	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori.....	7
2.2.1 <i>Machine Learning</i>	7
2.2.2 <i>Natural Language Processing (NLP)</i>	7
2.2.3 <i>Data Mining</i>	8
2.2.4 <i>Text Mining</i>	9
2.2.5 <i>Term Frequency-Inverse Document Frequency (TF-IDF)</i>	10
2.2.6 <i>Sentiment Analysis</i>	10
2.2.7 <i>Klasifikasi (Classification)</i>	11
2.2.8 <i>Performance Evaluation Measure (PEM)</i>	11
2.2.9 <i>Naïve Bayes Classifier (NBC)</i>	13
2.2.10 <i>Google Cloud Platform Console</i>	17
2.2.11 <i>RapidMiner</i>	18

2.2.12 <i>Jupyter Notebook</i>	18
2.2.13 <i>Streamlit</i>	18
BAB III METODE PENELITIAN	19
3.1 Alur Penelitian Tugas Akhir	19
3.2 Studi Literatur	20
3.3 Pengumpulan Data	20
3.4 <i>Labelling Dataset</i>	21
3.5 Perancangan Model	21
3.5.1 <i>Preprocessing</i> Data	21
3.5.2 Pembagian <i>Dataset</i>	25
3.5.3 Ekstraksi Fitur	26
3.5.4 Klasifikasi <i>Naïve Bayes</i>	26
3.2.7 Evaluasi Model	27
3.6 Perancangan Aplikasi Sentimen Analisis Berbasis <i>Web</i>	27
3.7 Pembahasan dan Kesimpulan	30
3.8 Alat Penelitian	31
BAB IV HASIL DAN PEMBAHASAN	32
4.1 Pengumpulan Data	32
4.1.1 <i>Create Credentials YouTube Data API Key</i>	32
4.1.2 <i>Implementasi Crawling Data Youtube Comments</i>	33
4.2 <i>Labelling Dataset</i>	35
4.3 Perancangan Model	37
4.3.1 <i>Preprocessing</i>	37
4.3.2 Pembagian <i>Dataset</i>	46
4.3.3 Ekstraksi Fitur TF-IDF	47
4.3.4 Klasifikasi <i>Naïve Bayes</i>	55
4.3.5 Evaluasi Model	64
4.3.5.1 Evaluasi Model Menggunakan 2 Kelas (Positif dan Negatif)	64
4.3.5.2 Evaluasi Model Menggunakan 3 Kelas (Positif, Negatif dan Netral)	72
4.4 Perancangan Aplikasi Sentimen Analisis Berbasis <i>Web</i>	83
4.4.1 Fitur <i>Crawling</i>	84
4.4.2 Fitur <i>Preprocessing</i>	85
4.4.3 Fitur <i>Classification</i>	89

BAB V KESIMPULAN DAN SARAN	94
5.1 Kesimpulan	94
5.2 Saran.....	95
DAFTAR PUSTAKA	96

DAFTAR TABEL

Tabel 2.1 <i>Confusion Matrix</i>	13
Tabel 2.2 Contoh data sentimen.....	15
Tabel 3.1 Detail keseluruhan jumlah data dengan kelas positif dan negatif.....	20
Tabel 3.2 Contoh Proses Penerapan <i>Cleansing</i>	22
Tabel 3.3 Contoh Proses Penerapan <i>Tokenizing</i>	24
Tabel 3.4 Contoh Proses Penerapan <i>Remove Stopword</i>	24
Tabel 3.5 Contoh Proses Penerapan <i>Stemming</i>	25
Tabel 3.6 Detail Perbandingan Pembagian data latih dan data uji.....	25
Tabel 4.1 Hasil <i>labelling</i> awal <i>dataset</i> secara manual.....	36
Tabel 4.2 Jumlah data <i>labelling</i> Positif dan Negatif secara manual.....	36
Tabel 4.3 Jumlah data <i>labelling</i> Positif dan Negatif setelah melalui <i>preprocessing</i>	37
Tabel 4.4 Hasil <i>preprocessing</i> pada tahap <i>cleansing</i>	38
Tabel 4.5 Hasil <i>preprocessing</i> pada tahap normalisasi.....	40
Tabel 4.6 Hasil <i>preprocessing</i> pada tahap <i>tokenizing</i>	41
Tabel 4.7 Hasil <i>preprocessing</i> pada tahap <i>stopword</i>	43
Tabel 4.8 Hasil <i>preprocessing</i> pada tahap <i>stemming</i>	44
Tabel 4.9 Hasil penerapan <i>join text</i>	45
Tabel 4.10 Perbandingan penggunaan data latih dan data uji.....	46
Tabel 4.11 Sampel daftar nilai pembobotan kata <i>Tf-Idf</i>	49
Tabel 4.12 Sampel <i>Dataset</i> komentar.....	49
Tabel 4.13 Sampel jumlah kemunculan fitur kata pada setiap dokumen.....	50
Tabel 4.14 Nilai perhitungan <i>Term Frequency</i> (TF).....	51
Tabel 4.15 Nilai perhitungan <i>Inverse Document Frequency</i> (IDF).....	53
Tabel 4.16 Hasil perhitungan <i>TF-IDF</i>	54
Tabel 4.17 Hasil prediksi klasifikasi model <i>Naïve Bayes</i>	56
Tabel 4.18 Sampel data latih dalam klasifikasi <i>Naïve Bayes Classifier</i>	57
Tabel 4.19 Sampel data uji dalam klasifikasi <i>Naïve Bayes Classifier</i>	57
Tabel 4.20 Sampel data uji pertama.....	58
Tabel 4.21 Sampel data uji kedua.....	60
Tabel 4.22 Sampel data uji ketiga.....	62

Tabel 4.23 Perbandingan model <i>Naïve Bayes</i> dengan fitur <i>Tf-Idf</i> dan tanpa fitur <i>Tf-idf</i> dengan 2 kelas (Positif dan Negatif).....	70
Tabel 4.24 Perbandingan model <i>Naïve Bayes</i> dengan fitur <i>Tf-Idf</i> dan tanpa fitur <i>Tf-idf</i> dengan 3 kelas (Positif, Netral dan Negatif).....	82

DAFTAR GAMBAR

Gambar 2.1 Proses <i>Knowledge Discovery in Database (KDD)</i>	8
Gambar 3.1 Alur Perencanaan Tugas Akhir.....	19
Gambar 3.2 Dataset hasil <i>labelling</i>	21
Gambar 3.3 Alur Proses Normalisasi.....	23
Gambar 3.4 Diagram Alir Sistem.....	27
Gambar 3.5 <i>Design UI</i> Halaman <i>Crawling Youtube Comment</i>	28
Gambar 3.6 <i>Design UI</i> Halaman <i>Preprocessing Data</i>	29
Gambar 3.7 <i>Design UI</i> Halaman <i>Classification Naïve Bayes</i>	30
Gambar 4.1 Tampilan halaman <i>project</i> pada <i>Console Google Cloud</i>	32
Gambar 4.2 Tampilan halaman <i>API & Services</i>	32
Gambar 4.3 <i>File excel dataset</i> hasil <i>crawling data komentar youtube</i>	35
Gambar 4.4 <i>File excel</i> kamus normalisasi kata <i>slang</i>	39
Gambar 4.5 Tampilan data ekstraksi fitur dengan pembobotan <i>TF-IDF</i>	48
Gambar 4.6 <i>Confusion Matrix Naïve Bayes</i> dengan <i>Tf-Idf</i> menggunakan 2 kelas.....	64
Gambar 4.7 <i>Confusion Matrix Naïve Bayes</i> tanpa <i>Tf-Idf</i> menggunakan 2 kelas.....	67
Gambar 4.8 <i>Confusion Matrix Naïve Bayes</i> dengan <i>Tf-Idf</i> menggunakan 3 kelas.....	72
Gambar 4.9 Ilustrasi penomoran petak <i>confusion matrix multiclass</i> dengan <i>Tf-Idf</i> ...	73
Gambar 4.10 <i>Confusion Matrix Naïve Bayes</i> tanpa <i>Tf-Idf</i> menggunakan 3 kelas.....	77
Gambar 4.11 Ilustrasi penomoran petak <i>confusion matrix multiclass</i> tanpa <i>Tf-Idf</i>	78
Gambar 4.12 Tampilan <i>website</i> sentimen analisis pada fitur <i>Crawling</i>	84
Gambar 4.13 Tampilan hasil <i>crawling data teks komentar</i>	85
Gambar 4.14 Tampilan <i>website</i> sentimen analisis pada fitur <i>Preprocessing</i>	85
Gambar 4.15 Tampilan hasil proses <i>cleansing</i> pada fitur <i>Preprocessing</i>	86
Gambar 4.16 Tampilan hasil proses normalisasi pada fitur <i>Preprocessing</i>	86
Gambar 4.17 Tampilan hasil proses <i>tokenize</i> pada fitur <i>Preprocessing</i>	87
Gambar 4.18 Tampilan hasil proses <i>stopwords</i> pada fitur <i>Preprocessing</i>	87
Gambar 4.19 Tampilan hasil proses <i>stemming</i> pada fitur <i>Preprocessing</i>	88
Gambar 4.20 Tampilan hasil dataset setelah <i>preprocessing</i> pada fitur <i>Preprocessin</i> ..	88
Gambar 4.21 Tampilan <i>website</i> sentimen analisis pada fitur <i>Classification</i>	89
Gambar 4.22 Tampilan ekstraksi fitur <i>Tf-Idf</i> pada fitur <i>Classification</i>	89
Gambar 4.23 Tampilan data hasil klasifikasi sentimen pada fitur <i>Classification</i>	90

Gambar 4.24 Tampilan grafik persentase klasifikasi sentimen pada fitur <i>Classification</i>	90
Gambar 4.25 Tampilan grafik <i>confusion matrix</i> pada fitur <i>Classification</i>	91
Gambar 4.26 Tampilan <i>classification report</i> pada fitur <i>Classification</i>	92

ABSTRAK

Perkembangan teknologi modern telah menciptakan kendaraan listrik, yang dianggap sebagai solusi ramah lingkungan dengan motor listrik dan baterai sebagai sumber tenaga. Maka analisis opini terhadap kendaraan listrik menjadi subjek penelitian yang menarik, dengan memanfaatkan data dari *Twitter* dan *YouTube*. Dalam upaya menganalisis opini masyarakat terhadap kendaraan listrik, algoritma *machine learning Naïve Bayes Classifier* digunakan untuk mengklasifikasikan probabilitas pendapat masyarakat menjadi sentimen positif atau negatif. Tujuan penelitian ini adalah untuk menilai persepsi masyarakat, mengevaluasi performa model klasifikasi analisis yang dibangun menggunakan algoritma *Naïve Bayes Classifier* dengan ekstraksi fitur *Term Frequency-Inverse Document Frequency (Tf-Idf)*, dan membangun aplikasi analisis opini berbasis *web* untuk klasifikasi, visualisasi, dan evaluasi data teks komentar. Data komentar dari *Twitter* dan *YouTube* diperoleh dengan teknik *crawling* dan melalui tahap *preprocessing* sebelum digunakan dalam analisis serta perancangan model dengan total data mencapai 3509. Model analisis sentimen dibangun dengan menggunakan algoritma *Naïve Bayes Classifier* dalam dua konfigurasi yaitu dengan ekstraksi fitur *Tf-Idf* (NBC TF-IDF) dan tanpa ekstraksi fitur *Tf-Idf* (NBC). Hasil penelitian menunjukkan bahwa model klasifikasi NBC TF-IDF dalam 2 kelas (Positif dan Negatif) mencapai akurasi sebesar 81%, sedangkan model klasifikasi menggunakan NBC mencapai akurasi sebesar 77%. Kemudian dalam skenario penggunaan 3 kelas (Positif, Netral, dan Negatif), model NBC TF-IDF mencapai akurasi 64%, sedangkan model NBC mencapai akurasi 62%. Model klasifikasi menggunakan NBC TF-IDF menunjukkan peningkatan performa yang lebih baik, baik dalam skenario 2 kelas maupun 3 kelas.

Kata kunci: Kendaraan listrik, *Crawling*, *Machine learning*, *Naïve Bayes Classifier*, *Term Frequency Inverse Document Frequency*

ABSTRACT

The development of modern technology has created electric vehicles, which are considered an environmentally friendly solution with electric motors and batteries as a power source. So the analysis of opinions on electric vehicles is an interesting research subject, utilizing data from Twitter and YouTube. In an effort to analyze public opinion on electric vehicles, the Naïve Bayes Classifier machine learning algorithm is used to classify the probability of public opinion into positive or negative sentiment. The purpose of this research is to assess public perception, evaluate the performance of the analysis classification model built using the Naïve Bayes Classifier algorithm with Term Frequency-Inverse Document Frequency (Tf-Idf) feature extraction, and build a web-based opinion analysis application for classification, visualization, and evaluation of comment text data. Comment data from Twitter and YouTube was obtained using crawling techniques and went through a preprocessing stage before being used in analysis and model design with a total of 3509 data. The sentiment analysis model was built using the Naïve Bayes Classifier algorithm in two configurations, namely with Tf-Idf feature extraction (NBC TF-IDF) and without Tf-Idf feature extraction (NBC). The results showed that the NBC TF-IDF classification model in 2 classes (Positive and Negative) achieved an accuracy of 81%, while the classification model using NBC achieved an accuracy of 77%. Then in the scenario of using 3 classes (Positive, Neutral, and Negative), the NBC TF-IDF model achieved an accuracy of 64%, while the NBC model achieved an accuracy of 62%. The classification model using NBC TF-IDF shows a better performance improvement, both in the 2-class and 3-class scenarios.

Keywords: *Electric vehicle, Crawling, Machine learning, Naïve Bayes Classifier, Term Frequency Inverse Document Frequency*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di era modern saat ini telah berkembang begitu pesat dengan membawa banyak perubahan di berbagai bidang, salah satunya bidang transportasi yang telah menciptakan inovasi baru yaitu kendaraan listrik. Kendaraan listrik adalah kendaraan yang menggunakan motor listrik sebagai sumber tenaga penggerakannya dan baterai sebagai penyimpanan energi. Kendaraan listrik memiliki potensi untuk menjadi solusi yang lebih ramah lingkungan karena mengurangi polusi udara dan emisi gas rumah kaca.

Dengan memanfaatkan data pada *platform* digital seperti *twitter* dan *Youtube* dapat dilakukan analisis opini publik terhadap kendaraan listrik dalam menganalisis pendapat, perasaan, penilaian, sikap dan emosi berdasarkan bahasa tulis atau teks komentar (Fatihin, 2022). Analisis opini publik terhadap kendaraan listrik ini menjadi topik yang menarik untuk diteliti dalam upaya menganalisis respon masyarakat berdasarkan data teks komentar dengan memanfaatkan penggunaan algoritma *machine learning* yaitu *Naïve Bayes Classifier* sebagai metode pengklasifikasian untuk memprediksi probabilitas pendapat masyarakat terhadap kendaraan listrik pada data teks komentar kedalam kelas sentimen positif atau negatif, sesuai dengan perhitungan probabilitas.

Beberapa penelitian terkait yang telah dilakukan mengenai analisis opini berdasarkan data teks komentar dengan algoritma *Naïve Bayes Classifier* diantaranya yaitu, Penelitian oleh (Irawansyah dkk, 2022) yang melakukan analisis sentimen terhadap program kampus merdeka pada *twitter* menggunakan algoritma *Naïve Bayes Classifier* yang dimana teknik *labelling* pada *dataset* dalam penelitian ini dilakukan secara otomatis dengan metode *TextBlob*. Namun, kekurangan dalam *labelling* data dengan *TextBlob* ini tidak *support* penuh terhadap data teks berbahasa indonesia. Kemudian, penelitian oleh (Toy dkk, 2021) yang melakukan penelitian menggunakan metode *Naïve Bayes* dengan *Relevance Frequency Feature Selection* terhadap kebijakan *New Normal*. Namun, kekurangan dari penelitian ini adalah penggunaan Metode *Naïve Bayes* dan *Relevance Frequency Feature Selection* menghasilkan akurasi yang masih terbilang rendah dikarenakan datah latih yang digunakan terlalu

sedikit dan kurang beragam. Selanjutnya, penelitian oleh (Riyadi dkk, 2022) yang melakukan pengukuran sentimen sosial terhadap teknologi kendaraan listrik untuk mengetahui bagaimana sentimen publik terhadap kendaraan listrik melalui komentar di media sosial *twitter* menggunakan analisis sentimen dengan metode klasifikasi *Naïve Bayes*, K-NN dan *Decision Tree*. Namun, pada penelitian ini visualisasi hasil dan analisis data masih dilakukan secara manual melalui *rapidminer*.

Berdasarkan uraian penelitian diatas, maka dalam penelitian ini penulis melakukan *Analysis Opini Publik Terhadap Kendaraan Listrik Menggunakan Algoritma Naïve Bayes Classifier Dengan Ekstraksi Fitur Tf-Idf*. Pada penelitian ini digunakan metode yang berbeda dalam menentukan kelas sentimen (*labelling data*) untuk *dataset*, yang dimana dilakukan *labelling* secara manual (*Manual Classifcation*) dengan bantuan dari pakar bahasa Indonesia yaitu Ibu Dra. Syamsinas Jafar, M. Hum yaang merupakan dosen jurusan Bahasa Indonesia FKIP Universitas Mataram. Kemudian, penelitian ini juga menggunakan ekstraksi fitur dengan metode *Tf-Idf* untuk melakukan perhitungan nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap fitur kata di setiap dokumen dalam *dataset* yang diharapkan mampu mendapatkan nilai akurasi yang lebih baik dengan pengklasifikasian pada algoritma *Naïve Bayes Classifier*.

Pemilihan metode *Naïve Bayes Classifier* pada penelitian ini didasari beberapa alasan seperti kemudahan dalam implementasinya, dan memiliki performa cepat dalam melatih model, serta dapat digunakan pada *dataset* yang lebih besar. Pada penelitian ini juga akan dilakukan implementasi model *mechine learning* kedalam bentuk sistem aplikasi analisis opini berdasarkan data teks komentar berbasis *web* menggunakan *framework streamlit*, sehingga proses klasifikasi, visualisasi dan evaluasi data analisis dapat ditampilkan dalam suatu sistem aplikasi berbasis *web*. Selain itu, dalam sistem yang akan dibuat juga nantinya akan di terapkan fitur *crawling* data komentar *youtube* dan *preprocessing dataset* komentar.

1.2 Rumusan Permasalahan

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka permasalahan yang akan dikaji dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana persepsi dan opini masyarakat terhadap kendaraan listrik?

- b. Bagaimana performa yang dihasilkan dari model klasifikasi analisis menggunakan algoritma *Naïve Bayes Classifier* dengan ekstraksi fitur *Tf-Idf*?
- c. Bagaimana penerapan model klasifikasi analisis dengan algoritma *Naïve Bayes Classifier* terkait opini publik terhadap kendaraan listrik?

1.3 Batasan Masalah

Untuk menjaga fokus dalam penelitian maka beberapa batasan yang diberikan dalam penelitian adalah sebagai berikut:

- a. Data yang digunakan adalah data opini atau teks komentar masyarakat di media sosial *twitter* dan *youtube* dengan rentang waktu data mulai dari 15 April 2021 sampai dengan 25 Mei 2023, yang dapat diakses pada link *github* (<https://github.com/ryasakbar060/Dataset-Komentar-Kendaraan-Listrik>).
- b. Penelitian ini akan berfokus pada teks komentar masyarakat yang berbahasa Indonesia.
- c. Metode yang akan digunakan untuk pengklasifikasian adalah metode algoritma *Naïve Bayes Classifier* dengan ekstraksi fitur *Tf-Idf*.
- d. *Output* dari penelitian ini adalah memberikan hasil analisis opini masyarakat terhadap kendaraan listrik menggunakan algoritma *Naïve Bayes Classifier* kedalam visualisasi sistem berbasis *web*.

1.4 Tujuan Penelitian

Adapun tujuan yang akan dicapai dari penelitian yang akan dilakukan adalah sebagai berikut:

- a. Untuk menilai persepsi masyarakat melalui analisis opini tentang kendaraan listrik berdasarkan teks komentar dengan *Naïve Bayes Classifier*.
- b. Untuk mengevaluasi performa model klasifikasi analisis yang dibangun menggunakan algoritma *Naïve Bayes Classifier* dengan ekstraksi fitur *Tf-Idf*.
- c. Untuk membangun sistem aplikasi analisis opini berbasis web untuk klasifikasi, visualisasi, dan evaluasi data teks komentar.

1.5 Manfaat Penelitian

Berdasarkan dengan permasalahan dan tujuan penelitian, maka penulis mengharapkan penelitian ini dapat memberikan beberapa manfaat antara lain:

- a. Memberikan pengetahuan kepada penulis terhadap kemampuan dan performa pengklasifikasian dengan metode *naïve bayes* pada bidang analisis sentimen.
- b. Memberikan pengetahuan dan pemahaman kepada penulis bagaimana cara mebgumpulkan, mengolah, dan mengubah data menjadi suatu informasi.
- c. Membuat sistem aplikasi berbasis *web* yang dapat digunakan untuk melakukan sentimen analisis terkait opini msyarakat terhadap kendaraan listrik dengan algoritma *naïve bayes classifier*.
- d. Diharapkan dapat menjadi acuan atau refrensi bagi penelitian selanjutnya apabila ingin mengembangkan dengan objek atau metode yang lainnya.
- e. Dapat menjadi infografis dalam bentuk data terkait sentimen masyarakat terhadap kendaraan listrik.

1.6 Sistematika Penulisan Laporan

Adapun sistematika penulisan dari laporan Tugas Akhir ini terbagi menjadi beberapa bab, sesuai dengan ketentuan yang ditetapkan di Fakultas Teknik, Universitas Mataram adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini berisi gambaran umum dan penjelasan mengenai latar belakang pemilihan judul penelitian, perumusan masalah yang dimana merupakan tulisan singkat berisi pertanyaan tentang topik penelitian yang nantinya akan dijawab oleh penulis sehingga penelitian yang dilakukan memiliki suatu kesimpulan dari hasil analisis dan visualisasi data penelitian yang dilakukan, selain itu terdapat pula batasan masalah yang merupakan batas-batas dari topik penelitian yang sedang dikaji atau diteliti, serta terdapat juga tujuan penelitian dan manfaat penelitian yang merupakan keuntungan yang didapat atau diperoleh oleh berbagai pihak dari penelitian yang dilakukan.

BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

Pada bab ini berisi mengenai landasan teori dan tinjauan pustaka, yang dimana dalam tinjauan pustaka mengulas beberapa penelitian sebelumnya yang sejenis mengenai analisis sentiment, sedangkan pada dasar teori membahas mengenai teori-teori yang berkaitan dengan analisis sentiment yang akan dilakukan pada penelitian yang didapatkan melalui beberapa sumber-sumber seperti jurnal, buku, dsb.

BAB III METODE PENELITIAN

Pada bab ini menjelaskan tentang langkah-langkah dalam pelaksanaan penelitian yang dimana terdiri dari alur penelitian yang membahas mengenai gambaran umum dalam bentuk diagram terkait alur penelitian yang dilakukan dan uraian metodologi yang membahas tahapan yang dilakukan pada alur penelitian diantaranya yaitu, pengambilan data, *preprocessing*, Pembagian Dataset, Ekstraksi Fitur, Klasifikasi *Naïve Bayes*, Evaluasi Model dan Perancangan aplikasi sentimen analisis berbasis *web*.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil dari penelitian yang telah dilakukan yang dimana terdiri dari hasil analisa, perancangan sistem aplikasi, dan klasifikasi yang dilakukan menggunakan algoritma *Naïve Bayes Classifier* dalam analisis sentimen.

BAB V KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan yang merupakan jawaban dari masalah yang dirumuskan dalam bentuk kalimat tanya pada rumusan masalah dan berisi saran yang membangun untuk pengembangan yang lebih baik di masa depan.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada tinjauan pustaka ini akan membahas mengenai beberapa penelitian-penelitian yang telah dilakukan sebelumnya yang relevan dan menjadi acuan daripada penelitian yang akan dilakukan oleh penulis.

Penelitian yang pertama adalah penelitian dengan judul analisis sentimen terhadap program merdeka belajar kampus merdeka (MBKM) pada *twitter* menggunakan algoritma *Naïve Byaes Classifier*. Dimana tujuan dari penelitian ini adalah untuk menganalisis bagaimana sentimen pada masyarakat tentang program merdeka belajar kampus merdeka dengan data yang diperoleh melalui *twitter* sebanyak 1499 data *tweets*, selain itu dalam penelitian ini juga *labelling* untuk dataset dilakukan secara otomatis dengan metode *TextBlob* dan menggunakan ekstraksi fitur *CountVectorizer* sebelum dilakukan pengklasifikasian dengan *Naïve Bayes* yang dimana hasil analisis sentiment yang diperoleh pada penelitian ini yaitu nilai akurasi sebesar 79,66%, Precision sebesar 79%, Recall sebesar 80% dan nilai F1-Score sebesar 79%. Sealin itu juga, pada penelitian ini hasil klasifikasi dan visualisasi dibuat dalam sebuah sistem analisis sentimen berbasis *web* (Irawansyah dkk, 2022).

Penelitian yang kedua adalah penelitian mengenai analisis sentimen *twitter* menggunakan metode *Naïve Bayes* dengan *Relevance Frequency Feature Selection* terhadap opini masyarakat mengenai kebijakan *new normal*. Tujuan dari penelitian ini yaitu untuk mengklasifikasikan sentimen masyarakat terhadap kebijakan *new normal* dengan penerapan metode *Naïve Bayes* dan penggunaan seleksi fitur RFFS (*Relevance Frequency Feature Selection*). Dalam penelitian ini digunakan *dataset* sebanyak 300 data opini masyarakat dengan pembagian data menggunakan *k-fold cross validation* dengan $k=5$. Hasil dari pengujian sebanyak 5 pengujian menggunakan klasifikasi *Naive Bayes*, diperoleh rata-rata akurasi sebesar 62,6%, sementara hasil pengujian akurasi klasifikasi dengan penambahan RFFS diperoleh rata-rata akurasi sebesar 65,3%. (Toy dkk, 2021).

Penelitian yang ketiga adalah penelitian dengan judul Pengukuran Sentimen Sosial Terhadap Teknologi Kendaraan Listrik. Tujuan dari penelitian ini adalah untuk mengetahui bagaimana sentimen publik terhadap kendaraan listrik melalui komentar

di *platform* media sosial *twitter* menggunakan analisis sentimen dengan metode klasifikasi *Naïve Bayes*, K-NN dan *Decision Tree*. Dalam penelitian ini data yang diperoleh dari *twitter* sebanyak 1084 *tweets*, yang dimana hasil dari analisis sentimen yang didapat menunjukkan bahwa metode klasifikasi *Naïve Bayes* memberikan hasil lebih baik dari K-NN dan *Decision Tree* dengan tingkat akurasi sebesar 94%, sentimen positif 53% dan sentimen negatif 38%. Adapun untuk proses visualisasi hasil analisis data dalam penelitian ini menggunakan aplikasi *rapidminer* untuk menampilkan grafik sentimen, *confusion matrix* dan *word cloud* (Riyadi dkk., 2022).

2.2 Dasar Teori

2.2.1 Machine Learning

Machine learning merupakan suatu ilmu yang membuat sistem dapat secara otomatis belajar sendiri tanpa harus berulang kali diprogram oleh manusia. *Machine Learning* sendiri merupakan salah satu disiplin ilmu dalam kecerdasan buatan atau yang sering biasa dikenal dengan *Artificial Intelligent* (AI). *Machine Learning* juga sering disebut dengan *Artificial Intelligent* (AI) konvensional karena merupakan kumpulan metode-metode yang digunakan dalam penerapannya. *Machine Learning* berfokus pada pengembangan program komputer yang dapat mengakses data dan menggunakannya untuk belajar sendiri. Sebelum *Machine Learning* bisa bekerja, maka ia membutuhkan data untuk *training* kemudian hasil dari *training* tersebut akan diuji atau dites dengan data yang sama atau bertolak belakang (Imron, 2019). Selain itu, metode dalam *Machine Learning* ini juga dapat menghitung akurasi. *Machine Learning* ini dikembangkan agar dapat mendeteksi pola, mengklasifikasikan pola, menghitung akurasi, serta membuat keputusan. Terdapat tiga metode *Machine Learning* yang sering digunakan, antara lain *supervised learning*, *unsupervised learning*, dan *reinforcement learning* (Khasanah, 2022).

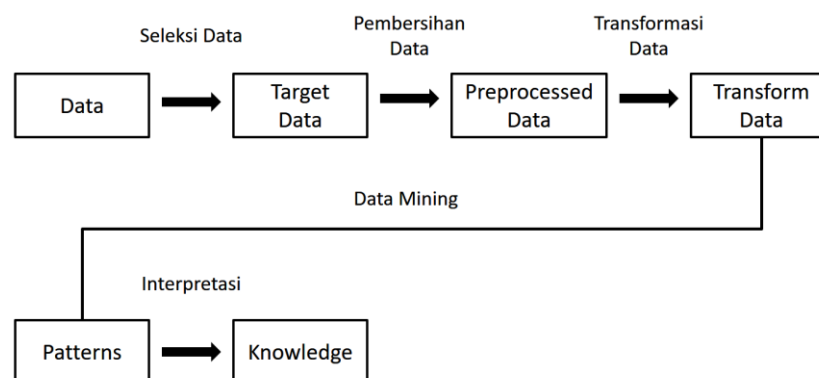
2.2.2 Natural Language Processing (NLP)

Natural Language Processing (NLP) didefinisikan sebagai kemampuan suatu komputer untuk memproses bahasa, baik lisan maupun tulisan yang digunakan oleh manusia dalam percakapan sehari-hari. Untuk proses komputasi, bahasa harus direpresentasikan sebagai rangkaian simbol yang memenuhi aturan tertentu. Secara sederhana, *Natural Language Processing (NLP)* adalah mencoba untuk membuat

komputer dapat mengerti perintah-perintah yang ditulis dalam standar bahasa manusia. Komputer dapat memahami bahasa alami manusia dengan cara membuat gambaran bahasa manusia dan mengubahnya menjadi angka-angka yang nantinya akan diproses dan dimasukkan ke dalam perhitungan dengan metode tertentu, yang kemudian akan menghasilkan respon kepada pengguna berupa tanggapan dari masukan yang telah diproses komputer sehingga membuat komputer terkesan dapat berinteraksi dengan pengguna menggunakan bahasa alami manusia (Imron, 2019).

2.2.3 Data Mining

Data mining merupakan metode pengolahan data berskala besar untuk menemukan pola yang tersembunyi dari data tersebut. Hasil dari pengolahan data dengan metode *data mining* dapat digunakan untuk mengambil keputusan di masa depan. *Data mining* sering juga disebut *Knowledge Discovery in Database (KDD)*, adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar. Dalam pelaksanaan aktivitas data mining sering kali digunakan berbagai teknik ataupun algoritma yang berasal dari berbagai disiplin ilmu misalnya statistik, *artificial intelligence* ataupun *machine learning*. Pada Gambar 2.1. dapat dilihat beberapa proses *Knowledge Discovery in Database (KDD)* (Pangestu, 2020).



Gambar 2.1. Proses *Knowledge Discovery in Database (KDD)*

1) Seleksi Data (*Data Selection*)

Pemilihan (seleksi) data dari sekumpulan data operasional perlu dilakukan sebelum tahap penggalian informasi dalam KDD dimulai. Data hasil seleksi yang digunakan untuk proses *data mining*, disimpan dalam suatu berkas, terpisah dari basis data operasional.

2) Pembersihan Data (*Data Cleaning*)

Pembersihan data merupakan proses menghilangkan *noise* dan data yang *inkonsisten* atau data tidak relevan. Pada umumnya data yang diperoleh, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Data-data yang tidak relevan tersebut lebih baik dibuang. Pembersihan data juga akan mempengaruhi performansi dari teknik data mining karena data yang ditangani akan berkurang jumlah dan komplektasinya.

3) Transformasi Data (*Data Transformation*)

Pada proses ini dilakukan transformasi bentuk data yang belum memiliki entitas yang jelas ke dalam bentuk data yang valid atau siap untuk dilakukan pada proses data mining

4) *Data Mining*

Data mining adalah proses mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode, atau algoritma dalam data mining sangat bervariasi. Pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara keseluruhan

5) Interpretasi (*Evaluation*)

Pada proses ini selanjutnya adalah menampilkan data tersebut dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Jadi, pola yang ditemukan akan diperiksa dan dicek apakah bertentangan dengan hipotesis sebelumnya atau tidak.

2.2.4 *Text Mining*

Text Mining merupakan penerapan konsep dari teknik data mining untuk mencari pola dalam teks, bertujuan untuk mencari informasi yang bermanfaat dengan tujuan tertentu. Berdasarkan ketidakteraturan struktur data teks, maka proses *text mining* memerlukan beberapa tahap awal yang pada intinya mempersiapkan agar teks dapat diubah menjadi lebih terstruktur. Proses *text mining* dibagi menjadi 3 tahap utama, yaitu proses awal terhadap teks (*text preprocessing*), transformasi teks (*text transformation*), dan penemuan pola (*pattern discovery*) (Pangestu, 2020).

2.2.5 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency Inverse Document Frequency (TF-IDF) adalah teknik penentuan seberapa *term* mewakili konten dalam dokumen dengan memberi bobot ke masing-masing kata yang terkandung di dalamnya. Nilai TF-IDF didapat dari perkalian TF dan IDF (Zhafira dkk, 2021). *Term frequency* (TF) adalah rasio dari jumlah kemunculan kata dalam dokumen dan *Inverse Document Frequency* (IDF) adalah kemunculan kata terhadap keseluruhan dokumen dalam data. Adapun rumus perhitungan *Tf-Idf* dapat dilihat sebagai berikut (Khomsah dkk, 2020):

- Rumus perhitungan TF

$$tf_{t,d} = \frac{n_{t,d}}{N} \quad (2-1)$$

Keterangan:

$n_{t,d}$ = Nilai istilah yang muncul

N = Total dokumen

tf = Frekuensi kemunculan kata pada sebuah dokumen

- Rumus perhitungan IDF

$$idf_d = \log \left(\frac{N}{df} \right) \quad (2-2)$$

Keterangan:

N = Total semua dokumen

df = Banyak dokumen yang mengandung term tersebut

- Rumus perhitungan TF-IDF

$$tfidf_{t,d} = tf_{t,d} \times idf_{t,d} \quad (2-3)$$

Keterangan:

$tf - idf$ = *Term Frequency-Inverse Document Frequency*

tf = Nilai TF

idf = Nilai IDF

2.2.6 Sentiment Analysis

Sentiment analysis atau *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *text mining* yang bertujuan menganalisis pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu. Sedangkan menurut Esuli &

Sebastiani, *Sentiment Analysis* merupakan sebuah penggambaran polaritas pada suatu teks atau kata. Jadi bisa disimpulkan bahwa sentimen analisis adalah pengolahan bahasa dari suatu teks atau kata yang dapat dianalisa dari pendapat, sikap atau penilaian dari seseorang terhadap suatu topik atau pembicaraan.

Sentiment analysis diterapkan di hampir setiap bisnis dan domain sosial karena opini adalah pusat dari hampir semua aktivitas manusia dan merupakan pengaruh utama dari perilaku kita, keyakinan, persepsi tentang kenyataan, dan pilihan yang dibuat, sebagian besar dikondisikan pada bagaimana orang lain melihat dan mengevaluasi dunia. Untuk alasan ini, ketika dalam membuat keputusan, kita sering mencari pendapat orang lain. Ini berlaku tidak hanya untuk individu tetapi juga untuk organisasi.

2.2.7 Klasifikasi (*Classification*)

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menggambarkan dan membedakan kelas data atau konsep dengan tujuan memprediksikan kelas untuk data yang tidak diketahui kelasnya. Di dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. Dalam melakukan klasifikasi data terdapat dua proses yang dilakukan yaitu:

1) Proses *Training*

Pada proses *training* digunakan training set yang telah diketahui label-labelnya untuk membangun model atau fungsi.

2) Proses *Testing*

Untuk mengetahui keakuratan model atau fungsi yang akan dibangun pada proses *training*, maka digunakan data yang disebut dengan *testing set* untuk memprediksi label-labelnya.

2.2.8 Performance Evaluation Measure (PEM)

Performance Evaluation Measure (PEM) atau dalam Bahasa Indonesia bisa disebut pengukuran evaluasi performa adalah satu tahapan yang digunakan untuk mengukur performa suatu sistem. PEM dalam banyak kasus digunakan dalam training data, tujuannya untuk mengevaluasi model yang sudah dibuat. Ada banyak

perhitungan untuk mendapatkan nilai PEM, biasanya diterapkan sebagai kombinasi atau juga secara parsial. Beberapa perhitungan dalam PEM antara lain (Imron, 2019):

a. *Precision*

Precision adalah tingkat ketepatan antara *request* pengguna dengan jawaban sistem.

Rumus *precision* (pre):

$$pre = \frac{TP}{FP+TP} \quad (2-4)$$

b. *Accuration*

Accuration adalah perbandingan antara informasi yang dijawab oleh sistem dengan benar dengan keseluruhan informasi.

Rumus *accuration* (acc):

$$acc = \frac{TN+TP}{FN+FP+TN+TP} \times 100\% \quad (2-5)$$

c. *Recall*

Recall adalah ukuran ketepatan antara informasi yang sama dengan informasi yang sudah pernah dipanggil sebelumnya.

Rumus *recall* (rec):

$$rec = \frac{TP}{FN+TP} \quad (2-6)$$

d. *F1-Score*

F1-score merupakan nilai perbandingan rata – rata antara *recall* dan *precision*.

Rumus *F1-Score*:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2-7)$$

Performance Evaluation Measure (PEM) biasanya digambarkan dalam *confusion matrix*, yaitu berupa tabel yang berisi hasil pengujian model yang telah dibandingkan dengan dataset, terdiri dari kelas true dan false.

Tabel 2.1. *Confusion Matrix*

<i>True Class</i>	<i>Predicted Class</i>	
	<i>Positive</i>	<i>Negative</i>
<i>Positive</i>	TP	FN
<i>Negative</i>	FP	TN

Keterangan:

TP (*true positive*) : contoh data bernilai positif yang diprediksi benar sebagai positif

TN (*true negative*) : contoh data bernilai negatif yang diprediksi benar sebagai negatif

FP (*false positive*) : contoh data bernilai negatif yang diprediksi salah sebagai positif

FN (*false negative*) : contoh data bernilai positif yang diprediksi salah sebagai negatif

2.2.9 *Naïve Bayes Classifier (NBC)*

Naive Bayes Classifier adalah salah satu algoritma klasifikasi yang digunakan dalam *machine learning* dan *data mining*. Algoritma ini didasarkan pada teorema *Bayes* dengan asumsi sederhana (*naive*), yaitu asumsi bahwa setiap fitur atau atribut dari data yang digunakan untuk melakukan klasifikasi adalah independen satu sama lain. Teorema ini bekerja dengan cara mempelajari pola dari data latih yang sudah diberikan label klasifikasi, lalu menggunakan pola tersebut untuk melakukan klasifikasi pada data uji yang belum diketahui labelnya. Algoritma ini cocok digunakan untuk mengklasifikasikan data dengan jumlah atribut yang besar dan terdistribusi secara acak.

Keuntungan penggunaan *Naive Bayes* adalah bahwa metode ini hanya membutuhkan jumlah data latih (*Training Data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. *Naive Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.

Naive Bayes Classifier juga merupakan salah satu metode yang populer digunakan pada *data mining* karena kemudahan penggunaannya serta waktu pemrosesannya yang cepat, mudah diimplementasikan dengan struktur yang cukup sederhana dan tingkat efektifitas yang tinggi. Metode ini menggunakan aturan *Bayesian*. *Bayesian Classification* merupakan klasifikasi secara statistik dengan memprediksi suatu data terprediksi ke dalam kelas tertentu (Siswandi, 2019).

Rumus *Naïve Bayes* secara umum dapat diberikan sebagai berikut :

$$P(H | X) = \frac{P(X | H) P(H)}{P(X)} \quad (2-8)$$

Keterangan:

X = Data dengan kelas yang belum diketahui

H = Hipotesis data X merupakan suatu class spesifik.

$P(H|X)$ = Probabilitas hipotesis H berdasarkan kondisi X (*posterior probability*)

$P(H)$ = Probabilitas hipotesis H sebelum data x diperoleh (*prior probability*)

$P(X/H)$ = Probabilitas dari data X diberikan hipotesis H (*likelihood*)

$P(X)$ = Probabilitas dari data X

Untuk menerapkan rumus persamaan metode *Naïve Bayes* diatas, terdapat beberapa langkah-langkah yang dapat dilakukan diantaranya sebagai berikut:

- 1) Menghitung probabilitas atau jumlah dokumen pada setiap kelas misalnya kelas “Positif” (H_Positif) dan kelas “Negatif” (H_Negatif) dalam data latih.
- 2) Menghitung probabilitas prior (P(H)) dengan persamaan berikut.

$$P(H_Positif) = \frac{\text{Jumlah dokumen Positif}}{\text{Total Dokumen}} \quad (2-9)$$

$$P(H_Negatif) = \frac{\text{Jumlah dokumen Negatif}}{\text{Total Dokumen}} \quad (2-10)$$

- 3) Menghitung probabilitas *likelihood* (P(X | H)), yang dimana untuk menghitung probabilitas *likelihood*, kita perlu menghitung kemunculan kata-kata dalam setiap dokumen dengan persamaan berikut.

$$P(X | H_Negatif) = \frac{\text{Jumlah kata dokumen Negatif}+1}{\text{Total kata dokumen Negatif}+\text{Total kata semua dokumen}} \quad (2-11)$$

$$P(X | H_Positif) = \frac{\text{Jumlah kata dokumen Positif}+1}{\text{Total kata dokumen Positif}+\text{Total kata semua dokumen}} \quad (2-12)$$

- 4) Menghitung probabilitas total (P(X)) untuk setiap kelas sentimen (Positif dan Negatif) berdasarkan data latih dengan persamaan berikut.

$$P(X) = P(X | H_Positif) \times P(H_Positif) + P(X | H_Negatif) \times P(H_Negatif) \quad (2-13)$$

- 5) Mengklasifikasikan dokumen uji untuk kelas Positif (H_{Positif}) dan Negatif (H_{Negatif}) dari perhitungan probabilitas posterior $P(H/X)$ menggunakan rumus persamaan Naïve Bayes seperti pada rumus (2-8).

Berikut adalah contoh proses perhitungan klasifikasi kelas sentimen dengan menggunakan metode *Naive Bayes*.

Tabel 2.2 Contoh data sentimen

No	Data	Kelas
D1	Kendaraan listrik mahal	Negatif
D2	Indonesia belum siap kendaraan listrik	Negatif
D3	Kendaraan listrik bikin susah	Negatif
D4	Kendaraan listrik ribet	Negatif
D5	Kendaraan listrik bagus	Positif
D6	Kendaraan listrik sangat ramah lingkungan	Positif
D7	Kendaraan listrik keren banget	Positif
D8	Isi daya Kendaraan listrik masih susah	?

- Menghitung probabilitas dalam dokumen data latih pada setiap kelas positif dan negatif berdasarkan tabel 2.2.
Diketahui:
 - Jumlah dokumen kelas Positif = 3
 - Jumlah dokumen kelas Negatif = 4
 - Total semua dokumen = 7
 - Total kata dalam dokumen kelas Positif = 12
 - Total kata dalam dokumen kelas Negatif = 15
 - Total semua kata dalam dokumen = 27
- Menghitung probabilitas prior ($P(H)$) dengan menggunakan rumus persamaan (2-9) dan (2-10)
 - $P(\text{Positif}) = \frac{3}{7} = 0,428$
 - $P(\text{Negatif}) = \frac{4}{7} = 0,571$
- Menghitung probabilitas *likelihood* ($P(X|H)$) dari dokumen uji dengan menggunakan rumus persamaan (2-11) dan (2-12).

- Kelas Positif

$$P(\text{isi} | \text{Positif}) = \frac{0+1}{12+27} = 0,026$$

$$P(\text{daya} | \text{Positif}) = \frac{0+1}{12+27} = 0,026$$

$$P(\text{kendaraan} | \text{Positif}) = \frac{3+1}{12+27} = 0,102$$

$$P(\text{listriik} | \text{Positif}) = \frac{3+1}{12+27} = 0,102$$

$$P(\text{masih} | \text{Positif}) = \frac{0+1}{12+27} = 0,026$$

$$P(\text{susah} | \text{Positif}) = \frac{0+1}{12+27} = 0,026$$

- Kelas Negatif

$$P(\text{isi} | \text{Negatif}) = \frac{0+1}{15+27} = 0,024$$

$$P(\text{daya} | \text{Negatif}) = \frac{0+1}{15+27} = 0,024$$

$$P(\text{kendaraan} | \text{Negatif}) = \frac{4+1}{15+27} = 0,119$$

$$P(\text{listriik} | \text{Negatif}) = \frac{4+1}{15+27} = 0,102$$

$$P(\text{masih} | \text{Negatif}) = \frac{0+1}{15+27} = 0,024$$

$$P(\text{susah} | \text{Negatif}) = \frac{1+1}{15+27} = 0,047$$

- Menghitung probabilitas total ($P(X)$) untuk kelas Negatif ($P(X | \text{Negatif})$) dan kelas Positif ($P(X | \text{Positif})$) kemudian menghitung probabilitas total $P(X)$ dengan persamaan (2-13).

- Positif

$$P(X | \text{Positif}) = 0,026 \times 0,026 \times 0,102 \times 0,102 \times 0,026 \times 0,026$$

$$P(X | \text{Positif}) = 4,754$$

- Negatif

$$P(X | \text{Negatif}) = 0,024 \times 0,024 \times 0,119 \times 0,102 \times 0,024 \times 0,047$$

$$P(X | \text{Negatif}) = 7,886$$

- Probabilitas Total $P(X)$

$$P(X) = 4,754 \times 0,428 + 7,886 \times 0,571$$

$$P(X) = 2,035 + 4,502$$

$$P(X) = 6,537$$

- Mengklasifikasikan dokumen uji untuk kelas Positif dan Negatif dari perhitungan probabilitas posterior $P(H/X)$ menggunakan rumus persamaan Naïve Bayes (2-8).

- Positif

$$P(\text{Positif} | X) = \frac{P(X | \text{Positif}) P(\text{Positif})}{P(X)}$$

$$P(\text{Positif} | X) = \frac{4,754 \times 0,428}{6,537}$$

$$P(\text{Positif} | X) = 0,31126$$

- Negatif

$$P(\text{Negatif} | X) = \frac{P(X | \text{Negatif}) P(\text{Negatif})}{P(X)}$$

$$P(\text{Negatif} | X) = \frac{7,886 \times 0,571}{6,537}$$

$$P(\text{Negatif} | X) = 0,68883$$

Dari perhitungan berdasarkan nilai probabilitas posterior diatas, maka diperoleh hasil 0,31126 untuk kelas Positif dan 0,68883 untuk kelas Negatif. Karena nilai probabilitas posterior pada kelas negatif memiliki nilai lebih besar dibandingkan dengan nilai probabilitas posterior kelas positif. Maka, data uji pada D8 terklasifikasi kedalam kelas Negatif.

2.2.10 Google Cloud Platform Console

Google Cloud Platform Console (GCP Console) adalah platform cloud computing yang dikembangkan dan dioperasikan oleh Google. GCP Console menawarkan berbagai layanan dan produk cloud, termasuk komputasi, penyimpanan, basis data, dan pemrosesan data. GCP Console dirancang untuk memungkinkan organisasi dan pengembang membangun, mengelola, dan meningkatkan aplikasi dan layanan mereka dengan mudah dan skalabilitas yang tinggi.

Salah satu keuntungan utama dari GCP Console adalah keandalannya. GCP Console menyediakan infrastruktur yang andal dan aman yang dikelola oleh Google, yang memiliki pengalaman dalam mengelola infrastruktur internet skala besar. Selain itu, GCP Console juga menawarkan layanan data analisis seperti BigQuery dan Dataflow, yang memungkinkan pengguna untuk mengumpulkan dan menganalisis data dalam skala besar dengan mudah (Carissa, 2023).

2.2.11 RapidMiner

RapidMiner merupakan perangkat lunak yang bersifat terbuka (*open source*). *RapidMiner* adalah sebuah solusi untuk melakukan analisis terhadap *data mining*, *text mining* dan analisis prediksi. *RapidMiner* menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik. *RapidMiner* memiliki kurang lebih 500 operator *data mining*, termasuk operator untuk *input*, *output*, *data preprocessing* dan visualisasi. *RapidMiner* merupakan *software* yang berdiri sendiri untuk analisis data dan sebagai mesin data mining yang dapat diintegrasikan pada produknya sendiri.

2.2.12 Jupyter Notebook

Jupyter Notebook adalah aplikasi web untuk membuat dan berbagi dokumen komputasi. Ini menawarkan pengalaman yang sederhana, efisien, dan berpusat pada dokumen. Antarmukanya yang fleksibel memungkinkan pengguna untuk mengkonfigurasi dan mengatur alur kerja dalam ilmu data, komputasi ilmiah, jurnalisme komputasi, dan *Machine Learning*. *Jupyter Notebook* sendiri merupakan sebuah format publikasi untuk alur kerja komputasi yang dapat ditiru/dieksekusi orang lain, atau secara sederhana, kode program yang ter-embed dapat dieksekusi dan dimodifikasi oleh orang lain. Oleh karena sifat publikasi yang *reproducible* tersebut, maka *Jupyter Notebook* juga dapat dimanfaatkan dalam mendukung kegiatan belajar mengajar mata kuliah-mata kuliah di bidang keteknikan (*Jupyter*, 2023).

2.2.13 Streamlit

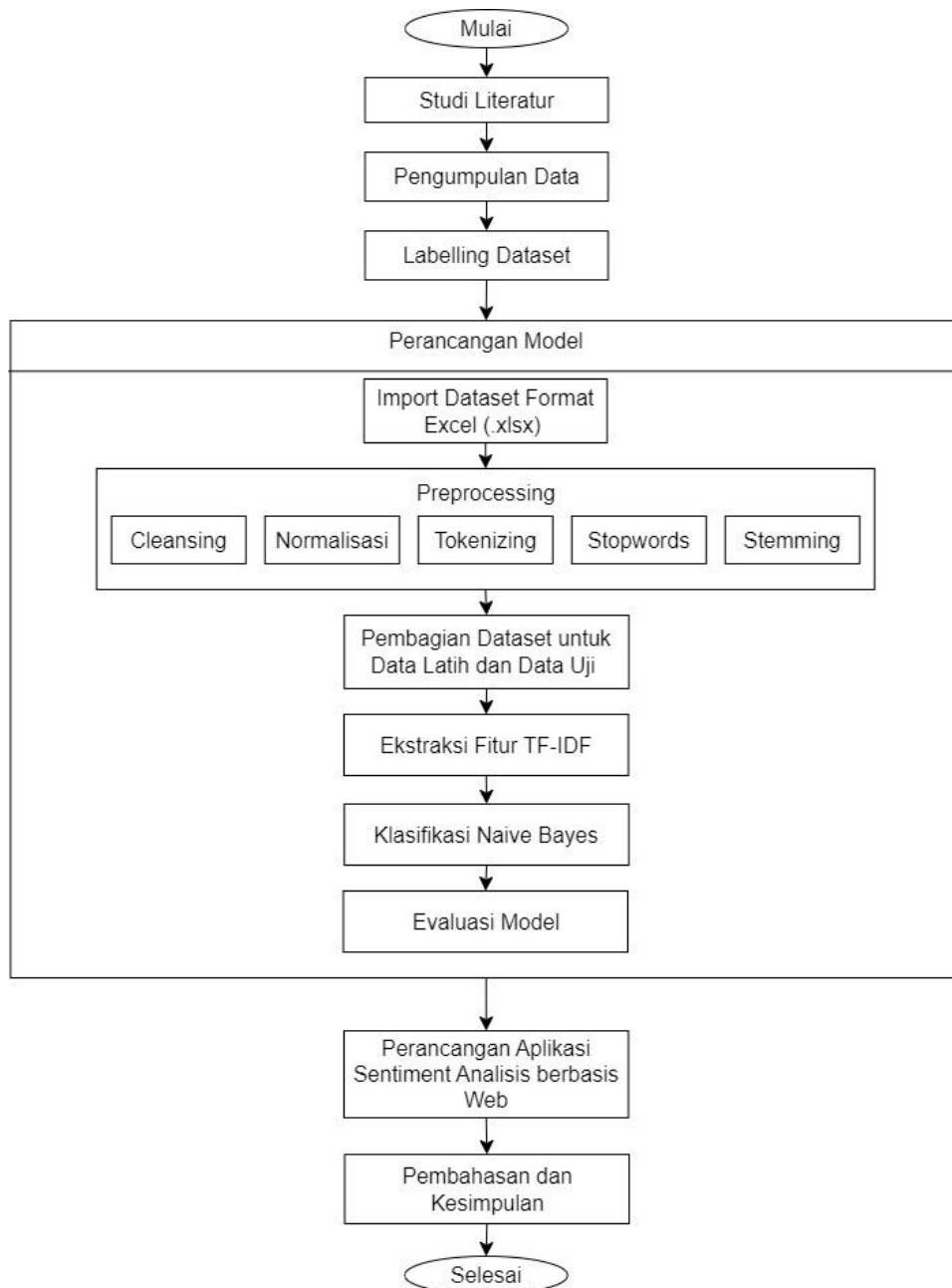
Streamlit adalah sebuah *framework* berbasis *Python* dan bersifat *open-source* yang dibuat untuk memudahkan dalam membangun aplikasi *web* di bidang sains data dan *machine learning* yang interaktif. *Streamlit* juga dapat didefinisikan sebagai kerangka kerja web yang ditujukan untuk menyebarkan model dan visualisasi dengan mudah menggunakan bahasa *Python*, yang cepat dan minimalis tetapi juga memiliki tampilan yang cukup baik serta ramah pengguna. Tersedia *widget* bawaan untuk masukan pengguna, seperti pengunggahan gambar, penggeser, masukan teks, dan elemen *hypertext markup language (HTML)* lain yang sudah dikenal, seperti *checkboxes* dan *radio buttons*.

BAB III

METODE PENELITIAN

3.1 Alur Penelitian Tugas Akhir

Perancangan alur dari pengerjaan tugas akhir ini merupakan gambaran umum terkait alur penelitian yang akan dilakukan dalam penelitian tugas akhir mulai dari awal hingga akhir. Alur kerja dari penelitian tugas akhir dapat dilihat pada gambar 3.1. berikut:



Gambar 3.1. Alur Perencanaan Tugas Akhir

3.2 Studi Literatur

Pada penelitian ini, dilakukan studi literatur sebagai tahapan awal untuk melakukan penelitian. Tujuan dari studi literatur ini adalah untuk memahami konsep dan dasar teori yang akan menjadi pembanding dan pendukung dalam penelitian berjudul “*Analysis Opini Publik Terhadap Kendaraan Listrik Menggunakan Algoritma Naïve Bayes Classifier (NBC) Dengan Ekstraksi Fitur TF-IDF*”

3.3 Pengumpulan Data

Pada penelitian ini data yang digunakan adalah data komentar dari *twitter* dengan *crawling* melalui aplikasi *rapidminer*. Selain itu, data juga dikumpulkan dari komentar video *youtube* mengenai respon atau opini masyarakat terhadap kendaraan listrik. Data yang dikumpulkan berupa data teks yang diambil dengan teknik *crawling* data menggunakan *youtube API v3* yang disediakan oleh *Google* dan menggunakan *library googleapiclient.discovery* yang memungkinkan untuk berinteraksi dengan *youtube Data API* dan mengambil data komentar dari video *youtube*. *Datset* hasil *crawling* dapat diakses pada *link github* (<https://github.com/ryasakbar060/Dataset-Komentar-Kendaraan-Listrik>). Data komentar yang diambil merupakan data komentar dari beberapa video *youtube* yang membahas mengenai teknologi kendaraan listrik dan perkembangan teknologi tersebut. Variabel meta data hasil *crawling* yang akan diambil adalah *username*, teks komentar serta tanggal dan waktu komentar. Hasil dari *crawling* data *twitter* dan *youtube* disimpan dalam *file* dengan format *excel (.xlsx)*.

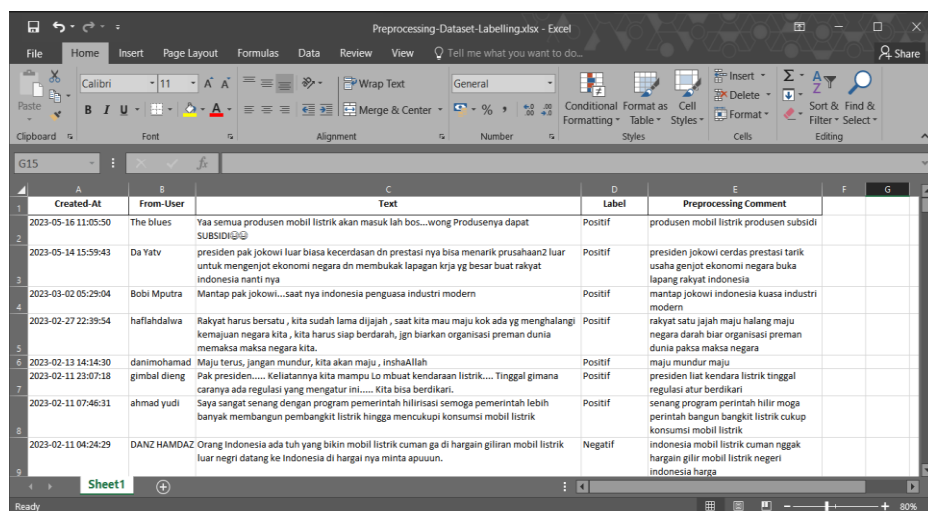
Jumlah data awal hasil *crawling* pada penelitian ini sebanyak 4270 data yang terdiri dari tiga kelas sentimen yaitu Positif, Naegatif dan Netral. Namun, dalam penelitian ini kelas sentimen yang digunakan hanya kelas Positif dan Negatif, sehingga banyaknya data setelah kelas netral dihilangkan menjadi sebanyak 3592 data. Kemudian, dari 3592 data tersebut akan dibersihkan terlebih dahulu pada tahap *preprocessing* seperti *cleansing*, *normalisasi*, *tokenizing*, *stopwords* dan *stemming* sehingga menghasilkan data bersih sebesar 3509. Berikut detail jumlah data bersih yang akan digunakan dalam penelitian ini dapat dilihat pada tabel 3.1.

Tabel 3.1. Datail keseluruhan jumlah data dengan kelas positif dan negatif

Positif	Negatif	Total Seluruh Data
1904	1605	3509

3.4 Labelling Dataset

Pada proses *labelling dataset* dilakukan pelabelan kelas sentimen pada data ulasan komentar yang telah didapatkan menjadi kelas positif, negatif dan netral. Pada proses ini dilakukan pelabelan data ulasan secara manual (*manual classification*) dengan bantuan dari pakar bahasa yaitu Ibu Dra. Syamsinas Jafar, M. Hum yang merupakan Dosen Jurusan Bahasa Indonesia FKIP Universitas Matram. *Manual classification* merupakan pengkategorian yang dilakukan dengan membaca data komentar atau ulasan satu per satu kemudian menentukan apakah komentar tersebut termasuk ke sentimen positif atau negatif. Keuntungan dari klasifikasi tersebut adalah hasil yang diperoleh lebih sesuai dengan kenyataan dan tidak memerlukan perhitungan lebih lanjut (Fatihin, 2022). Berikut dapat dilihat pada Gambar 3.2 *dataset* setelah dilakukan *labelling* data dalam *file* dengan format *excel*.



Created-At	From-User	Text	Label	Preprocessing Comment
2023-05-16 11:05:50	The blues	Yaa semua produsen mobil listrik akan masuk lah bos...wong Produsennya dapat SUBSIDIGI	Postif	produsen mobil listrik produsen subsidi
2023-05-14 15:59:43	Da Yatv	presiden pak jokowi luar biasa kecerdasan dn prestasi nya bisa menarik prusahaan2 luar untuk mengenjot ekonomi negara dn membukak lapangan krja yg besar buat rakyat indonesia nanti nya	Postif	presiden jokowi cerdas prestasi tarik usaha genjot ekonomi negara buka lapang rakyat indonesia
2023-03-02 05:29:04	Bobi Mputra	Mantap pak jokowi...saat nya indonesia penguasa industri modern	Postif	mantap jokowi indonesia kuasa industri modern
2023-02-27 22:39:54	hafflahdwa	Rakyat harus bersatu , kita sudah lama dijajah , saat kita mau maju kok ada yg menghalangi kemajuan negara kita , kita harus siap berdarah, jgn biarkan organisasi preman dunia memaksa maksa negara kita.	Postif	rakyat satu jajah maju halang maju negara darah biar organisasi preman dunia paksa maksa negara
2023-02-13 14:14:30	danimohamad	Maju terus, jangan mundur, kita akan maju , insyaAllah	Postif	maju mundur maju
2023-02-11 23:07:18	gimbal dieng	Pak presiden..... Keliatannya kita mampu Lo mbuat kendaraan listrik.... Tinggal gimana caranya ada regulasi yang mengatur ini..... Kita bisa berdikari.	Postif	presiden liat kendaraan listrik tinggal regulasi atur berdikari
2023-02-11 07:46:31	ahmad yudi	Saya sangat senang dengan program pemerintah hilirisasi semoga pemerintah lebih banyak membangun pembangkit listrik hingga mencukupi konsumsi mobil listrik	Postif	senang program perintah hilir moga perintah bangun bangkit listrik cukup konsumsi mobil listrik
2023-02-11 04:24:29	DANZ HAMDAR	Orang Indonesia ada tuh yang bikin mobil listrik cuman ga di hargain girilan mobil listrik luar negri datang ke Indonesia di hargai nya minta apuun.	Negatif	Indonesia mobil listrik cuman nggak hargain giril mobil listrik negeri Indonesia harga

Gambar 3.2. *Dataset* hasil *labelling*

3.5 Perancangan Model

3.5.1 Preprocessing Data

Tujuan dari tahap ini adalah untuk membersihkan dataset komentar hasil *crawling* sehingga menjadi data yang bersih dan siap dilakukan pengolahan dalam analisis data. Langkah-langkah *preprocessing* yang dilakukan disesuaikan berdasarkan data mengenai komentar atau opini masyarakat terhadap kendaraan listrik melalui kolom kementar pada video *youtube* dan *twitter*. Adapun beberapa tahap *preprocessing* yang dilakukan adalah:

a) *Cleansing*

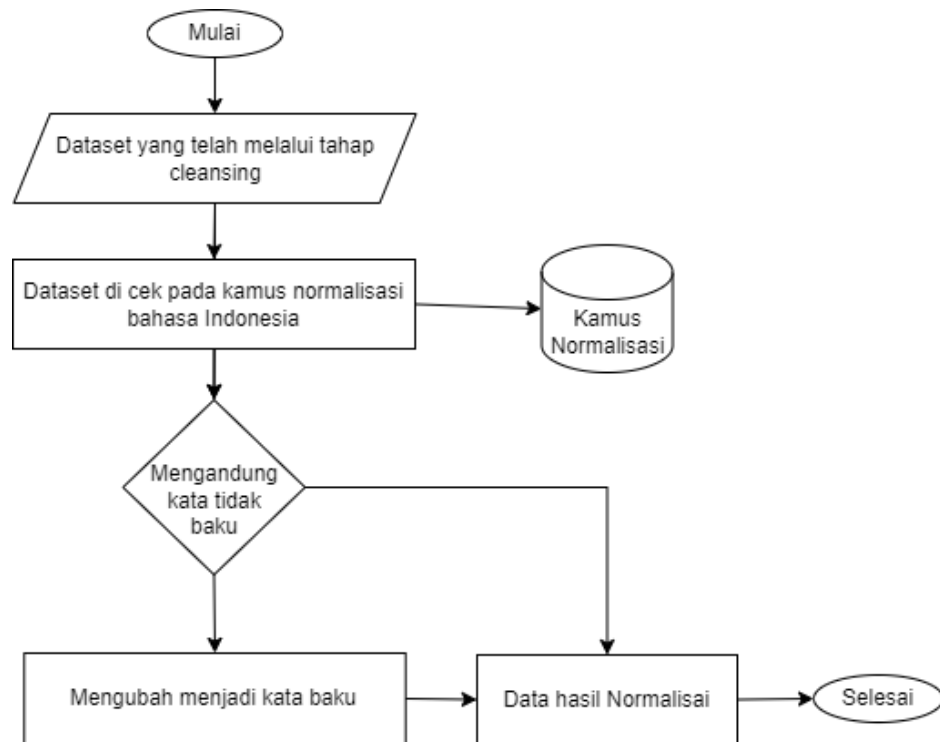
Pada tahap *cleansing*, elemen-elemen karakter spesial seperti *hashtag*, *mention username*, *URL*, tanda baca, *double space* (spasi berlebih), *emoticon* yang membuat data menjadi tidak efektif dan tidak memiliki arti akan dihapus dari data komentar pada *dataset*. Selain itu, simbol serta bilangan angka juga akan dihapus dari data teks komentar. Pada proses *cleansing* juga akan dilakukan tahap *case folding* untuk merubah komentar menjadi *lower-case* yang tujuannya adalah untuk mengurangi kesalahan acak (*noise*) pada data. Sehingga dalam tahapan *cleansing* nantinya akan digunakan beberapa modul pada pemrograman *python* yaitu modul *string* yang digunakan untuk menghilangkan karakter spesifik seperti karakter spasi, tanda baca, dan karakter lainnya yang tidak diperlukan dalam data teks serta modul *regex* (*regular expression*) yang dapat digunakan untuk menghapus karakter-karakter tertentu dalam data teks komentar, misalnya menghapus semua tautan *URL* atau menghapus semua karakter *non-huruf* dalam komentar. Contoh penerapan tahap *cleansing* dapat dilihat pada Tabel 3.1.

Tabel 3.2. Contoh Proses Penerapan Cleansing

Before	After
Saya sangat senang dengan program pemerintah hilirisasi semoga pemerintah lebih banyak membangun pembangkit listrik hingga mencukupi konsumsi mobil listrik	saya sangat senang dengan program pemerintah hilirisasi semoga pemerintah lebih banyak membangun pembangkit listrik hingga mencukupi konsumsi mobil listrik
Orang Indonesia ada tuh yang bikin mobil listrik cuman ga di hargain giliran mobil listrik luar negri datang ke Indonesia di hargai nya minta apuun.	orang indonesia ada tuh yang bikin mobil listrik cuman ga di hargain giliran mobil listrik luar negri datang ke indonesia di hargai nya minta apuun

b) *Normalisasi*

Proses normalisasi adalah proses mengubah kata *slang* atau tidak baku menjadi bahasa yang baku sesuai Kamus Besar Bahasa Indonesia (KBBI). Proses ini dilakukan dengan cara mengganti kata *slang* yang dideteksi dengan menggunakan acuan pada *dataset* normalisasi kamus bahasa *slang* bertipe *.xlsx* yang kemudian akan digantikan dengan kata yang sesuai KBBI. Setelah itu disimpan kembali untuk diproses lebih lanjut pada tahap berikutnya. Berikut Gambar 3.2 menunjukkan proses dari normalisasi kata.



Gambar 3.3. Alur Proses Normalisasi

c) *Tokenizing*

Tokenizing adalah proses memecah teks atau kalimat menjadi unit-unit yang lebih kecil, yang disebut token. Token berupa kata, frasa, atau elemen lain yang memiliki makna atau representasi tertentu dalam bahasa. Proses tokenisasi adalah langkah awal yang penting dalam analisis data sentimen karena memungkinkan kita untuk mengubah teks menjadi bentuk yang lebih terstruktur, yang kemudian dapat diolah lebih lanjut untuk memahami sentimen atau makna yang terkandung di dalamnya. Dalam penerapan pada penelitian atau analisis data sentimen, tokenisasi membantu untuk mengatasi kompleksitas teks dan mengubahnya menjadi representasi yang lebih terstruktur, yang memudahkan untuk dilakukan analisis lebih lanjut, seperti klasifikasi sentimen atau penambangan pola-pola tertentu dalam teks. Pada implementasinya terdapat salah satu modul pemrograman *python* yang akan digunakan pada penelitian ini yaitu modul *Natural Language Toolkit* (NLTK) yang merupakan *library python open-source* pada bahasa pemrograman *Python* yang berfokus pada pengolahan bahasa alami. Contoh penerapan pada tahap *tokenizing* dapat dilihat pada Tabel 3.3.

Tabel 3.3. Contoh Proses Penerapan *Tokenizing*

Before	After
saya sangat senang dengan program pemerintah hilirisasi semoga pemerintah lebih banyak membangun pembangkit listrik hingga mencukupi konsumsi mobil listrik	['saya', 'sangat', 'senang', 'dengan', 'program', 'pemerintah', 'hilirisasi', 'semoga', 'pemerintah', 'lebih', 'banyak', 'membangun', 'pembangkit', 'listrik', 'hingga', 'mencukupi', 'konsumsi', 'mobil', 'listrik']
orang indonesia ada tuh yang bikin mobil listrik cuman ga di hargain giliran mobil listrik luar negri datang ke indonesia di hargai nya minta apuun	['orang', 'indonesia', 'ada', 'tuh', 'yang', 'bikin', 'mobil', 'listrik', 'cuman', 'nggak', 'di', 'hargain', 'giliran', 'mobil', 'listrik', 'luar', 'negeri', 'datang', 'ke', 'indonesia', 'di', 'hargai', 'nya', 'minta', 'apuun']

d) *Stopwords Removal*

Pada tahap ini kata-kata yang kurang bermakna atau tidak memiliki arti dilakukan penghapusan seperti kata: “dan”, “atau”, “yang”, “di”, dan sebagainya. Tujuan dari tahap *stopwords removal* adalah membuat data menjadi lebih bersih dan meminimalisir terjadinya *noise* pada data. Pada penelitian ini proses *stopwords removal* menggunakan *library* NLTK yang sudah terdapat *sub-library* khusus untuk *remove stopwords*. Selain kamus *remove stopwords* yang sudah terdapat pada NLTK, pada penelitian ini penulis juga menambahkan kamus *stopwords* bahasa Indonesia berupa data bertipe *.txt* sebagai acuan dalam melakukan tahap *stopwords removal*. Contoh penerapan pada tahap *Stopwords Removal* dapat dilihat pada Tabel 3.4.

Tabel 3.4. Contoh Proses Penerapan Remove Stopword

Before	After
['saya', 'sangat', 'senang', 'dengan', 'program', 'pemerintah', 'hilirisasi', 'semoga', 'pemerintah', 'lebih', 'banyak', 'membangun', 'pembangkit', 'listrik', 'hingga', 'mencukupi', 'konsumsi', 'mobil', 'listrik']	['senang', 'program', 'pemerintah', 'hilirisasi', 'semoga', 'pemerintah', 'membangun', 'pembangkit', 'listrik', 'mencukupi', 'konsumsi', 'mobil', 'listrik']
['orang', 'indonesia', 'ada', 'tuh', 'yang', 'bikin', 'mobil', 'listrik', 'cuman', 'nggak', 'di', 'hargain', 'giliran', 'mobil', 'listrik', 'luar', 'negeri', 'datang', 'ke', 'indonesia', 'di', 'hargai', 'nya', 'minta', 'apuun']	['senang', 'program', 'pemerintah', 'hilirisasi', 'semoga', 'pemerintah', 'membangun', 'pembangkit', 'listrik', 'mencukupi', 'konsumsi', 'mobil', 'listrik']

e) *Stemming*

Stemming adalah tahap mengubah sebuah kata ke dalam bentuk kata dasarnya dengan menghapus kata imbuhan di depan maupun imbuhan di belakang

kata. Dalam proses *stemming* akan digunakan modul sastrawi yang merupakan modul *stemmer* bahasa Indonesia yang dibuat dengan bahasa pemrograman *python* yang memiliki fungsi untuk melakukan stemming pada kata-kata dalam bahasa Indonesia. Contoh penerapan pada tahap *stemming* dapat dilihat pada Tabel 3.5.

Tabel 3.5. Contoh Proses Penerapan Stemming

Before	After
['senang', 'program', 'pemerintah', 'hilirisasi', 'semoga', 'pemerintah', 'membangun', 'pembangkit', 'listrik', 'mencukupi', 'konsumsi', 'mobil', 'listrik']	senang program perintah hilir moga perintah bangun bangkit listrik cukup konsumsi mobil listrik
['senang', 'program', 'pemerintah', 'hilirisasi', 'semoga', 'pemerintah', 'membangun', 'pembangkit', 'listrik', 'mencukupi', 'konsumsi', 'mobil', 'listrik']	senang program perintah hilir moga perintah bangun bangkit listrik cukup konsumsi mobil listrik

3.5.2 Pembagian Dataset

Pada proses ini, dataset yang telah melalui tahap *preprocessing* akan dibagi menjadi data latih dan data uji. Data latih digunakan untuk melakukan ekstraksi fitur dengan TF-IDF serta melatih model klasifikasi Naïve Bayes sehingga model dapat belajar melalui data dan dapat mengenali pola dari data latih tersebut. Sedangkan data uji akan digunakan untuk mengukur kinerja model agar dapat melihat persentase algoritma klasifikasi yang berhasil mengklasifikasikan dengan benar. Jumlah data yang digunakan sebanyak 3509 data yang akan dibagi menjadi data latih dan data uji dengan perbandingan yang dapat dilihat pada tabel 3.6 berikut.

Tabel 3.6 Datail Perbandingan Pembagian data latih dan data uji

Perbandingan	Data Latih	Data Uji	Total Seluruh Data
90 : 10	3158	351	3509
80 : 20	2807	702	3509
70 : 30	2456	1053	3509
60 : 40	2105	1404	3509

Berdasarkan beberapa perbandingan yang digunakan dalam tabel 3.6 diatas, akan dicari perbandingan yang memiliki performa terbaik dalam melatih model klasifikasi dengan algoritma Naïve Bayes Classifier dengan ekstraksi fitur dengan *Tf-Idf*.

3.5.3 Ekstraksi Fitur

Setelah semua data sudah melewati tahap *labelling*, *preprocessing* serta pembagian data latih dan data uji selanjutnya adalah melakukan ekstraksi fitur untuk mempermudah proses klasifikasi. Pada tahap ekstraksi fitur dilakukan perubahan fitur kata atau teks menjadi sebuah representasi angka (numerik) dengan melihat kemunculan fitur kata atau teks terlebih dahulu dengan metode *CountVectorizer*, lalu dilanjutkan dengan mencari pembobotan kata menggunakan metode *Tf-Idf*. Metode *Tf-Idf* akan melakukan perhitungan nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap fitur kata di setiap dokumen dalam *dataset*.

3.5.4 Klasifikasi *Naïve Bayes*

Pada tahap ini *Naïve Bayes* akan melakukan prediksi kelas positif atau negatif dengan menguji data serta mempelajari pengetahuan yang terkandung dalam data latih. Pada data latih terdapat kelas sentimen positif dan negatif, *Naïve Bayes* akan mempelajari karakteristik kata-kata yang terdapat pada masing-masing kelas. Akan terdapat dua proses yang akan dilakukan sebagai berikut:

a) Melatih Model *Naïve Bayes*

Pada tahap melatih model *Naïve Bayes* dataset akan dibagi menjadi dua subset data latih dan data uji, subset data latih akan digunakan untuk melatih model *Naïve Bayes* yang dimana *Naïve Bayes* akan mempelajari pola dari data pelatihan untuk mengklasifikasikan sentimen pada suatu data teks komentar apakah termasuk sentiment positif atau negatif.

b) Menguji Model *Naïve Bayes*

Pada tahap menguji model dari *Naïve Bayes* akan dilakukan klasifikasi pada data yang belum pernah dilihat oleh model. Data uji akan digunakan oleh model *Naïve Bayes* yang telah dilatih pada subset pelatihan untuk memprediksi kelas dari data pengujian. *Naïve Bayes* akan menggunakan pola yang telah dipelajari selama pelatihan untuk dapat mengklasifikasikan data menjadi kelas yang sesuai pada tahap uji yaitu melakukan klasifikasi sentiment pada data komentar tentang kendaraan listrik apakah positif atau negatif.

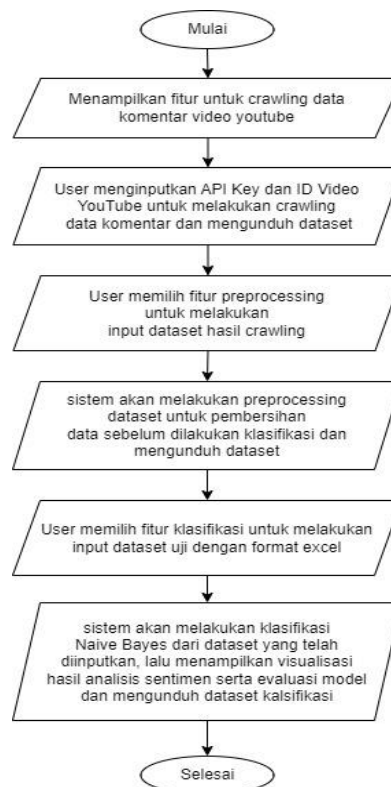
3.2.7 Evaluasi Model

Evaluasi model dilakukan untuk mengetahui kinerja model. Evaluasi model dilakukan dengan cara melihat tingkat akurasi metode melalui *confusion matrix* dan tabel akurasi serta presisi untuk tiap model. Setelah data *test* diujikan terhadap data *training*, maka akan menghasilkan daftar kelas-kelas dari data *test*, sebut saja prediksi kelas. Kemudian prediksi kelas dibandingkan dengan kelas yang sebenarnya dari data *test* yang disembunyikan sebelumnya. Sehingga dapat dilihat dan dihitung nilai *accuracy*, *precision*, *recall*, dan *f1-score*.

3.6 Perancangan Aplikasi Sentimen Analisis Berbasis Web

Dalam merancang sistem aplikasi yang dapat digunakan untuk analisis opini berdasarkan teks komentar dengan mengimplementasikan metode *Naïve Bayes Classifier* dibuatlah sebuah diagram alir untuk membantu dalam proses memahami alur dari proses sistem berlangsung. Diagram alir tersebut menggambarkan tahapan-tahapan dalam menjalankan sistem. Berikut akan dibahas alur kerja dari sebuah sistem dan interaksi sistem dengan user yang akan digambarkan melalui diagram UML dan *Design User Interface* (UI).

a) Diagram Alir



Gambar 3.4. Diagram Alir Sistem

Gambar 3.4. diatas merupakan diagram alir dari sistem klasifikasi analisis opini publik terhadap kendaraan listrik dengan algoritma *Naïve Bayes*. Diagram alir diatas menggambarkan alur kerja dari sistem yang dimana dimulai dari menampilkan halaman fitur *crawling* komenetar *youtube* lalu *user* dapat memasukkan *API Key* serta ID Video yang akan diambil data komentarnya, selanjutnya *user* dapat mengunduh hasil *crawling* menjadi sebuah *dataset* dengan format *excel*. Kemudian *user* dapat melanjutkan ke halaman fitur *preprocessing* untuk melakukan pembersihan dataset sebelum dilakukan pengujian data atau klasifikasi *Naïve Bayes* serta *user* dapat mengunduh hasil *preprocessing* data dalam format *excel*. Selanjutnya, setelah mengunduh dataset yang telah melalui *preprocessing*, *user* dapat melanjutkan untuk menginput dataset *preprocessing* pada halaman fitur klasifikasi untuk dilakukan pengklasifikasian dengan *Naïve Bayes* serta menampilkan proses ekstraksi fitur *tf-idf*, visualisasi hasil analisis sentimen dan evaluasi model.

b) *Design User Interface*

Design User Interface website ini dirancang untuk membantu dan memfasilitasi pembuatan aplikasi yang bertujuan untuk melakukan klasifikasi analisis opini publik terhadap kendaraan listrik dengan algoritma *Naïve Bayes Classifier* untuk menciptakan pengalaman pengguna yang optimal. Untuk membuat desain UI digunakan aplikasi *pencil*, sementara untuk mengembangkan aplikasi *web* digunakan *framework Streamlit*.

Sentiment analyze App

Pilihan Menu:

Crawling

Crawling Youtube Comments

Input API Key Youtube:

Input ID Video Youtube:

Crawl Comments

Gambar 3.5. *Design UI Halaman Crawling Youtube Comments*

Gambar 3.5. diatas merupakan *design* UI untuk halaman awal dari sistem aplikasi analisis opini publik terhadap kendaraan listrik dengan algoritma *Naïve Bayes Classifier* yang menampilkan fitur *crawling* data komentar *youtube*. Pada halaman ini, *user* dapat melakukan proses pengambilan data komentar dari sebuah video *youtube* dengan menginputkan *API Key youtube* dan ID Video *youtube* terkait yang ingin diambil komentarnya. Selanjutnya, *user* dapat melakukan *crawling* data komentar pada video *youtube* dan sistem akan menampilkan data komentar yang ingin diambil berdasarkan *API Key youtube* dan ID Video *youtube* yang diinputkan berupa *dataset* dengan format *excel* yang dapat di download.

Sentiment analyze App
Pilihan Menu:
Preprocessing ▼

Preprocessing Data

Upload File Dataset Excel (.xlsx)

[Browse File](#)

Preprocessing Result:

	Comment	Label
1	subsidi kuasa rakyat beli motor listrik rakyat minat beli motor mobil listrik	Negatif
2	indonesia bleh maju ide laku perintah jalan tambang hilir produksi baterai kendara	Negatif
3	tuju bangun kendara listrik	Positif
4	subsidi kendara listrik pacu manufaktur ekosistem kendara listrik indonesia industri cepat kembang salah satu	Positif

Download Preprocessing Result:
[Download Excel](#)

Gambar 3.6. Design UI Halaman Preprocessing Data

Gambar 3.6. diatas merupakan design UI pada halaman menu *Preprocessing* yang dimana, *user* nantinya dapat melakukan pembersihan data teks komentar yang sudah dikumpulkan melalui tahap *crawling* data sebelumnya. Pada halaman ini user dapat menginputkan dataset teks komentar dengan format *excel*, lalu sistem akan melakukan proses *cleansing*, normalisasi, *tokenize*, *stopwords*, *stemming* dan *user* dapat mendownload *file dataset excel* hasil *preprocessing* yang siap untuk dilakukan klasifikasi untuk analisis sentimen.

Sentiment analyze App

Pilihan Menu:

Classification

Classification Naive Bayes

Upload File Dataset Excel (.xlsx) untuk melakukan analisis sentimen:

Dataset.xlsx

Browse File

Ekstraksi Fitur TF-IDF:

	Subsidi	Motor	Mobil	Listrik	Produksi	Ekosistem	Industri
1	0.2967	0.2109	0.2110	0.2091	0	0	0
2	0	0	0	0	0.2354	0	0
3	0	0	0	0	0	0.2567	0
4	0	0	0	0.2100	0	0.2365	0.2567

Hasil Analisis Sentiment

	Comment	Label	Classification
1	subsidi kuasa rakyat beli motor listrik rakyat minat beli motor mobil listrik	Negatif	Negatif
2	Indonesia boleh maju ide laku perintah jalan tambang hilir produksi baterai kendaraan	Negatif	Positif
3	tuju bangun kendaraan listrik ekosistem	Positif	Positif
4	subsidi kendaraan listrik pacu manufaktur ekosistem kendaraan listrik Indonesia industri cepat kembang salah satu	Positif	Positif

Gambar 3.7. Design UI Halaman *Classification Naive Bayes*

Gambar 3.7. merupakan design UI pada halaman menu *Classification* yang dimana, *user* nantinya dapat menginputkan dataset teks komentar tentang opini publik terhadap kendaraan listrik dengan format *excel*, lalu sistem akan melakukan klasifikasi terhadap *dataset* yang diinputkan dengan menampilkan proses ekstraksi fitur dengan TF-IDF dan menampilkan hasil klasifikasi sentimen dengan algoritma *Naive Bayes* beserta persentase klasifikasi sentimen positif dan negatif berupa grafik *histogram*. Selain itu, sistem juga akan menampilkan evaluasi model *Naive Bayes* berupa akurasi, *classification report*, dan *confusion matrix*.

3.7 Pembahasan dan Kesimpulan

Setelah membuat model klasifikasi analisis opini publik terhadap kendaraan listrik di Indonesia dengan algoritma *Naive Bayes Classifier (NBC)*, pada tahap ini akan dilakukan analisis dari hasil klasifikasi yang telah dilakukan yaitu dari proses pengumpulan data (*crawling*) komentar melalui *youtube*, *preprocessing*, ekstraksi fitur dengan *tf-idf* atau pembobotan kata, pemodelan klasifikasi dengan *Naive Bayes* serta analisa hasil evaluasi model klasifikasi *naive bayes* seperti melihat *accuration*, *precision*, *recall*, dan *f1-score* yang dihasilkan. Selain itu akan dilakukan juga analisis terhadap aplikasi analisis opini publik berdasarkan teks komentar berbasis *web* yang dibuat dengan *framework streamlit* apakah sudah bekerja dengan baik. Kemudian

30

setelah melakukan analisis, akan ditarik kesimpulan dari analisis yang dilakukan serta saran untuk penelitian selanjutnya.

3.8 Alat Penelitian

Beberapa alat dan bahan yang akan digunakan dalam penelitian ini adalah sebagai berikut:

a) *Hardware*

- Laptop dengan spesifikasi *processor intel core i3, RAM 8 GB*

b) *Software*

- Sistem operasi *windows 10*.
- *Microsoft Excel* untuk menampilkan *dataset* mentah.
- *Microsoft Word* untuk membuat laporan penelitian.
- Aplikasi *Draw.io* untuk membuat diagram
- *Visual Studio Code* untuk kode editor dalam melakukan perancangan model dan perancangan sistem menggunakan *streamlit*.
- *Python* sebagai bahasa pemrograman dalam *machine learning*.

BAB IV

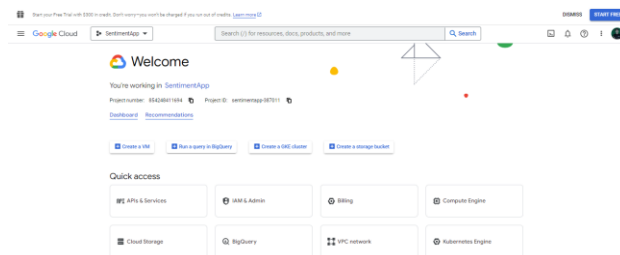
HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

Pengumpulan data dalam penelitian ini didapatkan melalui teknik *crawling* data komentar pada *twitter* dan video *youtube* yang membahas mengenai kendaraan listrik dengan rentang waktu data yang diambil mulai dari 15 April 2021 – 25 Mei 2023. Kemudian *dataset* hasil *crawling* disimpan dalam *dataframe* dengan format *excel* yang dapat diakses pada *link github* (<https://github.com/ryasakbar060/Dataset-Komentar-Kendaraan-Listrik>). Adapun beberapa proses atau langkah-langkah yang dilakukan dalam tahap *crawling* data *youtube* adalah sebagai berikut:

4.1.1 Create Credentials YouTube Data API Key

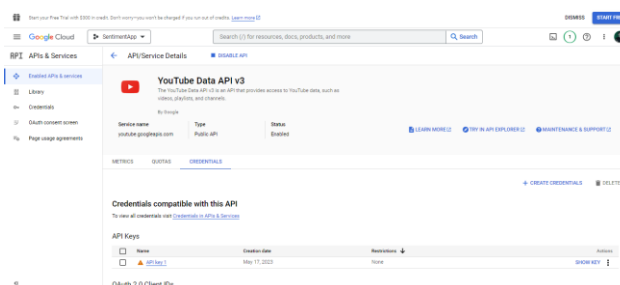
1) Create project pada Console Google Cloud



Gambar 4.1. Tampilan halaman *project* pada *Console Google Cloud*

Gambar 4.1. merupakan tampilan halaman setelah membuat *project* baru pada *Console Google Cloud*. Setelah *project* dibuat maka dapat dilakukan aktivasi terhadap *YouTube Data API* dengan memilih fitur *API & Services*, lalu mencari *YouTube Data API v3* pada kolom pencarian, kemudian mengaktifkan (*Enable*) pada hasil pencarian untuk aktivasi *YouTube Data API v3*.

2) Create Credential API Key



Gambar 4.2. Tampilan halaman *API & Services*

Gambar 4.2. merupakan tampilan halaman pada fitur *API & Services* setelah diaktifkan, yang dimana pada halaman tersebut akan dilakukan proses dalam tahap pembuatan *API Key* dengan memilih *Create Credentials*, lalu memilih opsi *API Key* untuk mendapatkan *API Key* yang akan digunakan untuk mengakses *YouTube Data API*.

4.1.2 Implementasi Crawling Data Youtube Comments

```
from googleapiclient.discovery import build

def video_comments(video_ids):
    # list kosong untuk menyimpan komentar dan balasan
    replies = []

    # membuat objek resource YouTube
    youtube = build('youtube', 'v3', developerKey=api_key)

    for video_id in video_ids:
        # Mengambil hasil video youtube
        video_response =
        youtube.commentThreads().list(part='snippet,replies',
        videoId=video_id).execute()

        while video_response:
            for item in video_response['items']:
                # Mengambil komentar
                published =
                item['snippet']['topLevelComment']['snippet']['publishedAt']
                user =
                item['snippet']['topLevelComment']['snippet']['authorDisplayName']
                comment =
                item['snippet']['topLevelComment']['snippet']['textDisplay']
                replies.append([published, user, comment])

                # menghitung jumlah balasan pada komentar
                replycount =
                item['snippet']['totalReplyCount']

                # jika ada balasan
                if replycount > 0:
                    for reply in
                    item['replies']['comments']:
                        # Mengambil balasan
                        published =
                        reply['snippet']['publishedAt']
                        user =
                        reply['snippet']['authorDisplayName']
                        repl =
                        reply['snippet']['textDisplay']
                        replies.append([published, user,
                        repl])

            # Ulangi lagi jika ada halaman berikutnya
            if 'nextPageToken' in video_response:
```



```

        video_response =
youtube.commentThreads().list(
    part='snippet,replies',
    pageToken=video_response['nextPageToken']
],
    videoId=video_id
).execute()
    else:
        break
# endwhile
return replies

# Input api key yang telah dibuat pada Console Google Cloud
api_key = 'AIzaSyCgJMvPcYZfRvyMnG6ZwumpyqYQ55yHOEQ'

# Input ID Video Youtube
video_ids = ["L4egbc3UvMQ"]

# Memanggil Fungsi
comments = video_comments(video_ids)

```

Kode di atas adalah implementasi untuk melakukan *crawling* komentar dari video *Youtube* menggunakan *API YouTube Data v3* dari *Console Google Cloud*. Pertama, fungsi `video_comments` digunakan untuk mengambil komentar beserta balasan komentar dari video *Youtube* berdasarkan daftar ID video yang diberikan sebagai *input*. Dalam fungsi ini, objek `youtube` dibangun menggunakan `build` dari `googleapiclient.discovery`, yang akan digunakan untuk melakukan permintaan atau akses ke *API YouTube*.

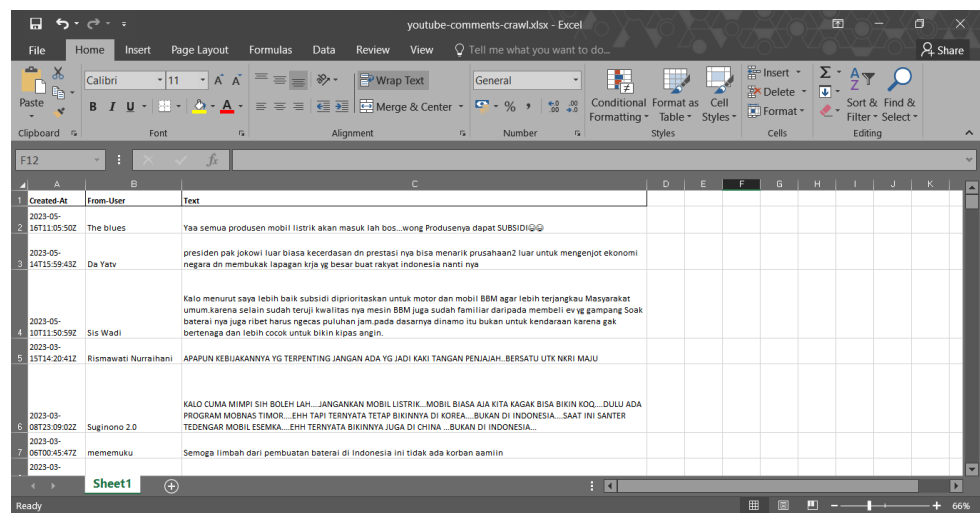
Selanjutnya, untuk setiap ID video, fungsi ini akan menggunakan `youtube.commentThreads().list` untuk mengambil komentar dan balasan yang terkait dengan video tersebut. Dalam hasil *respons*, data komentar diekstrak dari *JSON respons* dan disimpan dalam daftar `replies`. Jumlah balasan pada setiap komentar dihitung dengan `replycount`, dan jika ada balasan, balasan tersebut juga diekstrak dan ditambahkan ke dalam daftar `replies`.

Kemudian, fungsi ini melakukan halaman demi halaman dengan menggunakan `nextPageToken` untuk mendapatkan semua komentar dan balasannya jika ada lebih dari satu halaman hasil. Proses ini dilakukan menggunakan `loop while`, sehingga fungsi akan terus berulang hingga semua komentar dan balasannya dari semua halaman telah diambil.

```
# Menampilkan Hasil Crawling dalam DataFrame
df = pd.DataFrame(comments, columns=['Created-At', 'From-User', 'Text'])
df['Created-At'] = pd.to_datetime(df['Created-At']).dt.tz_localize(None)

# Mendownload Hasil Crawling dalam format excel
df.to_excel('youtube-comments-crawl.xlsx', index=False)
```

Selanjutnya, setelah selesai mengambil komentar dan balasan dari semua video yang diberikan, hasil *crawling* data komentar video *youtube* disimpan dalam *DataFrame* seperti implementasi pada kode diatas. *DataFrame* ini kemudian diubah format waktu di kolom '*Created-At*' menjadi objek *datetime*, dan hasilnya disimpan dalam file *Excel* dengan nama *youtube-comments-crawl.xlsx* seperti pada gambar 4.3. berikut.



Created-At	From-User	Text
2023-05-16T11:05:50Z	The blues	Yaa semua produsen mobil listrik akan masuk lah bos...wong Produsennya dapat SUBSIDIG
2023-05-14T15:59:43Z	Da Yatv	presiden pak Jokowi luar biasa kecerdasan dn prestasi nya bisa menarik perusahaan2 luar untuk mengentot ekonomi negara dn membuka lapangan kerja yg besar buat rakyat Indonesia nanti nya
2023-05-10T11:50:59Z	Sis Wadi	Kalo menurut saya lebih baik subsidi diprioritaskan untuk motor dan mobil BEM agar lebih terjangkau Masyarakat umum.karena selain sudah teruji kwalitas nya mesin BEM juga sudah familiar daripada membeli ev yg gampang Soak baterai nya juga ribet harus ngecas puluhan jam.pada dasarnya dinamo itu bukan untuk kendaraan karena gak bertenaga dan lebih cocok untuk bikin kipas angin.
2023-05-13T14:20:41Z	Rismawati Nurrahani	APAPUN KEBUAKANNYA YG TERPENTING JANGAN ADA YG JADI KAKI TANGAN PENAJAH..BERSATU UTK NKRI MAJU
2023-05-08T23:09:02Z	Suginono 2.0	KALO CUMA MIMPI SIH BOLEH LAH...JANGANKAN MOBIL LISTRIK...MOBIL BIASA AJA KITA KAGAK BISA BIKIN KOQ...DULU ADA PROGRAM MOBILNAS TIMOR...EHH TAPI TERNYATA TETAP BIKINYA DI KOREA...BUKAN DI INDONESIA...SAAT INI SANTER TEDENGAR MOBIL ESEMKA...EHH TERNYATA BIKINYA JUGA DI CHINA...BUKAN DI INDONESIA...
2023-05-06T00:45:47Z	mememuku	Semoga limbah dari pembuatan baterai di Indonesia ini tidak ada korban aamlin

Gambar 4.3. File excel dataset hasil *crawling* data komentar *youtube*

4.2 Labelling Dataset

Pelabelan dataset komentar mengenai opini publik terhadap kendaraan listrik dilakukan dengan teknik *labelling* manual dengan bantuan dari pakar bahasa indonesia yang dimana pada penelitian ini ialah Ibu Dra. Syamsinas Jafar, M. Hum yang merupakan salah satu Dosen Jurusan Bahasa Indonesia FKIP Universitas Mataram. Alasan penulis memilih *labelling* secara manual adalah agar label data yang diberikan lebih akurat dan dapat mengidentifikasi nuansa dan sentimen yang kompleks dalam teks serta dapat memahami konteks dan emosi dengan lebih baik, walaupun metode ini membutuhkan waktu dan usaha yang cukup ekstra dalam memberi label secara manual.

Proses dalam penentuan kelas sentimen dengan metode *manual classificaton (labelling manual)* dalam penelitian ini dimulai dari penentuan sentimen secara manual oleh penulis terlebih dahulu dengan menentukan tiga kelas sentimen yaitu Positif, Negatif dan Netral. Setelah dilakukan pelabelan data secara manual dari sisi penulis, data hasil *labelling* akan diteruskan dan di koreksi kembali oleh pakar. Berikut detail jumlah kelas sentimen hasil *labelling dataset* secara manual dapat dilihat pada tabel 4.1.

Tabel 4.1. Hasil *labelling* awal *dataset* secara manual

Positif	Negatif	Netral	Jumlah
1972	1620	678	4270
46,81%	37,94%	15,88%	100%

Berdasarkan tabel 4.1. dari hasil *labelling* data diatas yang terdiri dari kelas Positif, Negatif dan Netral. Pada penelitian ini kelas sentimen data yang akan digunakan untuk melatih model ialah hanya data berlabel Positif dan Negatif saja dengan alasan untuk menghindari ketidakseimbangan dalam jumlah sampel antara kelas Positif, Negatif dan Netral yang dapat menyebabkan model cenderung fokus pada kelas mayoritas (Positif dan Negatif) sehingga mengabaikan kelas minoritas (Netral). Maka dari itu, berikut adalah detail data setelah kelas Netral pada dataset dihilangkan dapat dilihat pada tabel 4.2.

Tabel 4.2. Jumlah data *labelling* Positif dan Negatif secara manual

Positif	Negatif	Jumlah
1972	1620	3592
54,93%	45,07%	100%

Selanjutnya, data pada tabel 4.2 diatas akan dibersihkan pada tahap *preprocessing* seperti *cleansing*, *normalisasi*, *tokenizing*, *stopwords* dan *stemming* sehingga menghasilkan data bersih sebesar 3509 yang akan digunakan untuk membangun model *mechine learning* dalam analisis sentimen menggunakan *Naïve Bayes Classifier* dengan ekstraksi fitur *Tf-Idf*. Berikut detail *dataset* yang akan digunakan dapat dilihat pada tabel 4.3.

Tabel 4.3. Jumlah data *labelling* Positif dan Negatif setelah melalui *preprocessing*

Positif	Negatif	Total Seluruh Data
1904	1605	3509

4.3 Perancangan Model

4.3.1 *Preprocessing*

Pada tahap ini akan dilakukan *preprocessing dataset* atau mempersiapkan data teks komentar yang telah didapatkan dari hasil *crawling* dan *labelling* sebelum dilakukan proses pengklasifikasian dengan algoritma *naïve bayes*. Tahap ini dilakukan untuk melakukan pembersihan data sehingga siap untuk di proses pada tahap selanjutnya. *Preprocessing* yang dilakukan terdiri menjadi beberapa tahapan diantaranya ialah *cleansing*, normalisasi, *tokenizing*, *stopwords* dan *stemming*. Berikut adalah proses dari beberapa tahapan tersebut.

a) *Cleansing*

Pada tahap *cleansing* bertujuan untuk membersihkan data teks komentar dari hal yang tidak diperlukan yang membuat data menjadi tidak efektif seperti *hashtag*, *mention username*, *URL*, tanda baca, *double space* (spasi berlebih), *emoticon*, bilangan angka, simbol dan *case folding* atau mengubah teks menjadi huruf kecil (*lower-case*). Berikut adalah implementasi kode python yang digunakan dalam proses *cleansing*.

```
import re
import emoji

def cleansing(Text):
    Text = re.sub(r'RT', '', Text) # remove RT
    Text = Text.replace("<br>", " ") # mengubah <br> menjadi
    spasi
    Text = ' '.join(re.sub("([@#][A-Za-z0-9]+)|(\w+:\/\/\S+)", "", Text).split()) # Menghapus mention,
    hashtag, dll
    Text = Text.lower() # Mengubah menjadi huruf kecil
    Text = re.sub(r"^[a-zA-Z0-9]", " ", Text)
    Text = emoji.demojize(Text) # Menghilangkan emoji
    Text = re.sub(r':[a-zA-Z_]+:', '', Text)
    Text = re.sub(r'^\w\s', '', Text) # remove tanda baca
    Text = re.sub(r'\d+', '', Text) # remove angka
    Text = Text.replace("http://", " ").replace("https://", "
") # remove URL
    Text = Text.replace('\t', " ").replace('\n', "
").replace('\u', " ").replace('\\', " ") # Menghapus karakter
    escape seperti '\t', '\n', '\u'
    return Text
```

Kode diatas merupakan sebuah fungsi *cleansing* yang digunakan untuk membersihkan data teks komentar dari karakter-karakter yang tidak diinginkan

dan mengubah teks menjadi bentuk yang lebih bersih dan seragam sehingga nantinya siap digunakan sebagai data latih untuk model analisis sentimen. Dalam fungsi `cleansing` diatas menggunakan *library regex (regular expression)* yang digunakan untuk menghapus karakter-karakter tertentu dalam data teks komentar dan *library emoji* untuk menghilangkan emoji dari data teks komentar.

```
df['Cleansing'] = df['Text'].apply(cleansing)
df.head()
```

Kode diatas merupakan proses penerapan fungsi `cleansing` pada setiap nilai dalam kolom *'Text'* dari dataset. Setelah membersihkan teks dengan fungsi `cleansing`, hasilnya akan disimpan dalam kolom baru bernama *'Cleansing'*. Fungsi `cleansing` akan diterapkan pada setiap nilai teks dalam kolom *'Text'* menggunakan metode `apply`. Fungsi `apply` digunakan untuk menerapkan fungsi tertentu pada setiap elemen dalam kolom *'Text'*. Hasil atau *output* dari penerapan fungsi `cleansing` dapat dilihat pada tabel 4.4. berikut.

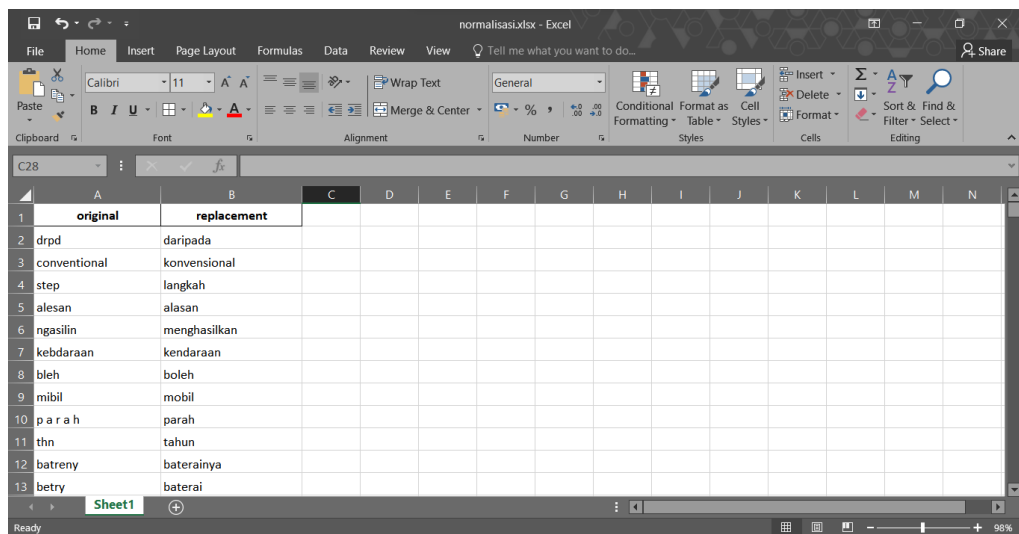
Tabel 4.4. Hasil *preprocessing* pada tahap *cleansing*

Text	Cleansing
Yaa semua produsen mobil listrik akan masuk lah bos...wong Produsenyanya dapat SUBSIDI😊😊😊	yaa semua produsen mobil listrik akan masuk lah bos wong produsenyanya dapat subsidi
presiden pak jokowi luar biasa kecerdasan dn prestasi nya bisa menarik prusahaan2 luar untuk mengenjot ekonomi negara dn membukak lapangan krja yg besar buat rakyat indonesia nanti nya	presiden pak jokowi luar biasa kecerdasan dn prestasi nya bisa menarik prusahaan luar untuk mengenjot ekonomi negara dn membukak lapangan krja yg besar buat rakyat indonesia nanti nya
Mantap pak jokowi...saat nya indonesia penguasa industri modern	mantap pak jokowi saat nya indonesia penguasa industri modern
Rakyat harus bersatu , kita sudah lama dijajah , saat kita mau maju kok ada yg menghalangi kemajuan negara kita , kita harus siap berdarah , jgn biarkan organisasi preman dunia memaksa maksa negara kita.	rakyat harus bersatu kita sudah lama dijajah saat kita mau maju kok ada yg menghalangi kemajuan negara kita kita harus siap berdarah jgn biarkan organisasi preman dunia memaksa maksa negara kita
Maju terus, jangan mundur, kita akan maju , inshaAllah	maju terus jangan mundur kita akan maju inshaallah

b) Normalisasi

Pada tahap normalisasi ini bertujuan untuk mengubah kata-kata singkatan atau kata yang tidak baku (*slang words*) dalam data teks komentar menjadi bahasa yang baku sesuai Kamus Besar Bahasa Indonesia (KBBI). Tahap ini dilakukan

dengan cara mengubah kata *slang* yang terdeteksi dengan menggunakan acuan dataset normalisasi kamus bahasa slang dengan format *excel*. Normalisasi *slang* ini berguna untuk memastikan konsistensi dan keakuratan dalam analisis teks. Dengan normalisasi ini, teks akan lebih mudah diinterpretasikan oleh model analisis sentimen atau algoritma pengklasifikasian, sehingga meningkatkan kualitas dan akurasi analisis teks secara keseluruhan. Berikut dapat dilihat pada gambar 4.4. file *excel* untuk kamus normalisasi bahasa *slang* yang dibuat pada penelitian ini.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	original	replacement												
2	drpd	daripada												
3	conventional	konvensional												
4	step	langkah												
5	alesan	alasan												
6	ngasilin	menghasilkan												
7	kebadaan	kendaraan												
8	bleh	boleh												
9	mibil	mobil												
10	p a r a h	parah												
11	thn	tahun												
12	batreny	baterainya												
13	betry	baterai												

Gambar 4.4. File *excel* kamus normalisasi kata *slang*

Berikut adalah implementasi proses normalisasi dalam *code python* pada *preprocessing dataset*.

```
def slang_normalization(text):
    df_slang = pd.read_excel("normalisasi.xlsx")
    slang_dict = dict(zip(df_slang['original'],
    df_slang['replacement']))
    text = ' '.join([slang_dict[word] if word in slang_dict
    else word for word in text.split()])
    return text

df['Normalisasi'] = df['Cleansing'].apply(slang_normalization)
df.head()
```

Kode diatas merupakan fungsi yang dibuat untuk melakukan normalisasi kata slang dalam teks menggunakan kamus normalisasi yang disimpan dalam file Excel ('normalisasi.xlsx'). Pertama, fungsi membaca *file Excel* dan menyimpannya dalam *DataFrame* *df_slang*. Selanjutnya, kamus normalisasi dibentuk dari kolom 'original' dan 'replacement'. Dalam fungsi ini, teks yang akan dinormalisasi dipecah menjadi kata-kata dan setiap kata diperiksa apakah ada

dalam kamus normalisasi. Jika kata ditemukan dalam kamus, kata tersebut digantikan dengan kata yang sesuai berdasarkan kamus normalisasi. Jika tidak ditemukan, kata tersebut dipertahankan tanpa perubahan. Setelah normalisasi selesai, kata-kata dalam teks digabungkan kembali. Hasil normalisasi slang kemudian dikembalikan sebagai *output* dari fungsi dalam bentuk DataFrame yang akan disimpan dalam kolom baru bernama '*Normalisasi*'. Lalu, fungsi normalisasi akan diterapkan pada setiap nilai teks dalam kolom '*Cleansing*' menggunakan metode `apply`. Hasil atau *output* dari penerapan fungsi normalisasi dapat dilihat pada tabel 4.5. berikut.

Tabel 4.5. Hasil *preprocessing* pada tahap normalisasi

Cleansing	Normalisasi
yaa semua produsen mobil listrik akan masuk lah bos wong produsennya dapat subsidi	iya semua produsen mobil listrik akan masuk lah bos wong produsennya dapat subsidi
presiden pak jokowi luar biasa kecerdasan dn prestasi nya bisa menarik prusahaan luar untuk mengenjot ekonomi negara dn membukak lapangan krja yg besar buat rakyat indonesia nanti nya	presiden pak jokowi luar biasa kecerdasan dan prestasi nya bisa menarik perusahaan luar untuk menggenjot ekonomi negara dan membuka lapangan kerja yang besar buat rakyat indonesia nanti nya
mantap pak jokowi saat nya indonesia penguasa industri modern	mantap pak jokowi saat nya indonesia penguasa industri modern
rakyat harus bersatu kita sudah lama dijajah saat kita mau maju kok ada yg menghalangi kemajuan negara kita kita harus siap berdarah jgn biarkan organisasi preman dunia memaksa maksa negara kita	rakyat harus bersatu kita sudah lama dijajah saat kita mau maju kok ada yang menghalangi kemajuan negara kita kita harus siap berdarah jangan biarkan organisasi preman dunia memaksa maksa negara kita
maju terus jangan mundur kita akan maju inshaallah	maju terus jangan mundur kita akan maju inshaallah

c) *Tokenizing*

Pada tahap *tokenizing* ini bertujuan untuk memecah teks atau kalimat menjadi unit-unit yang lebih kecil, yang disebut *token*. . Proses *tokenizing* adalah langkah awal yang penting dalam analisis sentimen karena memungkinkan untuk mengubah teks menjadi bentuk yang lebih terstruktur, yang kemudian dapat diolah lebih lanjut untuk memahami sentimen atau makna yang terkandung di dalamnya. Berikut adalah implementasi *tokenizing* dalam *code python* dapat dilihat sebagai berikut.

```

from nltk.tokenize import word_tokenize

def tokenization(Text):
    tokens = word_tokenize(Text)
    return tokens

df['Tokenize'] = df['Normalisasi'].apply(tokenization)
df.head()

```

Kode diatas merupakan fungsi tokenisasi teks yang menggunakan modul *Natural Language Toolkit* (NLTK) yang merupakan *library python open-source* pada bahasa pemrograman *Python* yang berfokus pada pengolahan bahasa alami dengan mengimpor modul `word_tokenize`. Fungsi `tokenization` menerima teks sebagai input, dan kemudian menggunakan fungsi `word_tokenize` untuk memecah teks menjadi token-token yang lebih kecil, yaitu kata-kata. Kemudian, fungsi `tokenization` akan diaplikasikan pada kolom 'Normalisasi' dari DataFrame menggunakan metode `apply`, dan hasil tokenisasi untuk setiap teks akan disimpan dalam kolom baru 'Tokenize' pada DataFrame. Selanjutnya, Hasil akhirnya adalah DataFrame yang berisi kolom 'Normalisasi' yang berisi teks yang telah dinormalisasi dan kolom 'Tokenize' yang berisi daftar token atau kata-kata yang dihasilkan dari setiap teks. Berikut adalah hasil atau *output* dari penerapan *tokenizing* yang dapat dilihat pada tabel 4.6. berikut.

Tabel 4.6. Hasil *preprocessing* pada tahap *tokenizing*

Normalisasi	Tokenize
iya semua produsen mobil listrik akan masuk lah bos wong produsennya dapat subsidi	['iya', 'semua', 'produsen', 'mobil', 'listrik', 'akan', 'masuk', 'lah', 'bos', 'wong', 'produsennya', 'dapat', 'subsidi']
presiden pak jokowi luar biasa kecerdasan dan prestasi nya bisa menarik perusahaan luar untuk menggenjot ekonomi negara dan membuka lapangan kerja yang besar buat rakyat indonesia nanti nya	['presiden', 'pak', 'jokowi', 'luar', 'biasa', 'kecerdasan', 'dan', 'prestasi', 'nya', 'bisa', 'menarik', 'perusahaan', 'luar', 'untuk', 'menggenjot', 'ekonomi', 'negara', 'dan', 'membuka', 'lapangan', 'kerja', 'yang', 'besar', 'buat', 'rakyat', 'indonesia', 'nanti', 'nya']
mantap pak jokowi saat nya indonesia penguasa industri modern	['mantap', 'pak', 'jokowi', 'saat', 'nya', 'indonesia', 'penguasa', 'industri', 'modern']
rakyat harus bersatu kita sudah lama dijajah saat kita mau maju kok ada yang menghalangi kemajuan negara kita kita harus siap berdarah jangan biarkan organisasi preman dunia memaksa maksa negara kita	['rakyat', 'harus', 'bersatu', 'kita', 'sudah', 'lama', 'dijajah', 'saat', 'kita', 'mau', 'maju', 'kok', 'ada', 'yang', 'menghalangi', 'kemajuan', 'negara', 'kita', 'kita', 'harus', 'siap', 'berdarah', 'jangan', 'biarkan', 'organisasi', 'preman', 'dunia', 'memaksa', 'maksa', 'negara', 'kita']

maju terus jangan mundur kita akan maju inshaallah	['maju', 'terus', 'jangan', 'mundur', 'kita', 'akan', 'maju', 'inshaallah']
---	--

d) *Stopwords*

Pada tahap *stopwords* ini akan dilakukan proses menghilangkan kata-kata yang tidak memiliki makna yang khusus dalam analisis teks dengan tujuan agar membuat data menjadi lebih bersih dan memfokuskan perhatian pada kata-kata yang lebih relevan dan informatif dalam pemrosesan data teks selanjutnya. Berikut implementasi *code python* untuk proses *stopwords*.

```
from nltk.corpus import stopwords

def remove_stopwords(tokens):
    list_stopwords = nltk.corpus.stopwords.words('indonesian')
    list_stopwords.extend(['yg', 'dg', 'dgn', 'ny', 'd', 'u',
        'klo', 'kalo', 'amp', 'biar', 'bikin', 'bilang', 'gak', 'ga',
        'krn', 'nya', 'nih', 'sih', 'si', 'tau', 'tdk', 'tuh', 'utk',
        'ya', 'jd', 'jgn', 'sdh', 'aja', 'n', 't', 'p', 'ak',
        'nyg', 'hehe', 'pen', 'u', 'nan', 'loh', 'rt', '&', 'yah',
        'ri', 'dci', 'di', 'iims', 'ge', 'eeehhhh', 'cman', 'pj',
        'kyk', 'jrg', 'nyahnyoh', 'kya', 'hp', 'jm', 'n', 'ny',
        'ama', 'halah', 'entut', 'drun', 'yook', 'dkk', 'a', 'lg',
        'rd', 'do', 'aq', 'woee', 'q', 'ha', 'brow', 'de',
        'kq', 'imho', 'hmm', 'ssh', 'aa', 'e', 'tx', 'i', 'iot',
        'mr', 'co', 'rd', 'dr', 'imho', 'bb', 'eh', 'kl', 'koq',
        'ati', 'mw', 'lo', 'b', 'pt', 'up', 'aaamiin', 'aama', 'aat',
        'zhejiang', 'zzxxx', 'zack', 'zarka', 'zimbabwe', 'abang',
        'abb', 'abbas', 'abisin', 'abrek', 'yudha', 'yuhuu', 'yupss',
        'yurioshi', 'yusuf', 'ac', 'abri', 'abu', 'yosua', 'your',
        'youtube', 'youtubee', 'yuan', 'mudahhan', 'klu'])
    txt_stopword = pd.read_csv("stopwords.txt",
        names=["stopwords"], header=None)
    list_stopwords.extend(txt_stopword["stopwords"][0].split(' '))
    list_stopwords = set(list_stopwords)
    tokens = [word for word in tokens if not word in
        list_stopwords]
    return tokens

df['StopWord'] = df['Tokenize'].apply(remove_stopwords)
df.head()
```

Kode di atas adalah sebuah fungsi untuk menghapus stopwords dari daftar token atau kata-kata dalam teks menggunakan pustaka *Natural Language Toolkit* (NLTK). Pada kode tersebut mengimpor modul stopwords dari NLTK dan membaca daftar stopwords bahasa Indonesia yang sudah disediakan. Fungsi *remove_stopwords* akan menerima daftar token sebagai *input*, kemudian membuat daftar *stopwords* bahasa Indonesia dari NLTK dan juga menambahkan beberapa *stopwords* tambahan yang telah ditentukan secara manual pada *list_stopwords.extend*. Selain itu, fungsi juga membaca daftar stopwords

tambahan dari file '*stopwords.txt*' dan menambahkannya ke dalam daftar *stopwords*.

Setelah fungsi dibuat, maka selanjutnya akan diterapkan pada kolom '*Tokenize*' dari *DataFrame* menggunakan metode `apply`, dan hasilnya disimpan dalam kolom baru '*StopWord*' pada *DataFrame* yang berisi data teks komentar yang telah melalui proses *stopwords*. Berikut adalah hasil atau *output* dari penerapan *stopwords* yang dapat dilihat pada tabel 4.7. berikut.

Tabel 4.7. Hasil *preprocessing* pada tahap *stopword*

Tokenize	StopWord
['iya', 'semua', 'produsen', 'mobil', 'listrik', 'akan', 'masuk', 'lah', 'bos', 'wong', 'produsennya', 'dapat', 'subsidi']	['produsen', 'mobil', 'listrik', 'produsennya', 'subsidi']
['presiden', 'pak', 'jokowi', 'luar', 'biasa', 'kecerdasan', 'dan', 'prestasi', 'nya', 'bisa', 'menarik', 'perusahaan', 'luar', 'untuk', 'menggenjot', 'ekonomi', 'negara', 'dan', 'membuka', 'lapangan', 'kerja', 'yang', 'besar', 'buat', 'rakyat', 'indonesia', 'nanti', 'nya']	['presiden', 'jokowi', 'kecerdasan', 'prestasi', 'menarik', 'perusahaan', 'menggenjot', 'ekonomi', 'negara', 'membuka', 'lapangan', 'rakyat', 'indonesia']
['mantap', 'pak', 'jokowi', 'saat', 'nya', 'indonesia', 'penguasa', 'industri', 'modern']	['mantap', 'jokowi', 'indonesia', 'penguasa', 'industri', 'modern']
['rakyat', 'harus', 'bersatu', 'kita', 'sudah', 'lama', 'dijajah', 'saat', 'kita', 'mau', 'maju', 'kok', 'ada', 'yang', 'menghalangi', 'kemajuan', 'negara', 'kita', 'kita', 'harus', 'siap', 'berdarah', 'jangan', 'biarkan', 'organisasi', 'preman', 'dunia', 'memaksa', 'maksa', 'negara', 'kita']	['rakyat', 'bersatu', 'dijajah', 'maju', 'menghalangi', 'kemajuan', 'negara', 'berdarah', 'biarkan', 'organisasi', 'preman', 'dunia', 'memaksa', 'maksa', 'negara']
['maju', 'terus', 'jangan', 'mundur', 'kita', 'akan', 'maju', 'inshaallah']	['maju', 'mundur', 'maju']

e) *Stemming*

Pada tahap *stemming* ini bertujuan untuk mengubah sebuah kata ke dalam bentuk kata dasarnya dengan menghapus kata imbuhan di depan maupun imbuhan di belakang kata atau mereduksi kata-kata dalam teks menjadi bentuk dasar, sehingga kata-kata yang memiliki arti yang sama atau terkait akan dikelompokkan menjadi satu bentuk. Hal ini membantu dalam meningkatkan konsistensi analisis teks dan mengurangi dimensi data teks dengan menyatukan kata-kata yang serupa. Berikut implementasi *code python* untuk proses *stemming*.

```

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

def stemming(Text):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    Text = [stemmer.stem(word) for word in Text]
    return Text

df['Stemming'] = df['StopWord'].apply(stemming)
df.head()

```

Kode di atas adalah sebuah fungsi untuk melakukan stemming pada daftar token atau kata-kata dalam teks menggunakan pustaka *Sastrawi* untuk Bahasa Indonesia. Pustaka *Sastrawi* adalah pustaka pemrosesan bahasa alami (NLP) yang sering digunakan untuk *stemming* bahasa Indonesia. Fungsi *stemming* akan mengambil daftar token sebagai input dan menggunakan pustaka *Sastrawi* untuk membuat objek *stemmer*. Kemudian, fungsi ini mengiterasi melalui setiap kata dalam daftar token dan menerapkan proses *stemming* pada kata-kata tersebut dengan menggunakan metode *stem* dari objek *stemmer*. Setelah fungsi *stemming* dibuat, maka selanjutnya akan diterapkan pada kolom 'StopWord' dari *DataFrame* menggunakan metode *apply*, dan hasilnya disimpan dalam kolom baru 'Stemming' pada *DataFrame* yang berisi data teks komentar yang telah melalui proses *stemming*. Berikut adalah hasil atau *output* dari penerapan *stemming* yang dapat dilihat pada tabel 4.8. berikut.

Tabel 4.8. Hasil *preprocessing* pada tahap *stemming*

StopWord	Stemming
['produsen', 'mobil', 'listrik', 'produsennya', 'subsidi']	['produsen', 'mobil', 'listrik', 'produsen', 'subsidi']
['presiden', 'jokowi', 'kecerdasan', 'prestasi', 'menarik', 'perusahaan', 'menggenjot', 'ekonomi', 'negara', 'membuka', 'lapangan', 'rakyat', 'indonesia']	['presiden', 'jokowi', 'cerdas', 'prestasi', 'tarik', 'usaha', 'genjot', 'ekonomi', 'negara', 'buka', 'lapang', 'rakyat', 'indonesia']
['mantap', 'jokowi', 'indonesia', 'penguasa', 'industri', 'modern']	['mantap', 'jokowi', 'indonesia', 'kuasa', 'industri', 'modern']
['rakyat', 'bersatu', 'dijajah', 'maju', 'menghalangi', 'kemajuan', 'negara', 'berdarah', 'biarkan', 'organisasi', 'preman', 'dunia', 'memaksa', 'maksa', 'negara']	['rakyat', 'satu', 'jajah', 'maju', 'halang', 'maju', 'negara', 'darah', 'biar', 'organisasi', 'preman', 'dunia', 'paksa', 'maksa', 'negara']
['maju', 'mundur', 'maju']	['maju', 'mundur', 'maju']

Setelah didapatkan hasil *stemming* seperti ouput diatas, yang dimana data teks komentar masih dalam bentuk tokenisasi. Maka dari itu, tahap selanjutnya

ialah menggabungkan data teks yang terbagi menjadi beberapa fitur kata (token) menjadi kalimat utuh yang telah melalui tahap *preprocessing* untuk digunakan sebagai dataset pada tahap selanjutnya. Berikut implementasi dalam *code python* untuk menggabungkan data teks komentar menjadi kalimat yang utuh.

```
def join_text(Text):
    Text = " ".join([char for char in Text if char not in
string.punctuation])
    return Text

df['Comment'] = df['Stemming'].apply(join_text)
df.head()
```

Kode di atas adalah sebuah fungsi untuk melakukan penggabungan karakter dalam teks dengan menghapus tanda baca yang dimana menerima satu argumen yaitu *Text*. Dalam fungsi tersebut sebuah pemrosesan dilakukan menggunakan *list comprehension*, dimana setiap karakter dalam *Text* dicek apakah termasuk dalam tanda baca atau tidak menggunakan modul *string.punctuation*. Karakter-karakter yang tidak termasuk dalam tanda baca akan digabungkan kembali menjadi sebuah *string* dengan pemisah spasi menggunakan metode *join*. Kemudian fungsi ini diterapkan ke sebuah *DataFrame* dengan mengaplikasikan fungsi *join_text* pada kolom '*Stemming*' dari *DataFrame* tersebut menggunakan metode *apply*. Hasil dari pemrosesan ini kemudian disimpan kembali pada kolom baru bernama '*Comment*' dalam *DataFrame*. Berikut adalah hasil atau *output* dari penerapan *join text* yang dapat dilihat pada tabel 4.9. berikut.

Tabel 4.9. Hasil penerapan *join text*

Stemming	Comment
['produsen', 'mobil', 'listrik', 'produsen', 'subsidi']	produsen mobil listrik produsen subsidi
['presiden', 'jokowi', 'cerdas', 'prestasi', 'tarik', 'usaha', 'genjot', 'ekonomi', 'negara', 'buka', 'lapang', 'rakyat', 'indonesia']	presiden jokowi cerdas prestasi tarik usaha genjot ekonomi negara buka lapang rakyat indonesia
['mantap', 'jokowi', 'indonesia', 'kuasa', 'industri', 'modern']	mantap jokowi indonesia kuasa industri modern
['rakyat', 'satu', 'jajah', 'maju', 'halang', 'maju', 'negara', 'darah', 'biar', 'organisasi', 'preman', 'dunia', 'paksa', 'maksa', 'negara']	rakyat satu jajah maju halang maju negara darah biar organisasi preman dunia paksa maksa negara
['maju', 'mundur', 'maju']	maju mundur maju

4.3.2 Pembagian Dataset

Setelah melalui tahap *preprocessing* dalam membersihkan data teks komentar yang dimana menghasilkan data bersih sebanyak 3509 data yang terdiri dari 1905 kelas positif dan 1606 kelas negatif. Selanjutnya, data bersih yang telah didapatkan dibagi menjadi data latih dan data uji. Data latih digunakan untuk melakukan ekstraksi fitur TF-IDF serta melatih model klasifikasi *Naïve Bayes*, sedangkan data uji digunakan untuk mengukur kinerja model agar dapat melihat persentase algoritma klasifikasi yang berhasil mengklasifikasikan dengan benar. Berikut dapat dilihat pada tabel 4.10 beberapa perbandingan data latih dan data uji yang dilakukan pada penelitian ini beserta akurasi yang dihasilkan.

Tabel 4.10. Perbandingan penggunaan data latih dan data uji

Perbandingan	Data Latih		Data Uji		Total Data		Akurasi
	Positif	Negatif	Positif	Negatif	Data Latih	Data Uji	
90 : 10	1723	1435	181	170	3158	351	81%
80 : 20	1535	1272	369	333	2807	702	78%
70 : 30	1357	1099	547	506	2456	1053	77%
60 : 40	1155	950	749	655	2105	1404	76%
Total Keseluruhan Data					3509		

Berdasarkan tabel 4.10 menunjukkan bahwa dari keempat perbandingan yang digunakan didapatkan bahwa pada perbandingan 90 : 10 atau 90% untuk data latih dan 10% untuk data uji menghasilkan performa atau tingkat nilai akurasi terbaik sehingga perbandingan ini dipilih untuk digunakan dalam melakukan analisis sentimen opini publik terhadap kendaraan listrik menggunakan algoritma *naïve bayes classifier* dan ekstraksi fitur dengan *tf-idf*.

Dari beberapa perbandingan pembagian dataset pada tabel 4.10 jumlah data latih dapat mempengaruhi hasil prediksi pada sistem. Dikarenakan semakin banyaknya data latih maka sistem memiliki kosakata yang lebih banyak dan beragam, sehingga ketika ada data baru maka sistem bisa lebih tepat dalam memprediksi kelas sentimen dan menghasilkan performa model yang lebih baik (Sasmita dkk, 2022).

Kemudian, dari pembagian dataset ini akan disimpan kedalam *file excel* untuk digunakan pada pengolahan data secara terus menerus tanpa harus mengulangi dalam membagi data. Berikut implementasi *code python* untuk pembagian *dataset*.

```

train_df, test_df = train_test_split(dfclean, test_size=0.1,
random_state=42)

train_df.to_excel('train.xlsx', index=False)
test_df.to_excel('test.xlsx', index=False)
train = pd.read_excel('train.xlsx')
test = pd.read_excel('test.xlsx')

```

Kode di atas adalah implementasi proses pemisahan dataset dalam pemrosesan data menggunakan bahasa pemrograman *Python* dengan *library pandas* dan *scikit-learn*. Pertama, dataset dibagi menjadi dua bagian, yaitu `train_df` dan `test_df`, dengan perbandingan 90% data latih dan 10% data uji (90 : 20). Penggunaan nilai 0.1 pada parameter `test_size` menunjukkan alokasi persentase data uji yang tepat. Selanjutnya, kedua dataset tersebut disimpan dalam format *file Excel* terpisah dengan nama *train.xlsx* dan *test.xlsx*. Setelah penyimpanan, dataset ini dibaca kembali dari *file Excel* dan dimuat ke dalam variabel `train` dan `test`. Dengan ini, dataset dapat digunakan dalam tahap pelatihan dan pengujian model *machine learning*.

4.3.3 Ekstraksi Fitur TF-IDF

Pada tahap ekstraksi fitur ini bertujuan untuk mengubah dataset kedalam sebuah representasi angka (numerik) dengan menggunakan *library python* yaitu *Scikit Learn* dengan metode *CountVectorizer* untuk melihat berapa kali fitur kata muncul dalam suatu *document* pada *dataset*, lalu kemudian dilanjutkan dengan dilakukan pembobotan kata dengan menggunakan metode *TF-IDF* yang juga merupakan metode dalam *Scikit Learn* yang dimana tujuan dilakukannya ekstraksi fitur dengan *TF-IDF* adalah untuk mengidentifikasi kata-kata yang paling relevan dan berpengaruh dalam suatu dokumen berdasarkan nilai pembobotan kata yang didapatkan. Berikut dapat dilihat implementasi kode *python* untuk ekstraksi fitur dengan *CountVectorizer* dan *TF-IDF*.

```

from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
import pandas as pd

# Membaca data latih dan data test
train_df = pd.read_excel('train.xlsx')
test_df = pd.read_excel('test.xlsx')

# Menginisialisasi objek CountVectorizer
count_vec = CountVectorizer()

# Menggunakan fit_transform pada data latih untuk mendapatkan
vektor count

```

```

train_count =
count_vec.fit_transform(train_df['Comment']).toarray()

# Menginisialisasi objek TfidfVectorizer
tfidf_vec = TfidfVectorizer(max_features=1500)

# Menggunakan fit_transform pada data latih untuk mendapatkan
vektor TF-IDF
train_tfidf =
tfidf_vec.fit_transform(train_df['Comment']).toarray()

# Membaca teks pada data test
comment_text = test_df['Comment']

# Menggunakan transform pada data test dengan CountVectorizer
test_count = count_vec.transform(comment_text).toarray()

# Menggunakan transform pada data test yang sudah dalam bentuk
vektor count untuk mendapatkan vektor TF-IDF
test_tfidf = tfidf_vec.transform(comment_text).toarray()

```

Kode diatas menunjukkan bahwa ekstraksi fitur dilakukan dengan kombinasi dua metode yaitu *CountVectorizer* dan TF-IDF. Pada kode tersebut dapat dilihat bahwa data latih dan data uji dibaca dari file Excel menggunakan pandas dan disimpan dalam dua *DataFrame*, yaitu *train_df* dan *test_df*. Selanjutnya, objek *CountVectorizer* dan *TfidfVectorizer* diinisialisasi. Kemudian, fitur-fitur dari data latih diubah menjadi vektor *count* menggunakan *CountVectorizer* dengan menggunakan *fit_transform*. Hasil dari vektor *count* data latih disimpan dalam *train_count*. Selanjutnya, fitur-fitur dari data latih diubah menjadi vektor *TF-IDF* menggunakan *TfidfVectorizer* dengan menggunakan *fit_transform*. Hasil dari vektor *TF-IDF* data latih disimpan dalam *train_tfidf*.

Setelah itu, teks pada data uji diambil dan diubah menjadi *vektor count* menggunakan *CountVectorizer* dengan menggunakan *transform*. Hasil dari vektor *count* data uji disimpan dalam *test_count*. Selanjutnya, teks pada data uji diubah menjadi vektor *TF-IDF* menggunakan *TfidfVectorizer* dengan menggunakan *transform*. Hasil dari vektor *TF-IDF* data uji disimpan dalam *test_tfidf*. Berikut adalah hasil atau output dari penerapan TF-IDF dalam bentuk *DataFrame*.

	WI	WJ	WK	WL	WM	WN	WO	WP	WQ	WR	WS	WT	WU
1	kembang	kemenko	emenperi	kenal	kencang	kendala	kendali	kendara	keras	keren	kereta	kerja	keruk
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0,19046	0	0	0	0	0
4	0	0	0	0	0	0	0	0,21037	0	0	0	0	0
5	0,25965	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0,23157	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0,26926	0	0	0	0	0	0	0,11152	0	0	0	0	0

Gambar 4.5. Tampilan data ekstraksi fitur dengan pembobotan *TF-IDF*

Pada Gambar 4.5 diatas dapat diartikan bahwa setiap barisnya mewakili setiap data komentar yang menjadi *dataset*. Kemudian setiap kolomnya merupakan seluruh fitur (kata). Pembobotan fitur kata dengan TF-IDF pada gambar diatas meggunakan *library python* yaitu *TfidfVectorizer*. Dalam gambar tersebut merepresentasikan matriks TF-IDF (*Term Frequency-Inverse Document Frequency*) setelah proses ekstraksi fitur menggunakan metode *TfidfVectorizer* pada teks. Matriks ini menunjukkan bagaimana kata-kata dalam teks direpresentasikan dalam bentuk vektor numerik berdasarkan konsep TF-IDF. Nilai dalam matriks tersebut adalah bobot TF-IDF untuk setiap kata dalam setiap dokumen. Nilai-nilai ini menunjukkan seberapa penting kata tersebut dalam dokumen teks komentar. Jika nilai bobot TF-IDF suatu kata dalam suatu dokumen mendekati 0, itu berarti kata tersebut tidak begitu relevan dalam dokumen tersebut. Sedangkan, jika nilai TF-IDF adalah 0 berarti sebagian besar dokumen mungkin tidak mengandung kata-kata tertentu. Berikut dapat dilihat pada tabel 4.11 sampel daftar nilai term (kata) tertinggi.

Tabel 4.11. Sampel daftar nilai pembobotan kata *Tf-Idf*

Kata	Nilai <i>Tf-Idf</i>
listrik	0,085291
kendara	0,073589
mobil	0,056195
indonesia	0,036669
subsidi	0,034352
baterai	0,022980
perintah	0,022494
motor	0,022307
beli	0,021205
pakai	0,020124

Untuk lebih jelasnya, berikut akan dipaparkan bagaimana proses ekstraksi fitur *Tf-Idf* secara perhitungan manual dalam persamaan matematika menggunakan 3 contoh sampel data yang dapat dilihat dalam penjabaran berikut.

Tabel 4.12. Sampel *dataset* komentar

<i>Document (D)</i>	Komentar
<i>D1</i>	mobil listrik harga mahal
<i>D2</i>	alih kendara hemat ramah lingkungan
<i>D3</i>	insentif kendara listrik jalan macet

Pada sampel data komentar diatas terdapat beberapa kata baku yang dianggap informatif yaitu “mobil”, “listrik”, “harga”, “mahal”, “alih”, “kendara”, “hemat”, “ramah”, “lingkung”, “insentif”, “jalan” dan “macet”. Pada tahap selanjutnya tiap dokumen diwujudkan sebagai sebuah vektor dengan elemen, jika terdapat kata yang bersangkutan didalam dokumen maka diberi nilai 1, jika tidak ada maka diberi nilai 0. Sebagai contoh dapat dilihat pada Tabel 4.13 berikut.

Tabel 4.13. Sampel jumlah kemunculan fitur kata pada setiap dokumen

	mobil	listrik	harga	mahal	alih	kendara	hemat	ramah	lingkung	insentif	jalan	macet
D1	1	1	1	1	0	0	0	0	0	0	0	0
D2	0	0	0	0	1	1	1	1	1	0	0	0
D3	0	1	0	0	0	1	0	0	0	1	1	1

Data yang sudah dalam representasi *vector* yang dilakukan dengan *CountVectorizer*, kemudian akan dilanjutkan untuk dilakukan ekstraksi fitur dengan TF-IDF yang dimana akan dihitung dengan rumus *Tf-Idf* sehingga menghasilkan nilai fitur kata dalam representasi *vector* yang sudah terbobot. Adapun *TF* (*Term Frequency*) adalah frekuensi dari kemunculan sebuah *term* dalam dokumen yang bersangkutan, sedangkan *IDF* (*Inverse Document Frequency*) merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan. Berikut langkah-langkah proses perhitungan pembobotan kata dengan *Tf-Idf*.

- **Langkah 1: Menghitung *Term Frequency* (TF)**

Term Frequency (TF) dilakukan untuk mengukur seberapa sering sebuah kata muncul dalam dokumen. Menggunakan Rumus (2-1), kita dapat menghitung nilai bobot TF untuk setiap kata dalam setiap dokumen:

- Dokumen 1 (D1):

Jumlah kata = 4

$$TF_{(mobil, d1)} = \frac{n_{t,d1}}{N} = \frac{1}{4} = 0,25$$

$$TF_{(listrik, d1)} = \frac{n_{t,d1}}{N} = \frac{1}{4} = 0,25$$

$$TF_{(harga, d1)} = \frac{n_{t,d1}}{N} = \frac{1}{4} = 0,25$$

$$TF_{(mahal, d1)} = \frac{n_{t,d1}}{N} = \frac{1}{4} = 0,25$$

- Dokumen 2 (D2):

Jumlah kata = 5

$$TF_{(alih, d2)} = \frac{n_{t,d1}}{N} = \frac{1}{6} = 0,17$$

$$TF_{(kendara, d2)} = \frac{n_{t,d1}}{N} = \frac{1}{6} = 0,17$$

$$TF_{(hemat, d2)} = \frac{n_{t,d1}}{N} = \frac{1}{6} = 0,17$$

$$TF_{(ramah, d2)} = \frac{n_{t,d1}}{N} = \frac{1}{6} = 0,17$$

$$TF_{(lingkung, d2)} = \frac{n_{t,d1}}{N} = \frac{1}{6} = 0,17$$

- Dokumen 3 (D3):

Jumlah kata = 5

$$TF_{(insentif, d3)} = \frac{n_{t,d1}}{N} = \frac{1}{5} = 0,2$$

$$TF_{(kendara, d3)} = \frac{n_{t,d1}}{N} = \frac{1}{5} = 0,2$$

$$TF_{(listrik, d3)} = \frac{n_{t,d1}}{N} = \frac{1}{5} = 0,2$$

$$TF_{(jalan, d3)} = \frac{n_{t,d1}}{N} = \frac{1}{5} = 0,2$$

$$TF_{(macet, d3)} = \frac{n_{t,d1}}{N} = \frac{1}{5} = 0,2$$

Sehingga didapatkan hasil perhitungan nilai *Term Frequency (TF)* seperti tabel 4.14 berikut.

Tabel 4.14. Nilai perhitungan *Term Frequency (TF)*

	<i>D1</i>	<i>D2</i>	<i>D3</i>
mobil	0,25	0,0	0,0
listrik	0,25	0,0	0,2
harga	0,25	0,0	0,0
mahal	0,25	0,0	0,0
alih	0,0	0,17	0,0
kendara	0,0	0,17	0,2
hemat	0,0	0,17	0,0
ramah	0,0	0,17	0,0
lingkung	0,0	0,17	0,0
insentif	0,0	0,0	0,2
jalan	0,0	0,0	0,2
macet	0,0	0,0	0,2

- **Langkah 2: Menghitung *Inverse Document Frequency* (IDF)**

Inverse Document Frequency (IDF) dilakukan untuk mengukur seberapa penting sebuah kata di dalam koleksi dokumen. Menggunakan Rumus (2-2), kita dapat menghitung nilai bobot *IDF* untuk setiap kata:

- Total dokumen (N) = 3
- Menghitung nilai DF (jumlah dokumen yang mengandung kata tersebut):

$$DF(\text{"mobil"}) = 1 \text{ (muncul pada D1)}$$

$$DF(\text{"listrik"}) = 2 \text{ (muncul pada D1 dan D3)}$$

$$DF(\text{"harga"}) = 1 \text{ (muncul pada D1)}$$

$$DF(\text{"mahal"}) = 1 \text{ (muncul pada D1)}$$

$$DF(\text{"alih"}) = 1 \text{ (muncul pada D2)}$$

$$DF(\text{"kendara"}) = 2 \text{ (muncul pada D2 dan D3)}$$

$$DF(\text{"hemat"}) = 1 \text{ (muncul pada D2)}$$

$$DF(\text{"ramah"}) = 1 \text{ (muncul pada D2)}$$

$$DF(\text{"lingkung"}) = 1 \text{ (muncul pada D2)}$$

$$DF(\text{"insentif"}) = 1 \text{ (muncul pada D3)}$$

$$DF(\text{"jalan"}) = 1 \text{ (muncul pada D3)}$$

$$DF(\text{"macet"}) = 1 \text{ (muncul pada D3)}$$

- Menghitung nilai *Inverse Document Frequency* (IDF):

$$idf_{(mobil)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(listrik)} = \log \left(\frac{3}{2} \right) = 0,1761$$

$$idf_{(harga)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(mahal)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(alih)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(kendara)} = \log \left(\frac{3}{2} \right) = 0,1761$$

$$idf_{(hemat)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(ramah)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(lingkung)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(insentif)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(jalan)} = \log \left(\frac{3}{1} \right) = 0,4771$$

$$idf_{(macet)} = \log \left(\frac{3}{1} \right) = 0,4771$$

Sehingga didapatkan hasil perhitungan nilai *Inverse Document Frequency (IDF)* seperti tabel 4.15 berikut.

Tabel 4.15. Nilai perhitungan *Inverse Document Frequency (IDF)*

T (Term)	DF (Document Frequency)	IDF (Inverse Document Frequency)
mobil	1	$\log \left(\frac{3}{1} \right) = 0,4771$
listrik	2	$\log \left(\frac{3}{2} \right) = 0,1761$
harga	1	$\log \left(\frac{3}{1} \right) = 0,4771$
mahal	1	$\log \left(\frac{3}{1} \right) = 0,4771$
alih	1	$\log \left(\frac{3}{1} \right) = 0,4771$
kendara	2	$\log \left(\frac{3}{2} \right) = 0,1761$
hemat	1	$\log \left(\frac{3}{1} \right) = 0,4771$
ramah	1	$\log \left(\frac{3}{1} \right) = 0,4771$
lingkung	1	$\log \left(\frac{3}{1} \right) = 0,4771$
insentif	1	$\log \left(\frac{3}{1} \right) = 0,4771$
jalan	1	$\log \left(\frac{3}{1} \right) = 0,4771$
macet	1	$\log \left(\frac{3}{1} \right) = 0,4771$

- **Langkah 3: Menghitung *TF-IDF***

Menggunakan Rumus (2-3), kita dapat menghitung nilai pembobotan kata dengan *TF-IDF* untuk setiap kata dalam setiap dokumen:

- Dokumen 1 (D1):

$$tfidf_{(mobil, d1)} = 0,25 \times 0,4771 = 0,119275$$

$$tfidf_{(listrik, d1)} = 0,25 \times 0,1761 = 0,044025$$

$$tfidf_{(harga, d1)} = 0,25 \times 0,4771 = 0,119275$$

$$tfidf_{(mahal, d1)} = 0,25 \times 0,4771 = 0,119275$$

- Dokumen 2 (D2):

$$tfidf_{(alih, d2)} = 0,17 \times 0,4771 = 0,081107$$

$$tfidf_{(kendara, d2)} = 0,17 \times 0,1761 = 0,029937$$

$$tfidf_{(hemat, d2)} = 0,17 \times 0,4771 = 0,081107$$

$$tfidf_{(ramah, d2)} = 0,17 \times 0,4771 = 0,081107$$

$$tfidf_{(lingkung, d2)} = 0,17 \times 0,4771 = 0,081107$$

- Dokumen 3 (D3):

$$tfidf_{(insentif, d3)} = 0,2 \times 0,4771 = 0,09542$$

$$tfidf_{(kendara, d3)} = 0,2 \times 0,1761 = 0,03522$$

$$tfidf_{(listrik, d3)} = 0,2 \times 0,1761 = 0,03522$$

$$tfidf_{(jalan, d3)} = 0,2 \times 0,4771 = 0,09542$$

$$tfidf_{(macet, d3)} = 0,2 \times 0,4771 = 0,09542$$

Sehingga didapatkan hasil perhitungan nilai *Term Frequency-Inverse Document Frequency (TF-IDF)* seperti tabel 4.16 berikut.

Tabel 4.16. Hasil perhitungan *TF-IDF*

Term	TF			DF	IDF	TF – IDF = TF x IDF		
	D1	D2	D3			D1	D2	D3
mobil	0,25	0	0	1	0,4771	0,119275	0	0
listrik	0,25	0	0,2	2	0,1761	0,044025	0	0,03522
harga	0,25	0	0	1	0,4771	0,119275	0	0
mahal	0,25	0	0	1	0,4771	0,119275	0	0
alih	0	0,17	0	1	0,4771	0	0,081107	0
kendara	0	0,17	0,2	2	0,1761	0	0,029937	0,03522
hemat	0	0,17	0	1	0,4771	0	0,081107	0
ramah	0	0,17	0	1	0,4771	0	0,081107	0
lingkung	0	0,17	0	1	0,4771	0	0,081107	0
insentif	0	0	0,2	1	0,4771	0	0	0,09542
jalan	0	0	0,2	1	0,4771	0	0	0,09542
macet	0	0	0,2	1	0,4771	0	0	0,09542

4.3.4 Klasifikasi Naïve Bayes

Dalam melakukan klasifikasi dengan menggunakan algoritma *naïve bayes*, *dataset* komentar pada penelitian ini harus melalui tahap pembagian *dataset* terlebih dahulu menjadi data latih dan data uji. Seperti yang telah dilakukan pada tahap sebelumnya yang dimana dilakukan pembagian data latih dan data uji dengan perbandingan 90 : 10 atau 90% untuk data latih dan 10% untuk data uji. Perbandingan tersebut dipilih atas dasar hasil percobaan yang dilakukan yang dimana sebelumnya telah dilakukan percobaan dengan beberapa perbandingan pembagian data latih dan data uji untuk menemukan hasil perbandingan yang memiliki akurasi terbaik yang dapat diterapkan pada analisis sentimen opini publik terhadap kendaraan listrik menggunakan algoritma *naïve bayes classifier*. Kemudian, setelah dilakukan pembagian *dataset* selanjutnya dilakukan ekstraksi fitur menggunakan *Tf-Idf* untuk mendapatkan pembobotan dari data latih dan data uji yang dimana hasil data yang telah melalui tahap ekstraksi fitur menggunakan *Tf-Idf* nantinya akan digunakan untuk melatih dan menguji model klasifikasi *naïve bayes*. Berikut beberapa proses dalam melatih dan menguji model menggunakan algoritma *Naïve Bayes Classifier* dalam analisis sentimen opini publik terhadap kendaraan listrik.

a) Melatih model *Naïve Bayes*

Berikut ini adalah tahap membangun model pelatihan dengan menggunakan algoritma *Naïve Bayes Classifier* yang dapat dilihat pada kode python berikut.

```
from Nbc import MultinomialNBC  
  
x_train = train_tfidf  
y_train = train_labels  
  
modelNBC = MultinomialNBC()  
modelNBC.fit(x_train, y_train)
```

Kode diatas menunjukkan implementasi dari proses pelatihan sebuah model klasifikasi *Naïve Bayes* dengan menggunakan kelas `MultinomialNBC` yang telah didefinisikan sebelumnya. Kemudian, dalam kode diatas data latih dalam bentuk matriks dari ekstraksi fitur *Tf-Idf* `train_tfidf` disimpan dalam variabel `x_train` dan label kelas dari data latih yang disimpan pada variabel `y_train` di inisialisasi untuk melatih model latih *Naïve Bayes*. Selanjutnya dilakukan pembuatan objek `modelNBC` dari kelas `MultinomialNBC`, yang dimana untuk

melatih model dengan memanggil metode `fit(x_train, y_train)`. Setelah melalui tahap ini, model siap untuk digunakan dalam memprediksi kelas pada data uji. Dengan memahami proses ini, kita dapat memanfaatkan model *Naive Bayes* yang telah dilatih untuk mengklasifikasikan data baru dengan memanfaatkan informasi yang diperoleh dari data latih.

b) Menguji model *Naïve Bayes*

Berikut ini adalah tahap menguji model dengan menggunakan model algoritma *Naïve Bayes Classifier* yang telah dilatih sebelumnya. Berikut implementasinya pada kode python.

```
x_test = test_tfidf
y_test = test_labels

predictNBC = modelNBC.predict(x_test)
predictNBC
```

Kode diatas menunjukkan bagaimana model klasifikasi *Naive Bayes* yang telah dilatih digunakan untuk memprediksi kelas pada data uji. Pada kode diatas data uji dalam bentuk matriks dari ekstraksi fitur *Tf-Idf* `test_tfidf` disimpan dalam variabel `x_test` dan label kelas dari data latih yang disimpan pada variabel `y_test` di inisialisasi untuk menguji model yang dilatih sebelumnya yaitu `modelNBC` digunakan untuk melakukan prediksi kelas pada data uji dengan memanggil metode `predict(x_test)`. Kemudian, hasil prediksi bisa digunakan untuk mengukur performa model, membandingkannya dengan label kelas yang sebenarnya, serta mendapatkan wawasan lebih lanjut tentang bagaimana model berfungsi pada data yang belum pernah dilihat sebelumnya. Berikut sampel output atau hasil klasifikasi yang dapat dilihat pada tabel 4.17.

Tabel 4.17. Hasil prediksi klasifikasi model *Naïve Bayes*

<i>Comment</i>	<i>Label</i>	<i>Classification</i>
pabrik mobil konvensional mobil konvensional untung	Positif	Negatif
energi baru tinggal mau perintah sungai indonesia ribu sumber tenaga matahari angin berimpah bukti maju mafia koruptor cokol perintah	Negatif	Negatif
perintah resmi terbit atur beri bantu beli kendara motor listrik bas baterai	Positif	Positif
sedih negara asai pihak guna kendara listrik guna krl subsidi kendara listrik triliun rupiah gampang gelontor gilir nambah gerbong bekas ditolakin	Negatif	Negatif

jelas ahli ekonomi tepat subsidi usaha mobil listrik buka maka langsung subsidi muka bantu usaha usaha mobil listrik salah usaha toba anak usah milik opung kali opung untung milik jokowi opung untung jokowi singkat usaha kelas sumatera utara usaha kenal percaya jokowi tajir limbah bijak bijak jokowi kamuflase rakyat bangsa kartu pintar kartu pra minerba omnibus law ujung balik duga proyek kaya kroni	Negatif	Negatif
jalan bisnis mobil listrik starup main alih	Positif	Positif
mobil buang mesin ganti dinamo batrae carger solar sell	Negatif	Negatif
kelas toyota gampang mobil listrik	Positif	Positif
masyarakat bus susah ngerubah biasa masyarakat warung meter pilih motor subsidi kendara listrik kurang guna bbm subsidi ngehemat anggaran negara jangka	Positif	Positif
situ ngotot bangun ikn subsidi kendara listrik nyedot anggaran	Negatif	Negatif

Berdasarkan proses klasifikasi yang dilakukan diatas menggunakan konsep probabilitas atar kalimat terhadap setiap kelas agar mampu menghasilkan prediksi data yang dimasukkan. Data uji yang digunakan pada penelitian ini sebanyak 351 data yang akan digunakan untuk menguji model klasifikasi dengan menggunakan algoritma *Naïve Bayes Classifier*. Untuk lebih jelasnya, berikut akan dipaparkan bagaimana algoritma *Naïve Bayes* dalam melakukan pengklasifikasian secara perhitungan manual dalam konsep probabilitas dengan menggunakan 5 sample data latih dan 3 sample data uji yang dapat dilihat dalam penjabaran berikut.

Tabel 4.18. Sampel data latih dalam klasifikasi *Naïve Bayes Classifier*

No	Comment	Label
1	moga mobil listrik anak bangsa saing mobil listrik negara	Positif
2	buru alih kendara listrik	Positif
3	hadir motor listrik jalan penuh sesak motor	Negatif
4	ngumpulin duit pajak rakyat kasih subsidi kaya goblok	Negatif
5	subsidi konversi kendara listrik ancam bencana lingkung hidup	Negatif

Tabel 4.19. Sampel data uji dalam klasifikasi *Naïve Bayes Classifier*

No	Comment	Label
1	kaya subsidi mobil listrik bodoh	?
2	beli kendara listrik murah subsidi	?
3	indonesia hebat maju industri otomotif	?

Berdasarkan tabel diatas menunjukkan bahwa data telah dibagi menjadi 2 *subset* yaitu data latih pada tabel 4.18 dan data uji pada tabel 4.19. Berikut adalah proses perhitungan klasifikasi kelas sentimen dengan menggunakan metode *Naive Bayes* berdasarkan tabel diatas.

- Menghitung probabilitas dalam dokumen data latih pada setiap kelas positif dan negatif berdasarkan tabel 4.18.

Diketahui:

- Jumlah dokumen kelas Positif = 2
 - Jumlah dokumen kelas Negatif = 3
 - Total semua dokumen = 5
 - Total kata dalam dokumen kelas Positif = 13
 - Total kata dalam dokumen kelas Negatif = 23
 - Total semua kata dalam dokumen = 36
- Menghitung probabilitas prior (P(H)) dengan menggunakan rumus persamaan (2-9) dan (2-10)
 - $P(Positif) = \frac{2}{5} = 0,4$
 - $P(Negatif) = \frac{3}{5} = 0,6$

Dengan mengacu pada data yang telah diketahui diatas, tahap selanjutnya ialah menghitung probabilitas kata dalam setiap kelas dan mengklasifikasikan dokumen dengan persamaan metode *Naive Bayes*, dimana dapat dilihat pada proses berikut:

- Pengujian Data Uji 1:

Tabel 4.20. Sampel data uji pertama

Comment	Label
kaya subsidi mobil listrik bodoh	?

- Menghitung probabilitas *likelihood* (P(X|H)) dari dokumen uji dengan menggunakan rumus persamaan (2-11) dan (2-12).
 - Kelas Positif

$$P(kaya | Positif) = \frac{0+1}{13+36} = 0,020$$

$$P(subsidi | Positif) = \frac{0+1}{13+36} = 0,020$$

$$P(mobil | Positif) = \frac{2+1}{13+36} = 0,061$$

$$P(\text{listrik} | \text{Positif}) = \frac{3+1}{13+36} = 0,087$$

$$P(\text{bodoh} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

- Kelas Negatif

$$P(\text{kaya} | \text{Negatif}) = \frac{1+1}{23+36} = 0,033$$

$$P(\text{subsidi} | \text{Negatif}) = \frac{2+1}{23+36} = 0,051$$

$$P(\text{mobil} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

$$P(\text{listrik} | \text{Negatif}) = \frac{2+1}{23+36} = 0,051$$

$$P(\text{bodoh} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

- Menghitung probabilitas total ($P(X)$) untuk kelas Negatif ($P(X | \text{Negatif})$) dan kelas Positif ($P(X | \text{Positif})$) kemudian menghitung probabilitas total $P(X)$ dengan persamaan (2-13).

- Positif

$$P(X | \text{Positif}) = 0,020 \times 0,020 \times 0,061 \times 0,087 \times 0,020$$

$$P(X | \text{Positif}) = 0,000000021$$

- Negatif

$$P(X | \text{Negatif}) = 0,033 \times 0,051 \times 0,017 \times 0,051 \times 0,017$$

$$P(X | \text{Negatif}) = 0,000000079$$

- Probabilitas Total $P(X)$

$$P(X) = 0,000000021 \times 0,4 + 0,000000079 \times 0,6$$

$$P(X) = 0,0000000084 + 0,0000000474$$

$$P(X) = 0,0000000558$$

- Mengklasifikasikan dokumen uji untuk kelas Positif dan Negatif dari perhitungan probabilitas posterior $P(H/X)$ menggunakan rumus persamaan Naïve Bayes (2-8).

- Positif

$$P(\text{Positif} | X) = \frac{P(X | \text{Positif}) P(\text{Positif})}{P(X)}$$

$$P(\text{Positif} | X) = \frac{0,000000021 \times 0,4}{0,0000000558}$$

$$P(\text{Positif} | X) = 0,151$$

- Negatif

$$P(\text{Negatif} | X) = \frac{P(X | \text{Negatif}) P(\text{Negatif})}{P(X)}$$

$$P(\text{Negatif} | X) = \frac{0,000000079 \times 0,6}{0,0000000558}$$

$$P(\text{Negatif} | X) = 0,849$$

Dari perhitungan berdasarkan nilai probabilitas posterior diatas, maka diperoleh hasil 0,151 untuk kelas Positif dan 0,849 untuk kelas Negatif. Karena nilai probabilitas posterior pada kelas negatif memiliki nilai lebih besar dibandingkan dengan nilai probabilitas posterior kelas positif. Maka, data uji ke-1 terklasifikasi kedalam kelas Negatif.

- Pengujian Data Uji 2:

Tabel 4.21. Sampel data uji kedua

Comment	Label
beli kendaraan listrik murah subsidi	?

- o Menghitung probabilitas *likelihood* ($P(X|H)$) dari dokumen uji dengan menggunakan rumus persamaan (2-11) dan (2-12).

- Kelas Positif

$$P(\text{beli} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

$$P(\text{kendaraan} | \text{Positif}) = \frac{1+1}{13+36} = 0,041$$

$$P(\text{listrik} | \text{Positif}) = \frac{3+1}{13+36} = 0,087$$

$$P(\text{murah} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

$$P(\text{subsidi} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

- Kelas Negatif

$$P(\text{beli} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

$$P(\text{kendaraan} | \text{Negatif}) = \frac{1+1}{23+36} = 0,034$$

$$P(\text{listrik} | \text{Negatif}) = \frac{2+1}{23+36} = 0,051$$

$$P(\text{murah} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

$$P(\text{subsidi} | \text{Negatif}) = \frac{1+1}{23+36} = 0,034$$

- Menghitung probabilitas total ($P(X)$) untuk kelas Negatif ($P(X | \text{Negatif})$) dan kelas Positif ($P(X | \text{Positif})$) kemudian menghitung probabilitas total $P(X)$ dengan persamaan (2-13).

- Positif

$$P(X | \text{Positif}) = 0,020 \times 0,041 \times 0,087 \times 0,020 \times 0,020$$

$$P(X | \text{Positif}) = 0,000000028536$$

- Negatif

$$P(X | \text{Negatif}) = 0,017 \times 0,034 \times 0,051 \times 0,017 \times 0,034$$

$$P(X | \text{Negatif}) = 0,000000017038284$$

- Probabilitas Total $P(X)$

$$P(X) = 0,000000028536 \times 0,4 + 0,000000017038284 \times 0,6$$

$$P(X) = 0,00000001141 + 0,0000000102229$$

$$P(X) = 0,0000000216329$$

- Mengklasifikasikan dokumen uji untuk kelas Positif dan Negatif dari perhitungan probabilitas posterior $P(H/X)$ menggunakan rumus persamaan Naïve Bayes (2-8).

- Positif

$$P(\text{Positif} | X) = \frac{P(X | \text{Positif}) P(\text{Positif})}{P(X)}$$

$$P(\text{Positif} | X) = \frac{0,000000028536 \times 0,4}{0,0000000216329}$$

$$P(\text{Positif} | X) = 0,527$$

- Negatif

$$P(\text{Negatif} | X) = \frac{P(X | \text{Negatif}) P(\text{Negatif})}{P(X)}$$

$$P(\text{Negatif} | X) = \frac{0,000000017038284 \times 0,6}{0,0000000216329}$$

$$P(\text{Negatif} | X) = 0,473$$

Dari perhitungan berdasarkan nilai probabilitas posterior diatas, maka diperoleh hasil 0,527 untuk kelas Positif dan 0,473 untuk kelas Negatif. Karena nilai probabilitas posterior pada kelas positif memiliki nilai lebih besar dibandingkan dengan nilai probabilitas posterior kelas negatif. Maka, data uji ke-2 terklasifikasi kedalam kelas Positif.

- Pengujian Data Uji 3:

Tabel 4.22. Sampel data uji ketiga

Comment	Label
indonesia hebat maju industri otomotif	?

- o Menghitung probabilitas *likelihood* ($P(X|H)$) dari dokumen uji dengan menggunakan rumus persamaan (2-11) dan (2-12).

- Kelas Positif

$$P(\text{indonesia} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

$$P(\text{hebat} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

$$P(\text{maju} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

$$P(\text{industri} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

$$P(\text{otomotif} | \text{Positif}) = \frac{0+1}{13+36} = 0,020$$

- Kelas Negatif

$$P(\text{indonesia} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

$$P(\text{hebat} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

$$P(\text{maju} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

$$P(\text{industri} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

$$P(\text{otomotif} | \text{Negatif}) = \frac{0+1}{23+36} = 0,017$$

- o Menghitung probabilitas total ($P(X)$) untuk kelas Negatif ($P(X | \text{Negatif})$) dan kelas Positif ($P(X | \text{Positif})$) kemudian menghitung probabilitas total $P(X)$ dengan persamaan (2-13).

- Positif

$$P(X | Positif) = 0,020 \times 0,020 \times 0,020 \times 0,020 \times 0,020$$

$$P(X | Positif) = 0,0000000032$$

- Negatif

$$P(X | Negatif) = 0,017 \times 0,017 \times 0,017 \times 0,017 \times 0,017$$

$$P(X | Negatif) = 0,0000000014$$

- Probabilitas Total $P(X)$

$$P(X) = 0,0000000032 \times 0,4 + 0,0000000014 \times 0,6$$

$$P(X) = 0,00000000128 + 0,00000000084$$

$$P(X) = 0,00000000212$$

- o Mengklasifikasikan dokumen uji untuk kelas Positif dan Negatif dari perhitungan probabilitas posterior $P(H/X)$ menggunakan rumus persamaan Naïve Bayes (2-8).

- Positif

$$P(Positif | X) = \frac{P(X | Positif) P(Positif)}{P(X)}$$

$$P(Positif | X) = \frac{0,0000000032 \times 0,4}{0,00000000212}$$

$$P(Positif | X) = 0,604$$

- Negatif

$$P(Negatif | X) = \frac{P(X | Negatif) P(Negatif)}{P(X)}$$

$$P(Negatif | X) = \frac{0,0000000014 \times 0,6}{0,00000000212}$$

$$P(Negatif | X) = 0,396$$

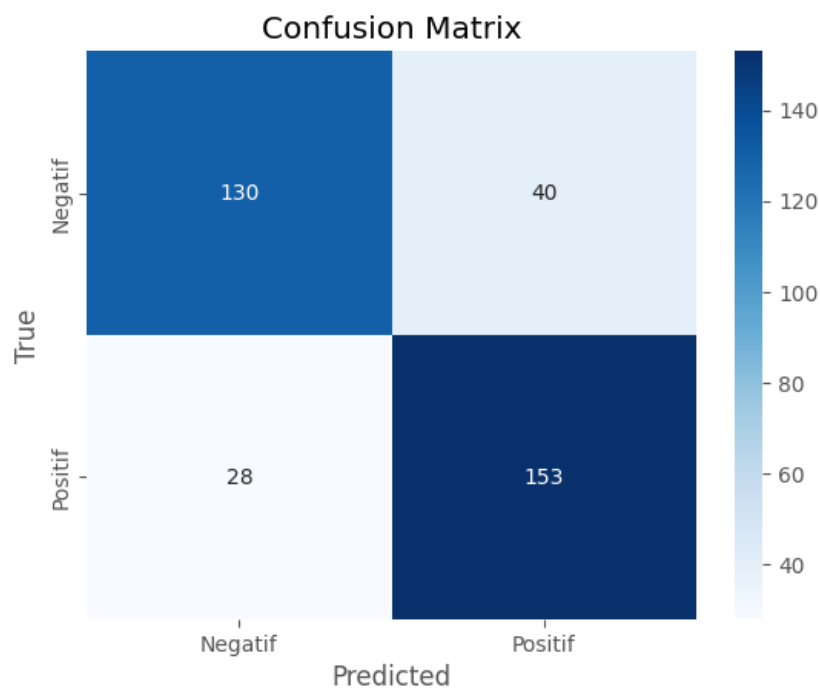
Dari perhitungan berdasarkan nilai probabilitas posterior diatas, maka diperoleh hasil 0,604 untuk kelas Positif dan 0,396 untuk kelas Negatif. Karena nilai probabilitas posterior pada kelas positif memiliki nilai lebih besar dibandingkan dengan nilai probabilitas posterior kelas negatif. Maka, data uji ke-3 terklasifikasi kedalam kelas Positif.

4.3.5 Evaluasi Model

Pada tahap ini akan dilakukan evaluasi model yang tujuannya untuk mendapatkan bagaimana performa yang dihasilkan oleh model *machine learning* yang telah dibuat dalam melakukan analisis sentimen. Maka dari itu, untuk melakukan evaluasi model, akan dilihat nilai-nilai seperti akurasi, presisi, *recall*, dan *F1-Score*. Untuk mendapatkan nilai-nilai tersebut, maka dilakukan pengujian *confusion matrix* untuk evaluasi hasil dari prediksi yang dilakukan oleh model. Berikut adalah hasil *confusion matrix* dari model *Naïve Bayes* dengan ekstraksi fitur *Tf-Idf* dan tanpa ekstraksi fitur *Tf-Idf* serta membandingkan penggunaan 2 kelas (Positif dan Negatif) dengan 3 kelas (Positif, Netral dan Negatif).

4.3.5.1 Evaluasi Model Menggunakan 2 Kelas (Positif dan Negatif)

a) *Confusion Matrix* Model *Naïve Bayes* TF-IDF



Gambar 4.6. *Confusion Matrix* *Naïve Bayes* dengan *Tf-Idf* menggunakan 2 kelas

Berdasarkan gambar 4.6. menunjukkan bahwa *dataset* komentar mengenai opini publik terhadap kendaraan listrik dengan algoritma *naïve bayes classifier* menggunakan ekstraksi fitur *Tf-Idf*, didapatkan 153 data positif yang diprediksi benar oleh model kedalam sentimen positif (*True Positive*) dan 28 data positif yang diprediksi salah oleh model kedalam sentimen negatif (*False Negative*), Kemudian terdapat 130 data negatif yang diprediksi benar oleh model

kedalam sentimen negatif (*True Negative*) dan 40 data negatif yang diprediksi salah oleh model kedalam sentimen positif (*False Positive*). Maka klasifikasi data yang didapatkan berdasarkan *confusion matrix* diatas adalah sebagai berikut.

- *True Positive (TP)*
 $TP = 153 \text{ data}$
- *False Negative (FN)*
 $FN = 28 \text{ data}$
- *True Negative (TN)*
 $TP = 130 \text{ data}$
- *False Positive (FP)*
 $FP = 40 \text{ data}$

Selanjutnya, dengan mengacu pada data *confusion matrix* diatas maka dapat dilakukan perhitungan akurasi, presisi, *recall*, dan *f1-score* dengan langkah perhitungan sebagai berikut:

- Akurasi

Untuk menghitung akurasi digunakan persamaan (2-5) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan akurasi:

$$\begin{aligned}
 \text{Accuracy} &= \frac{\text{True positive} + \text{True Negative}}{\text{Total data test}} \times 100\% \\
 &= \frac{153 + 130}{351} \times 100\% \\
 &= 80,62\%
 \end{aligned}$$

- Sentimen Positif

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned}
 \text{Precision} &= \frac{TP}{FP + TP} \\
 &= \frac{153}{40 + 153} \\
 &= 0,79
 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\begin{aligned} \text{Recall} &= \frac{TP}{FN + TP} \\ &= \frac{153}{28 + 153} \\ &= 0,85 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned} F1 - \text{Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \times \frac{0.79 \times 0.85}{0.79 + 0.85} \\ &= 0,82 \end{aligned}$$

- Sentimen Negatif

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} \text{Precision} &= \frac{TN}{FN + TN} \\ &= \frac{130}{28 + 130} \\ &= 0,82 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\text{Recall} = \frac{TN}{FP + TN}$$

$$= \frac{130}{40 + 130}$$

$$= 0,76$$

- *F1-Score*

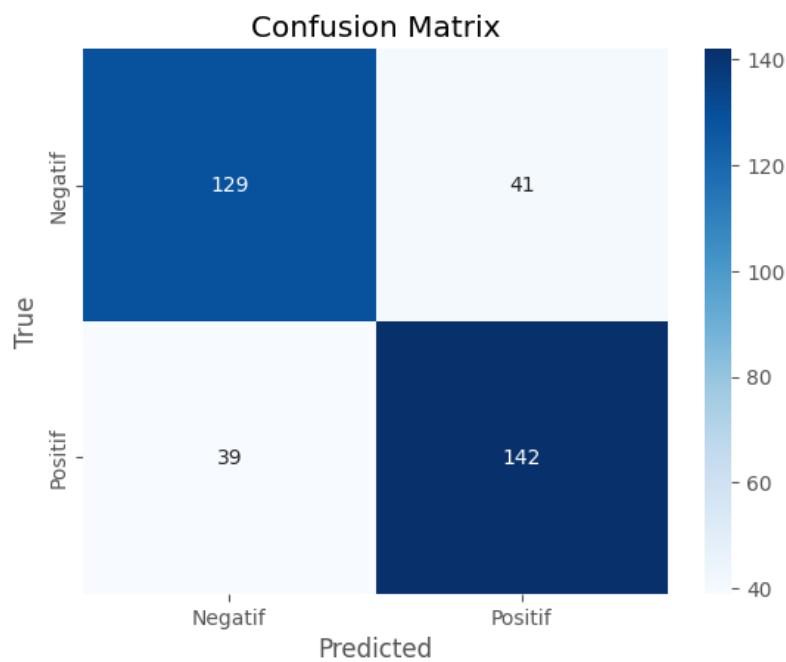
Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$= 2 \times \frac{0,82 \times 0,76}{0,82 + 0,76}$$

$$= 0,79$$

b) *Confusion Matrix Model Naïve Bayes tanpa Tf-Idf*



Gambar 4.7. *Confusion Matrix Naïve Bayes tanpa Tf-Idf* menggunakan 2 kelas

Berdasarkan gambar 4.7. menunjukkan bahwa dataset komentar mengenai opini publik terhadap kendaraan listrik dengan algoritma *naïve bayes classifier* tanpa ekstraksi fitur *Tf-Idf*, didapatkan 142 data positif yang diprediksi benar oleh model kedalam sentimen positif (*True Positive*) dan 39 data positif

yang diprediksi salah oleh model kedalam sentimen negatif (*False Negative*), Kemudian terdapat 129 data negatif yang diprediksi benar oleh model kedalam sentimen negatif (*True Negative*) dan 41 data negatif yang diprediksi salah oleh model kedalam sentimen positif (*False Positive*). Maka klasifikasi data yang didapatkan berdasarkan *confusion matrix* diatas adalah sebagai berikut.

- *True Positive (TP)*
 $TP = 142 \text{ data}$
- *False Negative (FN)*
 $FN = 39 \text{ data}$
- *True Negative (TN)*
 $TP = 129 \text{ data}$
- *False Positive (FP)*
 $FP = 41 \text{ data}$

Selanjutnya, dengan mengacu pada data *confusion matrix* diatas maka dapat dilakukan perhitungan akurasi, presisi, *recall*, dan *f1-score* dengan langkah perhitungan sebagai berikut:

- Akurasi

Untuk menghitung akurasi digunakan persamaan (2-5) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan akurasi:

$$\begin{aligned}
 \text{Accuracy} &= \frac{\text{True positive} + \text{True Negative}}{\text{Total data test}} \times 100\% \\
 &= \frac{142 + 129}{351} \times 100\% \\
 &= 77,20\%
 \end{aligned}$$

- Sentimen Positif

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned}
 \text{Precision} &= \frac{TP}{FP + TP} \\
 &= \frac{142}{41 + 142} \\
 &= 0,78
 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\begin{aligned} \text{Recall} &= \frac{TP}{FN + TP} \\ &= \frac{142}{39 + 142} \\ &= 0,78 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned} F1 - \text{Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \times \frac{0,78 \times 0,78}{0,78 + 0,78} \\ &= 0,78 \end{aligned}$$

- Sentimen Negatif

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} \text{Precision} &= \frac{TN}{FN + TN} \\ &= \frac{129}{39 + 129} \\ &= 0,77 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\text{Recall} = \frac{TN}{FP + TN}$$

$$= \frac{129}{41 + 129}$$

$$= 0,76$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$= 2 \times \frac{0,77 \times 0,76}{0,77 + 0,76}$$

$$= 0,76$$

Berdasarkan hasil perhitungan dua bentuk evaluasi model diatas dengan klasifikasi 2 kelas sentimen (positif dan negatif), maka didapatkan hasil evaluasi model seperti pada tabel 4.23 berikut.

Tabel 4.23. Perbandingan model *Naïve Bayes* menggunakan fitur *Tf-Idf* dan tanpa fitur *Tf-idf* dengan 2 kelas (Positif dan Negatif)

Model	Akurasi	Presisi		Recall		F1-Score	
		Positif	Negatif	Positif	Negatif	Positif	Negatif
NBC TF-IDF	80,62%	0,79	0,82	0,85	0,76	0,82	0,79
NBC	77,20%	0,78	0,77	0,78	0,76	0,78	0,76

Berdasarkan tabel 4.23 diatas menunjukkan hasil perbandingan evaluasi dua model klasifikasi analisis sentimen menggunakan *Naïve Bayes Classifier* dengan ekstraksi fitur TF-IDF (NBC TF-IDF) dan model analisis sentimen dengan *Naïve Bayes Classifier* tanpa menggunakan ekstraksi fitur TF-IDF (NBC). Dalam penerapannya, digunakan sejumlah metrik evaluasi standar seperti akurasi, presisi, *recall*, dan *F1-Score*. Metrik-metrik ini memberikan wawasan yang lebih dalam tentang kemampuan model dalam mengklasifikasikan data dengan benar.

Model NBC TF-IDF memiliki kinerja yang lebih tinggi dengan persentase akurasi sebesar 80,62%, sedangkan model NBC tanpa TF-IDF mencapai akurasi sekitar 77,20%. Ini menunjukkan bahwa model NBC TF-IDF berhasil

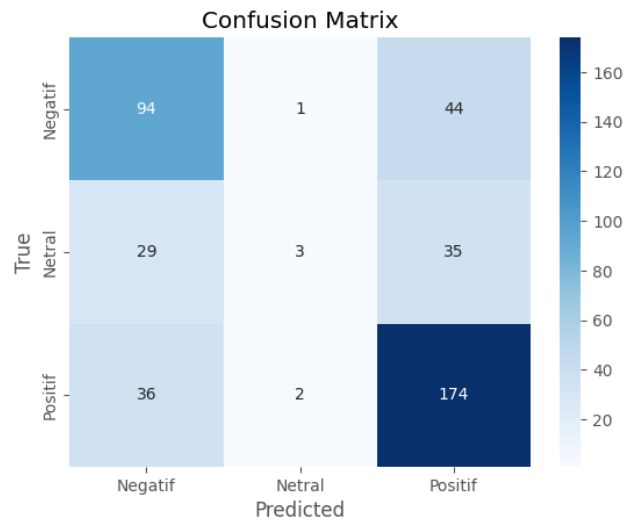
mengklasifikasikan lebih banyak data dengan benar dari keseluruhan data uji. Selanjutnya, kita memeriksa nilai-nilai presisi untuk mengukur proporsi prediksi positif yang benar dan *recall* untuk mengukur proporsi instance positif yang diidentifikasi dengan benar oleh model. Model NBC TF-IDF memiliki presisi sebesar 0,79 dan *recall* sebesar 0,85 untuk kelas positif, serta presisi sebesar 0,82 dan *recall* sebesar 0,76 untuk kelas negatif. Sedangkan, model NBC tanpa TF-IDF memiliki presisi sekitar 0,78 dan *recall* sekitar 0,78 untuk kelas positif, serta presisi sekitar 0,77 dan *recall* sekitar 0,76 untuk kelas negatif. Sehingga, dari hasil tersebut menunjukkan bahwa model NBC TF-IDF memiliki presisi dan *recall* yang lebih tinggi untuk kedua kelas, hal ini menunjukkan bahwa model ini lebih baik dalam mengidentifikasi dan mengklasifikasikan instance positif dan negatif.

Sealin itu, terdapat pula nilai *F1-Score* yang menggabungkan presisi dan *recall*, juga menunjukkan superioritas model NBC TF-IDF. *F1-Score* untuk kelas positif pada model NBC TF-IDF adalah 0,82, sementara pada model NBC tanpa TF-IDF hanya 0,78. Untuk kelas negatif, model NBC TF-IDF memiliki *F1-Score* sekitar 0,79, sementara model NBC tanpa TF-IDF hanya mencapai 0,76.

Oleh karena itu, secara keseluruhan dapat dilihat bahwa analisis sentimen menggunakan *Naïve Bayes Classifier* dengan ekstraksi fitur TF-IDF (NBC TF-IDF) memiliki kinerja yang lebih unggul daripada model analisis sentimen menggunakan *Naïve Bayes Classifier* tanpa ekstraksi fitur TF-IDF (NBC). Model NBC TF-IDF menunjukkan akurasi yang lebih tinggi dalam mengklasifikasikan data dengan 2 kelas klasifikasi sentimen yaitu positif dan negatif serta memiliki nilai presisi dan *recall* yang lebih baik untuk kedua kelas baik itu positif maupun negatif. Nilai *F1-Score* juga lebih tinggi pada model NBC TF-IDF yang menunjukkan kemampuannya dalam menjaga keseimbangan antara presisi dan *recall*.

4.3.5.2 Evaluasi Model Menggunakan 3 Kelas (Positif, Negatif dan Netral)

a) *Confusion Matrix* Model *Naïve Bayes* TF-IDF



Gambar 4.8. *Confusion Matrix* *Naïve Bayes* dengan *Tf-Idf* menggunakan 3 kelas

Berdasarkan gambar 4.8 menunjukkan bahwa *dataset* komentar mengenai opini publik terhadap kendaraan listrik dengan algoritma *naïve bayes classifier* menggunakan ekstraksi fitur *Tf-Idf* dengan menggunakan 3 kelas klasifikasi yaitu positif, netral dan negatif. Berdasarkan grafik *confusion matrix* diatas didapatkan 174 data positif, 3 data netral, 94 data negatif yang di prediksi dengan benar oleh model (*True Positive* dan *True Negative*). Kemudian terdapat data yang di prediksi salah oleh model (*False Positive* dan *False Negative*) yaitu 36 data kedalam kelas negatif yang seharusnya positif, 29 data kedalam kelas negatif yang seharusnya netral, 44 data kedalam kelas positif yang seharusnya negatif, 35 data kedalam kelas positif yang seharusnya netral, 2 data kedalam kelas netral yang seharusnya positif, 1 data netral yang seharusnya negatif. Untuk mendapatkan perhitungan nilai *True Positive*, *False Positive*, *True Negative* dan *False Negative* dapat dilakukan sebagai berikut dengan mengacu pada gambar 4.10.

1	2	3
4	5	6
7	8	9

Gambar 4.9. Ilustrasi penomoran petak *confusion matrix multiclass* dengan *Tf-Idf*

Selanjutnya, dengan mengacu pada data *confusion matrix* diatas maka dapat dilakukan perhitungan akurasi, presisi, *recall*, dan *f1-score* dengan langkah perhitungan sebagai berikut:

- Akurasi

Untuk menghitung akurasi digunakan persamaan (2-5) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan akurasi:

$$\begin{aligned}
 \text{Accuracy} &= \frac{\text{True positive} + \text{True Negative} + \text{True Ntral}}{\text{Total data test}} \times 100\% \\
 &= \frac{174 + 94 + 3}{418} \times 100\% \\
 &= 64,83\%
 \end{aligned}$$

- Sentimen Positif

- Data klasifikasi *confusion matrix*

- *True Positive (TP)*

$$TP = \text{petak 9} = 174 \text{ data}$$

- *False Positive (FP)*

$$FP = \text{petak 3} + \text{petak 6} = 44 + 35 = 79 \text{ data}$$

- *True Negative (TN)*

$$TN = \text{petak 1} + \text{petak 2} + \text{petak 4} + \text{petak 5}$$

$$TN = 94 + 1 + 29 + 3 = 127 \text{ data}$$

- *False Negative (FN)*

$$FN = \text{petak 7} + \text{petak 8} = 36 + 2 = 38 \text{ data}$$

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} Precision &= \frac{TP}{FP + TP} \\ &= \frac{174}{79 + 174} \\ &= 0,69 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\begin{aligned} Recall &= \frac{TP}{FN + TP} \\ &= \frac{174}{38 + 174} \\ &= 0,82 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned} F1 - Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\ &= 2 \times \frac{0,69 \times 0,82}{0,69 + 0,82} \\ &= 0,75 \end{aligned}$$

- Sentimen Netral

- Data klasifikasi *confusion matrix*

- *True Positive (TP)*

$$TP = \text{petak 5} = 3 \text{ data}$$

- *False Positive (FP)*

$$FP = \text{petak 2} + \text{petak 8} = 1 + 2 = 3 \text{ data}$$

- *True Negative (TN)*

$$TN = \text{petak 1} + \text{petak 3} + \text{petak 7} + \text{petak 9}$$

$$TN = 94 + 44 + 36 + 174 = 348 \text{ data}$$

- *False Negative (FN)*

$$FN = \text{petak 4} + \text{petak 6} = 29 + 35 = 64 \text{ data}$$

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} \text{Precision} &= \frac{TP}{FP + TP} \\ &= \frac{3}{3 + 3} \\ &= 0,50 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\begin{aligned} \text{Recall} &= \frac{TP}{FN + TP} \\ &= \frac{3}{64 + 3} \\ &= 0,04 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned} F1 - \text{Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \times \frac{0,50 \times 0,04}{0,50 + 0,04} \\ &= 0,08 \end{aligned}$$

- Sentimen Negatif

- Data klasifikasi *confusion matrix*

- *True Positive (TP)*

$$TP = \text{petak 1} = 94 \text{ data}$$

- *False Positive (FP)*

$$FP = \text{petak 4} + \text{petak 7} = 29 + 36 = 65 \text{ data}$$

- *True Negative (TN)*

$$TN = \text{petak 5} + \text{petak 6} + \text{petak 8} + \text{petak 9}$$

$$TN = 3 + 35 + 2 + 174 = 214 \text{ data}$$

- *False Negative (FN)*

$$FN = \text{petak 2} + \text{petak 3} = 1 + 44 = 45 \text{ data}$$

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} \text{Precision} &= \frac{TP}{FP + TP} \\ &= \frac{94}{65 + 94} \\ &= 0,59 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

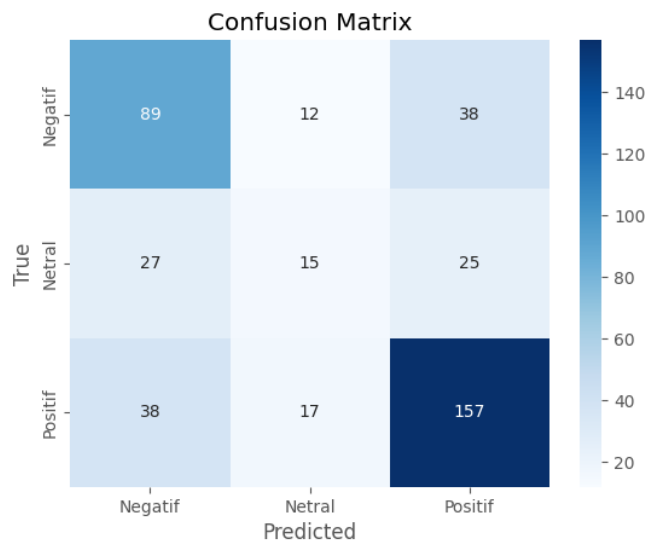
$$\begin{aligned} \text{Recall} &= \frac{TP}{FN + TP} \\ &= \frac{94}{45 + 94} \\ &= 0,68 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned}
 F1 - Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\
 &= 2 \times \frac{0,59 \times 0,68}{0,59 + 0,68} \\
 &= 0,63
 \end{aligned}$$

b) *Confusion Matrix Model Naïve Bayes tanpa Tf-Idf*



Gambar 4.10. *Confusion Matrix Naïve Bayes tanpa Tf-Idf menggunakan 3 kelas*

Berdasarkan gambar 4.10 menunjukkan bahwa *dataset* komentar mengenai opini publik terhadap kendaraan listrik dengan algoritma *naïve bayes classifier* menggunakan ekstraksi fitur *Tf-Idf* dengan menggunakan 3 kelas klasifikasi yaitu positif, netral dan negatif. Berdasarkan grafik *confusion matrix* diatas didapatkan 157 data positif, 15 data netral, 89 data negatif yang di prediksi dengan benar oleh model (*True Positive* dan *True Negative*). Kemudian terdapat data yang di prediksi salah oleh model (*False Positive* dan *False Negative*) yaitu 38 data kedalam kelas negatif yang seharusnya positif, 27 data kedalam kelas negatif yang seharusnya netral, 38 data kedalam kelas positif yang seharusnya negatif, 25 data kedalam kelas positif yang seharusnya netral, 17 data kedalam kelas netral yang seharusnya positif, 12 data netral yang seharusnya negatif. Untuk mendapatkan perhitungan nilai *True Positive*, *False Positive*, *True Negative* dan *False Negative* dapat dilakukan sebagai berikut dengan mengacu pada gambar 4.12.

1	2	3
4	5	6
7	8	9

Gambar 4.11. Ilustrasi penomoran petak *confusion matrix multiclass* tanpa *Tf-Idf*

Selanjutnya, dengan mengacu pada data *confusion matrix* diatas maka dapat dilakukan perhitungan akurasi, presisi, *recall*, dan *f1-score* dengan langkah perhitungan sebagai berikut:

- Akurasi

Untuk menghitung akurasi digunakan persamaan (2-5) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan akurasi:

$$\begin{aligned}
 Accuracy &= \frac{True\ positive + True\ Negative + True\ Ntral}{Total\ data\ test} \times 100\% \\
 &= \frac{157 + 89 + 15}{418} \times 100\% \\
 &= 62,44\%
 \end{aligned}$$

- Sentimen Positif

- Data klasifikasi *confusion matrix*

- *True Positive (TP)*

$$TP = \text{petak 9} = 157 \text{ data}$$

- *False Positive (FP)*

$$FP = \text{petak 3} + \text{petak 6} = 38 + 25 = 63 \text{ data}$$

- *True Negative (TN)*

$$TN = \text{petak 1} + \text{petak 2} + \text{petak 4} + \text{petak 5}$$

$$TN = 89 + 12 + 27 + 15 = 143 \text{ data}$$

- *False Negative (FN)*

$$FN = \text{petak 7} + \text{petak 8} = 38 + 17 = 55 \text{ data}$$

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} Precision &= \frac{TP}{FP + TP} \\ &= \frac{157}{63 + 157} \\ &= 0,71 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\begin{aligned} Recall &= \frac{TP}{FN + TP} \\ &= \frac{157}{55 + 157} \\ &= 0,74 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned} F1 - Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\ &= 2 \times \frac{0.71 \times 0.74}{0.71 + 0.74} \\ &= 0,73 \end{aligned}$$

- Sentimen Netral

- Data klasifikasi *confusion matrix*

- *True Positive (TP)*

$$TP = \text{petak 5} = 15 \text{ data}$$

- *False Positive (FP)*

$$FP = \text{petak 2} + \text{petak 8} = 12 + 17 = 29 \text{ data}$$

- *True Negative (TN)*

$$TN = \text{petak 1} + \text{petak 3} + \text{petak 7} + \text{petak 9}$$

$$TN = 89 + 38 + 38 + 157 = 322 \text{ data}$$

- *False Negative (FN)*

$$FN = \text{petak 4} + \text{petak 6} = 27 + 25 = 52 \text{ data}$$

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} \text{Precision} &= \frac{TP}{FP + TP} \\ &= \frac{15}{19 + 15} \\ &= 0,34 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\begin{aligned} \text{Recall} &= \frac{TP}{FN + TP} \\ &= \frac{15}{52 + 15} \\ &= 0,22 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned} F1 - \text{Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \times \frac{0,34 \times 0,22}{0,34 + 0,22} \\ &= 0,27 \end{aligned}$$

- Sentimen Negatif

- Data klasifikasi *confusion matrix*

- *True Positive (TP)*

$$TP = \text{petak 1} = 89 \text{ data}$$

- *False Positive (FP)*

$$FP = \text{petak 4} + \text{petak 7} = 27 + 38 = 65 \text{ data}$$

- *True Negative (TN)*

$$TN = \text{petak 5} + \text{petak 6} + \text{petak 8} + \text{petak 9}$$

$$TN = 15 + 25 + 17 + 157 = 214 \text{ data}$$

- *False Negative (FN)*

$$FN = \text{petak 2} + \text{petak 3} = 12 + 38 = 50 \text{ data}$$

- *Precision*

Untuk menghitung *precision* digunakan persamaan (2-4) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *precision*:

$$\begin{aligned} \text{Precision} &= \frac{TP}{FP + TP} \\ &= \frac{89}{65 + 89} \\ &= 0,58 \end{aligned}$$

- *Recall*

Untuk menghitung *recall* digunakan persamaan (2-6) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *recall*:

$$\begin{aligned} \text{Recall} &= \frac{TP}{FN + TP} \\ &= \frac{89}{50 + 89} \\ &= 0,64 \end{aligned}$$

- *F1-Score*

Untuk menghitung *F1-Score* digunakan persamaan (2-7) seperti yang dijelaskan pada bab 2. Berikut adalah perhitungan *F1-Score*:

$$\begin{aligned}
 F1 - Score &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \\
 &= 2 \times \frac{0,58 \times 0,64}{0,58 + 0,64} \\
 &= 0,61
 \end{aligned}$$

Berdasarkan hasil perhitungan dua bentuk evaluasi model diatas dengan klasifikasi 3 kelas sentimen (positif, netral dan negatif), maka didapatkan hasil evaluasi model seperti pada tabel 4.15 berikut.

Tabel 4.24. Perbandingan model *Naïve Bayes* menggunakan fitur *Tf-Idf* dan tanpa fitur *Tf-idf* dengan 3 kelas (Positif, Netral dan Negatif)

Model	Akurasi	Presisi			Recall			F1-Score		
		Positif	Negatif	Netral	Positif	Negatif	Netral	Positif	Negatif	Netral
NBC TF-IDF	64,83%	0,69	0,59	0,50	0,82	0,68	0,04	0,75	0,63	0,08
NBC	62,44%	0,71	0,58	0,34	0,74	0,64	0,22	0,73	0,62	0,27

Berdasarkan tabel 4.24 diatas menunjukkan hasil perbandingan evaluasi dua model klasifikasi analisis sentimen menggunakan *Naïve Bayes Classifier* dengan ekstraksi fitur TF-IDF (NBC TF-IDF) dan model analisis sentimen dengan *Naïve Bayes Classifier* tanpa menggunakan ekstraksi fitur TF-IDF (NBC). Dalam penerapannya, digunakan sejumlah metrik evaluasi standar seperti akurasi, presisi, *recall*, dan *F1-Score*. Metrik-metrik ini memberikan wawasan yang lebih dalam tentang kemampuan model dalam mengklasifikasikan data dengan benar.

Model NBC TF-IDF memiliki kinerja yang lebih tinggi dengan persentase akurasi sebesar 64,83%, sedangkan model NBC tanpa TF-IDF mencapai akurasi sekitar 62,44%. Ini menunjukkan bahwa model NBC TF-IDF berhasil mengklasifikasikan lebih banyak data dengan benar dari keseluruhan data uji. Selanjutnya, kita memeriksa nilai-nilai presisi untuk mengukur proporsi prediksi positif yang benar dan *recall* untuk mengukur proporsi instance positif yang diidentifikasi dengan benar oleh model. Model NBC TF-IDF memiliki presisi sebesar 0,69 dan *recall* sebesar 0,82 untuk kelas positif, presisi sebesar 0,50 dan *recall* sebesar 0,04 untuk kelas netral, serta presisi sebesar 0,59 dan *recall* sebesar 0,68 untuk kelas negatif. Sedangkan, model NBC tanpa TF-IDF memiliki presisi sekitar 0,71 dan *recall* sekitar 0,74 untuk kelas positif, presisi sebesar 0,34 dan *recall* sebesar 0,22 untuk kelas netral, serta presisi sekitar 0,58 dan *recall* sekitar 0,64 untuk kelas negatif.

Sehingga, dari hasil tersebut menunjukkan bahwa model NBC TF-IDF memiliki rata-rata presisi dan *recall* yang lebih tinggi untuk ketiga kelas sentimen, hal ini menunjukkan bahwa model ini lebih baik dalam mengidentifikasi dan mengklasifikasikan instance positif dan negatif.

Sealin itu, terdapat pula nilai *F1-Score* yang menggabungkan presisi dan *recall*, juga menunjukkan superioritas model NBC TF-IDF. *F1-Score* untuk kelas positif pada model NBC TF-IDF adalah 0,75, sementara pada model NBC tanpa TF-IDF hanya 0,73. Untuk kelas netral, model NBC TF-IDF memiliki *F1-Score* sekitar 0,08, sementara model NBC tanpa TF-IDF hanya mencapai 0,27. Kemudian untuk kelas negatif, model NBC TF-IDF memiliki *F1-Score* sekitar 0,63, sementara model NBC tanpa TF-IDF hanya mencapai 0,61

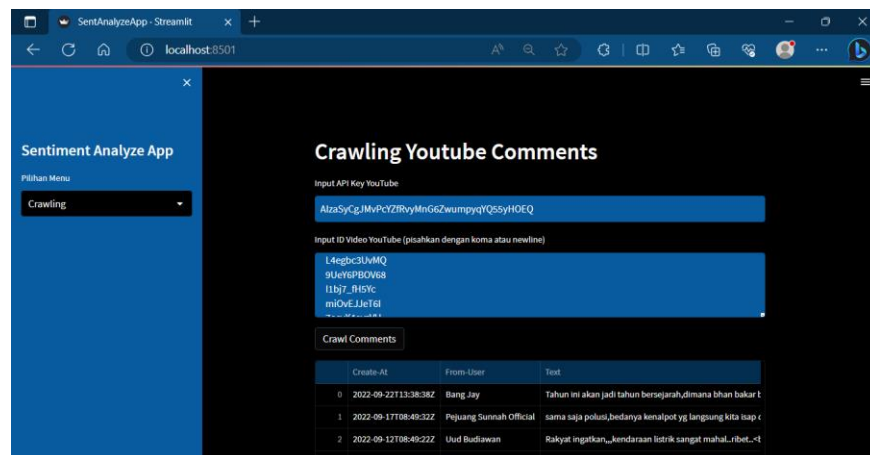
Oleh karena itu, secara keseluruhan dapat dilihat bahwa analisis sentimen menggunakan *Naïve Bayes Classifier* dengan ekstraksi fitur TF-IDF (NBC TF-IDF) memiliki kinerja yang lebih unggul daripada model analisis sentimen menggunakan *Naïve Bayes Classifier* tanpa ekstraksi fitur TF-IDF (NBC). Model NBC TF-IDF menunjukkan akurasi yang lebih tinggi dalam mengklasifikasikan data dengan 3 kelas klasifikasi sentimen yaitu positif, netral dan negatif. Namun, pengklasifikasian dengan menggunakan 3 kelas (positif, netral dan negatif) dalam penelitian ini menunjukkan bahwa performa yang dihasilkan model tidak jauh lebih baik dibandingkan dengan hanya menggunakan 2 kelas (positif dan negatif) saja. Hal ini dikarenakan keseimbangan jumlah data, yang dimana data dengan penambahan kelas netral memiliki jumlah data yang kurang banyak sehingga model lebih cenderung terfokus pada data yang lebih besar yaitu data positif dan negatif yang memungkinkan model menjadi terlalu spesifik terhadap kelas mayoritas (positif atau negatif) dan tidak mampu menggeneralisasi dengan baik untuk kelas netral yang kurang banyak.

4.4 Perancangan Aplikasi Sentimen Analisis Berbasis Web

Setelah melalui tahapan perancangan model analisis opini publik terhadap kendaraan listrik menggunakan algoritma *Naïve Bayes Classifier*. Selanjutnya akan dilakukan implementasi model klasifikasi *machine learning* dari model yang telah dibuat kedalam sebuah sistem berbasis *website*. Pada *website* sentimen analisis yang dibuat nantinya dapat mengklasifikasikan *dataset* komentar untuk melihat sentimen masyarakat terhadap kendaraan listrik dengan algoritma *Naïve Bayes Classifier*, yang

dimana terdapat beberapa fitur yang disajikan seperti *Crawling*, *Preprocessing* dan *Classification*. *Website* ini dibuat menggunakan *framework streamlit* yang dimana model analisis yang telah dibuat akan di implementasi dan di integrasikan kedalam *framework streamlit*. Berikut penyajian hasil dan penjelasan mengenai perancangan aplikasi sentimen analisis berbasis *website* dengan beberapa fitur yang disajikan.

4.4.1 Fitur *Crawling*



Gambar 4.12. Tampilan *website* sentimen analisis pada fitur *Crawling*

Berdasarkan gambar 4.12 merupakan tampilan pada fitur *Crawling* yang dimana pada fitur ini sistem akan melakukan pengambilan data teks komentar pada video *youtube* dengan menginputkan *API Key youtube* yang telah dibuat melalui *google cloud* dan ID Video *youtube* yang ingin diambil komentarnya. Pada gambar diatas juga dapat dilihat dalam menginput ID Video *youtube* dapat dilakukan dengan beberapa ID Video bukan hanya satu ID saja dengan pemisah dengan koma atau *newline*. Kemudian, setelah *API Key* dan ID Video *youtube* diinputkan sistem akan mulai menampilkan hasil *crawling* data teks komentar setelah menekan tombol *Crawl Comments* sehingga menghasilkan hasil data *crawling* seperti pada gambar 4.13 berikut.

	Create-At	From-User	Text
0	2022-09-22T13:38:38Z	Bang Jay	Tahun ini akan jadi tahun bersejarah, dimana bahan bakar t
1	2022-09-17T08:49:32Z	Pejuang Sunnah Official	sama saja polusi, bedanya kentalipol yg langsung kita isap c
2	2022-09-12T08:49:22Z	Uud Budawan	Rakyat ingatkan, kendaraan listrik sangat mahal, ribet, <4
3	2022-09-08T14:54:37Z	Dudi Dudi	Yg demo itu buta n tult...bukannya cari solusi malah mengo
4	2022-09-04T18:46:21Z	Risky	"gak tau kapade"
5	2022-09-03T04:03:38Z	Pro Max	Stop penjualan kendaraan bahan bakar fosil di seluruh ind
6	2022-09-01T21:28:55Z	Elya Elroy	Hitung2annya tergantung juga dengan TDL. Kalau TDL nai
7	2022-09-01T12:09:10Z	Fakru Uyee	Kabar gembira bagi pelaku Renewable Jadi bisa Bf
8	2022-09-01T11:18:18Z	BAGAJUL	Ini berita yg sangat menarik dan souf! juga Pointnya
9	2022-09-01T10:07:53Z	sarana dakwah islam offi	Nanti udah pada pindah listrik listriknya naik,

Gambar 4.13. Tampilan hasil *crawling* data teks komentar

Pada gambar 4.13 diatas dapat dilihat hasil data teks komentar setelah dilakukan *crawling* dengan *API Key* dan ID Video *youtube* yang diinputkan akan menghasilkan *dataFrame* dengan variable yang diambil adalah tanggal serta waktu komentar (*Create-At*), *From-User* dan teks komentar (*Text*). Setelah data teks komentar berhasil ditampilkan pada sistem, makan selanjutnya data hasil *crawling* dapat di *download* menjadi *dataset* dalam *format excel* dengan menekan link *Download Excel*.

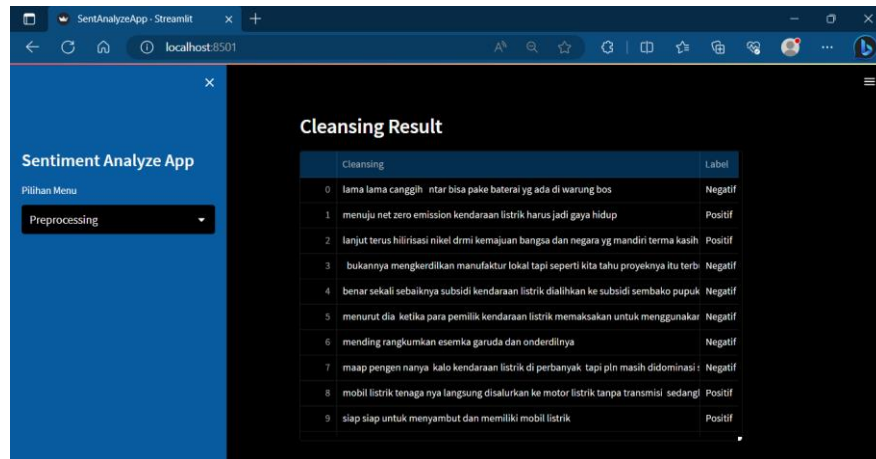
4.4.2 Fitur Preprocessing

	From-User	Text
0	Suherman Bachtar	Kalau subsidi itu diberikan kepada Penguasa bukan kepada Rakyat sebagai Pembeli
1	Alfitri Afrizal	Feisal baeri hanya pandai kritik, pengamat dodol selalu berfikir negatif pantas aja ind
2	Marupe Bangmas	MAJU TERUS DENGAN SENDIRINYA SEMUA TERBENTUK
3	Heri Sabrianto	Seharusnya pemerintah membangun infrastrukturnya dulu Kesiapan stasiun per
4	Alfitri Afrizal	Kapan Indonesia bleh maju kalo seperti ide anda...apa yg di lakukan pemerintah saat

Gambar 4.14. Tampilan *website* sentimen analisis pada fitur *Preprocessing*

Berdasarkan gambar 4.14 merupakan tampilan pada fitur *Preprocessing* yang dimana pada fitur ini sistem akan melakukan tahap *preprocessing* untuk membersihkan data sehingga dapat dilakukan pengklasifikasian pada tahap selanjutnya. Pada gambar diatas dapat dilihat sistem akan memproses dataset yang

diinputkan melalui sistem dengan format *excel*, lalu sistem akan melakukan tahapan *preprocessing* seperti *cleansing*, normalisasi, *tokenize*, *stopwords* dan *stemming*.

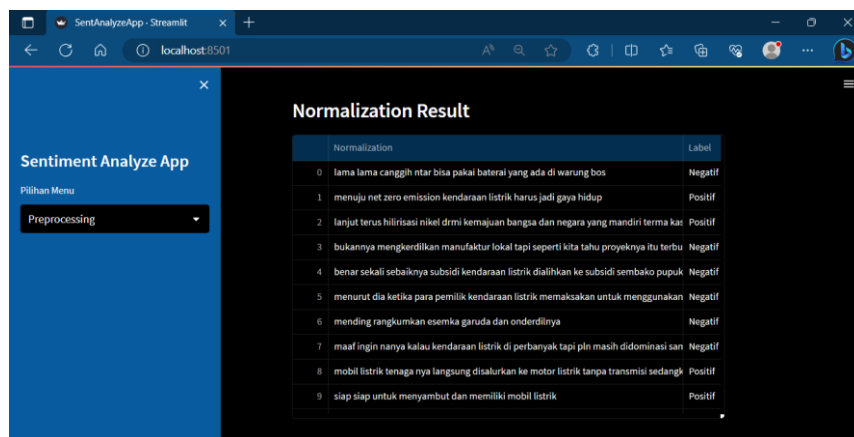


The screenshot shows a web application titled "Sentiment Analyze App" with a sidebar menu containing "Pilihan Menu" and "Preprocessing". The main content area displays a table titled "Cleansing Result". The table has two columns: "Cleansing" and "Label". It contains 10 rows of data, each with a number, a sentence, and a sentiment label.

	Cleansing	Label
0	lama lama canggih ntar bisa pake baterai yg ada di warung bos	Negatif
1	menuju net zero emission kendaraan listrik harus jadi gaya hidup	Positif
2	lanjut terus hilirisasi nikel drmi kemajuan bangsa dan negara yg mandiri terma kasih	Positif
3	bukannya mengkerdikan manufaktur lokal tapi seperti kita tahu proyeknya itu terbu	Negatif
4	benar sekali sebaiknya subsidi kendaraan listrik dialihkan ke subsidi sembako pupuk	Negatif
5	menurut dia ketika para pemilik kendaraan listrik memaksakan untuk menggunakan	Negatif
6	mending rangkumkan esemka garuda dan onderdilnya	Negatif
7	maap pengen nanya kalo kendaraan listrik di perbanyak tapi pln masih didominasi i	Negatif
8	mobil listrik tenaga nya langsung disalurkan ke motor listrik tanpa transmisi sedangl	Positif
9	siap siap untuk menyambut dan memiliki mobil listrik	Positif

Gambar 4.15. Tampilan hasil proses *cleansing* pada fitur *Preprocessing*

Pada gambar 4.15 dapat dilihat bahwa sistem pada fitur *Preprocessing* melakukan proses *cleansing* yang dimana tujuannya untuk membersihkan data teks komentar dari hal yang tidak diperlukan yang membuat data menjadi tidak efektif seperti *hashtag*, *mention username*, *URL*, tanda baca, *double space* (spasi berlebih), *emoticon*, bilangan angka, simbol dan *case folding* atau mengubah teks menjadi huruf kecil (*lower case*).



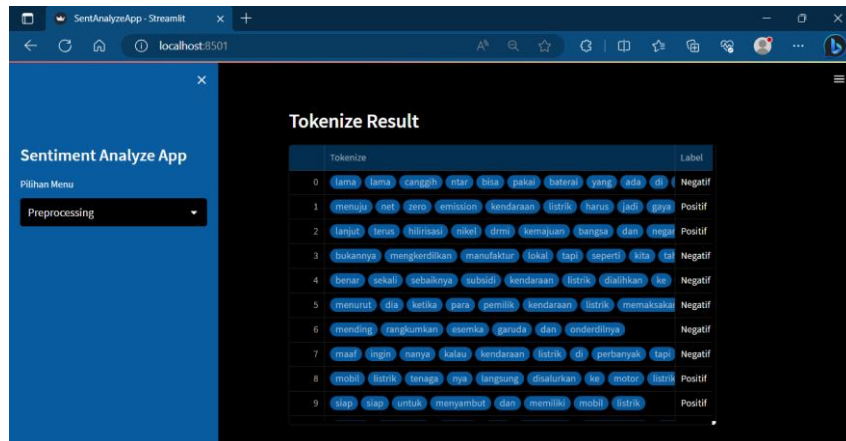
The screenshot shows the same web application as before, but the main content area displays a table titled "Normalization Result". The table has two columns: "Normalization" and "Label". It contains 10 rows of data, each with a number, a sentence, and a sentiment label. The sentences in this table are the result of the normalization process applied to the original sentences.

	Normalization	Label
0	lama lama canggih ntar bisa pakai baterai yang ada di warung bos	Negatif
1	menuju net zero emission kendaraan listrik harus jadi gaya hidup	Positif
2	lanjut terus hilirisasi nikel drmi kemajuan bangsa dan negara yang mandiri terma kas	Positif
3	bukannya mengkerdikan manufaktur lokal tapi seperti kita tahu proyeknya itu terbu	Negatif
4	benar sekali sebaiknya subsidi kendaraan listrik dialihkan ke subsidi sembako pupuk	Negatif
5	menurut dia ketika para pemilik kendaraan listrik memaksakan untuk menggunakan	Negatif
6	mending rangkumkan esemka garuda dan onderdilnya	Negatif
7	maaf ingin nanya kalau kendaraan listrik di perbanyak tapi pln masih didominasi san	Negatif
8	mobil listrik tenaga nya langsung disalurkan ke motor listrik tanpa transmisi sedang	Positif
9	siap siap untuk menyambut dan memiliki mobil listrik	Positif

Gambar 4.16. Tampilan hasil proses normalisasi pada fitur *Preprocessing*

Kemudian, setelah dilakukan proses *cleansing* maka dilanjutkan dengan peroses normalisasi seperti gambar 4.16 daiatas. Sistem akan melakukan tahap normalisasi dengan tujuan untuk mengubah kata-kata singkatan atau kata yang tidak

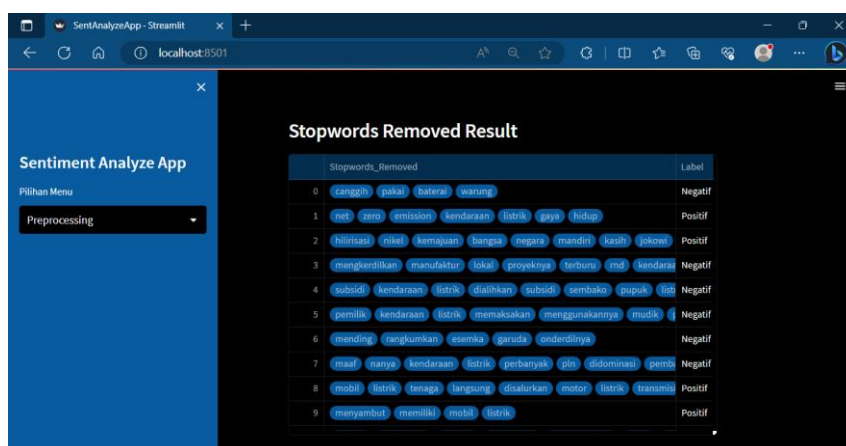
baku (*slang words*) dalam dataset teks komentar yang diinputkan menjadi bahasa yang baku sesuai Kamus Besar Bahasa Indonesia (KBBI).



	Tokenize	Label
0	lana lama cangah ntar bisa pakai batara yang ada	Negatif
1	menuju net zero emission kendaraan listrik harus jadi gaya	Positif
2	lanjut terus hilirasi nikel dms kemajuan bangsa dan nega	Positif
3	bukannya mengendikan manufaktur lokal tapi seperti kita ta	Negatif
4	benar sekali sebaiknya subsidi kendaraan listrik dialihkan ke	Negatif
5	memurut dia ketika para pemilik kendaraan listrik memaksaka	Negatif
6	mending rangkumkas esemka garuda dan onderdinya	Negatif
7	maaf dinya nanya kalap kendaraan listrik di perbanyak lagi	Negatif
8	mobli listrik tenaga nya langsung disalurkan ke motor listrik	Positif
9	slap slap untuk menyambung dan memuka mobil listrik	Positif

Gambar 4.17. Tampilan hasil proses *tokenize* pada fitur *Preprocessing*

Setelah itu, pada gambar 4.17 menunjukkan proses *tokenize* setelah dilakukan tahap normalisasi kata pada dataset. Pada tahap *tokenize* dilakukan untuk memecah teks atau kalimat menjadi unit-unit yang lebih kecil, yang disebut token dalam memudahkan pengubahan teks menjadi lebih terstruktur sehingga dapat diolah pada tahap selanjutnya.



	Stopwords_Removed	Label
0	cangah pakai batara warung	Negatif
1	net zero emission kendaraan listrik gaya hidup	Positif
2	hilirasi nikel kemajuan bangsa negara mandiri kasih jolow	Positif
3	mengendikan manufaktur lokal proyeknya terbung dmd kendara	Negatif
4	subsidi kendaraan listrik dialihkan subsidi sembako pupuk list	Negatif
5	pemilik kendaraan listrik memaksakan menggunakannya mudik	Negatif
6	mending rangkumkas esemka garuda onderdinya	Negatif
7	maaf dinya kendaraan listrik perbanyak spin didominasi pemli	Negatif
8	mobli listrik tenaga langsung disalurkan motor listrik tranomi	Positif
9	menyambung memuka mobil listrik	Positif

Gambar 4.18. Tampilan hasil proses *stopwords* pada fitur *Preprocessing*

Kemudian terdapat pula proses *stopwords* seperti gambar 4.18 diatas yang dilakukan untuk menghilangkan kata-kata yang tidak memiliki makna yang khusus dalam analisis teks dengan tujuan agar membuat data memnjadi lebih bersih dan

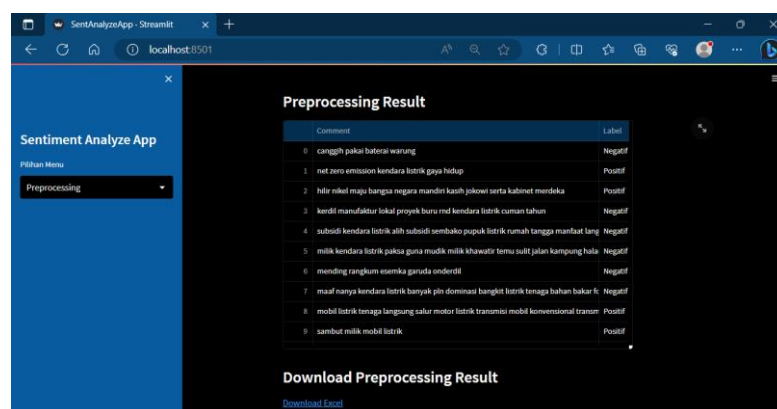
memfokuskan perhatian pada kata-kata yang lebih relevan dan informatif berdasarkan hasil *tokenize* sebelumnya pada *dataset*.



	Stemming	Label
0	canggih pakek baterai warung	Negatif
1	net zero emission kendaraan listrik gaya hidup	Positif
2	hilir nikel maju bangsa negara mandiri kaph Jokowi serta kabinet merdeka	Positif
3	kendri manufaktur lokal proyek buru rmd kendaraan listrik cuman tahun	Negatif
4	subsidi kendaraan listrik alih subsidi sembako pupuk listrik rt	Negatif
5	milik kendaraan listrik paksa guna mudik milik khawatir temu	Negatif
6	mending rangkum esemka garuda onderdri	Negatif
7	maaf hanya kendaraan listrik banyak pin dominasi bangkit listrik	Negatif
8	mobil listrik tenaga langsung salur motor listrik transmisi mobil konvensional transm	Positif
9	sambut milik mobil listrik	Positif

Gambar 4.19. Tampilan hasil proses *stemming* pada fitur *Preprocessing*

Selanjutnya terdapat proses *stemming* seperti pada gambar 4.19 diatas yang bertujuan untuk mengubah sebuah kata ke dalam bentuk kata dasarnya dengan menghapus kata imbuhan di depan maupun imbuhan di belakang kata atau mereduksi kata-kata dalam teks menjadi bentuk dasar, sehingga kata-kata yang memiliki arti yang sama atau terkait akan dikelompokkan menjadi satu bentuk berdasarkan data yang telah melalui tahap *stopwords* yang masih dalam bentuk token.



	Comment	Label
0	canggih pakek baterai warung	Negatif
1	net zero emission kendaraan listrik gaya hidup	Positif
2	hilir nikel maju bangsa negara mandiri kaph Jokowi serta kabinet merdeka	Positif
3	kendri manufaktur lokal proyek buru rmd kendaraan listrik cuman tahun	Negatif
4	subsidi kendaraan listrik alih subsidi sembako pupuk listrik rumah tangga manfaat tang	Negatif
5	milik kendaraan listrik paksa guna mudik milik khawatir temu sulit jalan kampung hulu	Negatif
6	mending rangkum esemka garuda onderdri	Negatif
7	maaf hanya kendaraan listrik banyak pin dominasi bangkit listrik tenaga bahan bakar fi	Negatif
8	mobil listrik tenaga langsung salur motor listrik transmisi mobil konvensional transm	Positif
9	sambut milik mobil listrik	Positif

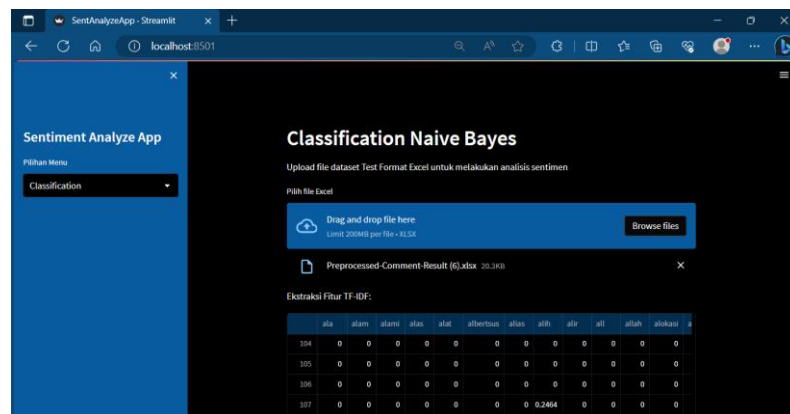
Download Preprocessing Result
[Download Excel](#)

Gambar 4.20. Tampilan hasil *dataset* setelah *preprocessing* pada fitur *Preprocessing*

Kemudian, setelah dilakukan beberapa tahapan *preprocessing* seperti *cleansing*, normalisasi, *tokenize*, *stopwords* dan *stemming*. Selanjutnya adalah sistem pada fitur *preprocessing* akan menampilkan hasil akhir *dataset* yang sudah dibersihkan seperti gambar 4.20 diatas yang dimana dapat dilihat data yang

sebelumnya terpisah menjadi beberapa token fitur kata telah di satukan kembali sehingga membentuk suatu kalimat yang utuh sebagai hasil *preprocessing* yang telah dilakukan. Berdasarkan gambar diatas juga dapat dilihat bahwa pada fitur *Preprocessing* hasil dari dataset yang sudah melewati tahap *preprocessing* dapat di *download* dalam format *excel* dengan menekan *link Download Excel* untuk digunakan sebagai *dataset* yang akan di klasifikasi dalam analisis sentimen.

4.4.3 Fitur Classification



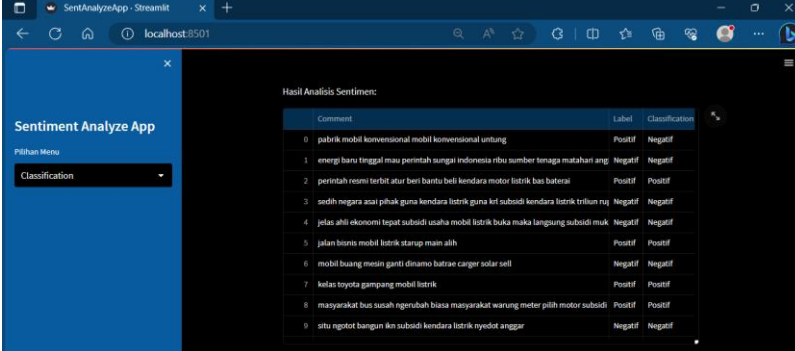
Gambar 4.21. Tampilan *website* sentimen analisis pada fitur *Classification*

Berdasarkan gambar 4.21 merupakan tampilan pada fitur *Classification* yang dimana pada fitur ini sistem akan melakukan klasifikasi dataset uji dengan format *excel* yang diinputkan untuk dilakukan analisis sentimen opini publik terhadap kendaraan listrik menggunakan algoritma *Naïve Bayes Classifier*. Kemudian pada fitur *Classification* ini juga akan dilakukan beberapa proses mulai dari ekstraksi fitur dengan TF-IDF, menampilkan data hasil klasifikasi *Naïve Bayes*, menampilkan grafik analisis sentimen, menampilkan *classification report* dan *confussion matrix* sebagai evaluasi dari model *Naïve Bayes* dalam melakukan klasifikasi *dataset*.

	lambuh	lupah	lingkung	lupat	lirik	lithium	logika	lokal	lokasi	lomba	lpg	luar
253	0	0	0	0	0.1227	0	0	0	0	0	0	0
254	0	0	0	0	0.0781	0	0	0	0	0	0	0
255	0	0	0	0	0.1592	0	0	0	0	0	0	0
256	0	0	0	0	0	0	0	0	0	0	0	0
257	0.2271	0	0	0	0.0563	0	0	0	0	0	0	0
258	0	0	0	0	0.1824	0	0	0	0	0	0	0
259	0	0	0	0	0	0	0	0	0	0	0	0
260	0.0969	0	0.0713	0	0.1802	0.1854	0	0	0	0	0	0
261	0	0	0	0	0.1005	0	0	0	0	0	0	0
262	0	0	0.4783	0	0.1904	0	0	0	0	0	0	0

Gambar 4.22. Tampilan ekstraksi fitur Tf-Idf pada fitur *Classification*

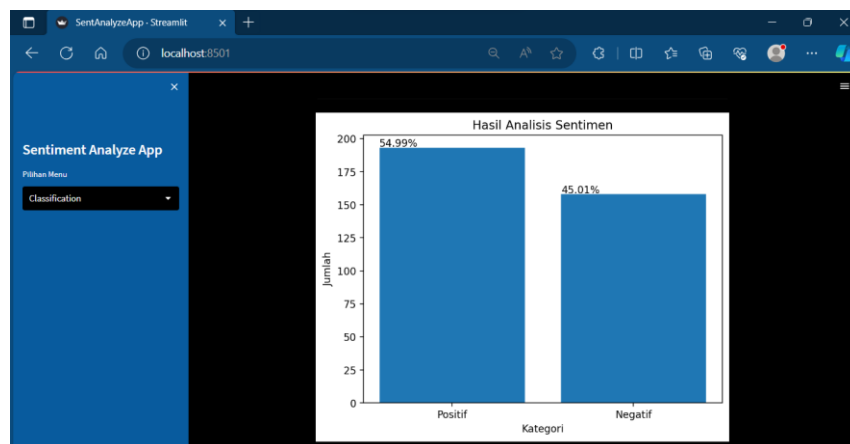
Berdasarkan gambar 4.22 merupakan proses ekstraksi fitur TF-IDF yang ditampilkan pada fitur *Classification*. Pada proses tersebut akan dilakukan pembobotan fitur kata dan merepresentasikan matriks TF-IDF (*Term Frequency-Inverse Document Frequency*) yang menunjukkan bagaimana kata-kata dalam teks direpresentasikan dalam bentuk vektor numerik berdasarkan konsep TF-IDF dari dataset yang diinputkan.



Comment	Label	Classification
0 pabrik mobil konvensional mobil konvensional untung	Positif	Negatif
1 energi baru tinggal mau perintah sungsai sumber tenaga matahari ang	Negatif	Negatif
2 perintah resmi terbit atur beri bantu beli kendaraan motor listrik bas baterai	Positif	Positif
3 sedih negara asi pihak guna kendaara listrik guna ket subsidi kendaara listrik tritung ma	Negatif	Negatif
4 jelas ahli ekonomi tepat subsidi usaha mobil listrik buka muka langsung subsidi muk	Negatif	Negatif
5 jalan bisnis mobil listrik starup main alih	Positif	Positif
6 mobil buang mesin ganti dinamo batrai carger solar seli	Negatif	Negatif
7 kelas toyota gampang mobil listrik	Positif	Positif
8 masyarakat bus susah ngubah biasa masyarakat warung meter pilih motor subsidi	Positif	Positif
9 situ ngotot bangun kn subsidi kendaara listrik nyedot anggar	Negatif	Negatif

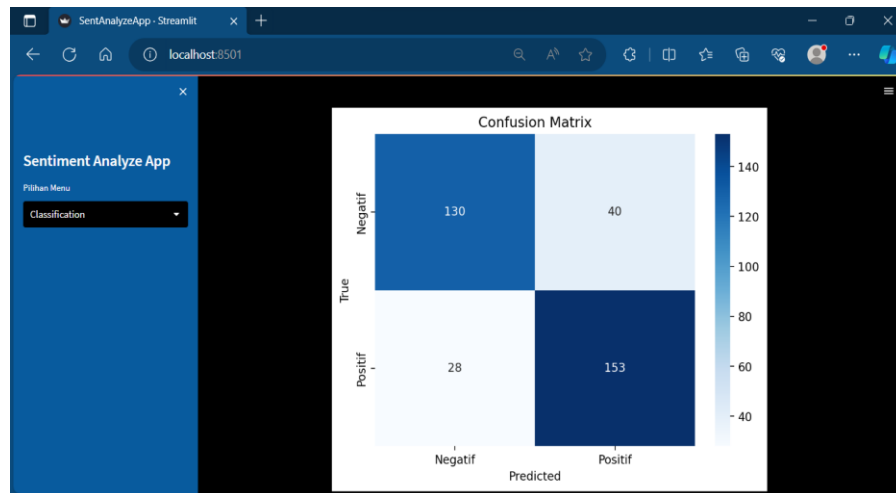
Gambar 4.23. Tampilan data hasil klasifikasi sentimen pada fitur *Classification*

Kemudian gambar 4.23 menampilkan proses hasil klasifikasi dataset teks komentar dalam analisis sentimen opini publik terhadap kendaraan listrik menggunakan algoritma *Naïve Bayes Classifier*. Dimana, pada gambar diatas ditampilkan tabel data yang terdiri dari kolom *Comment*, *Label* dan *Classification* sebagai hasil klasifikasi sentiment dengan *Naïve Bayes*.



Gambar 4.24. Tampilan grafik persentase klasifikasi sentimen pada fitur *Classification*

Sealin itu, sistem juga menampilkan grafik persentase dari hasil klasifikasi yang didapatkan seperti pada gambar 4.24 diatas. Dimana, pada gambar diatas menunjukkan bahawa sentimen positif yang di klasifikasikan oleh sistem sebesar 52,84% dan sentimen negatif sebesar 47,16% yang menunjukkan bahawa hasil klasifikasi sentimen terhadap kendaraan listrik dengan algoritma *Naïve Bayes Classifier* lebih dominan pada sentimen positif dan akurasi yang didapatkan sebesar 82%.



Gambar 4.25. Tampilan grafik *confusion matrix* pada fitur *Classification*

Selain itu juga ditampilkan visualisasi *confusion matrix* seperti gambar 4.25 diatas untuk melakukan evaluasi kinerja model klasifikasi yang dimana beberapa nilai yaitu:

- *True Positives (TP)*
Jumlah *instance* positif yang benar diprediksi oleh model sebagai positif. Dimana pada sistem menunjukkan nilai *True Positives (TP)* sebesar 153 data.
- *True Negatives (TN)*
Jumlah *instance* negatif yang benar diprediksi oleh model sebagai negatif. Dimana pada sistem menunjukkan nilai *True Negatives (TN)* sebesar 130 data.
- *False Positives (FP)*
Jumlah *instance* negatif yang salah diprediksi oleh model sebagai positif. Dimana pada sistem menunjukkan nilai *False Positives (FP)* sebesar 40 data.
- *False Negatives (FN)*
Jumlah *instance* positif yang salah diprediksi oleh model sebagai negatif. Dimana pada sistem menunjukkan nilai *False Negatives (FN)* sebesar 28 data.

Sentiment Analyze App

Pilihan Menu

Classification

Classification Report:

	precision	recall	f1-score	support
Negatif	0.8200	0.7600	0.7900	170.0000
Positif	0.7900	0.8500	0.8200	181.0000
accuracy	0.8100	0.8100	0.8100	0.8100
macro avg	0.8100	0.8100	0.8100	351.0000
weighted avg	0.8100	0.8100	0.8100	351.0000

Download Analysis Result

[Download Excel](#)

Gambar 4.26. Tampilan *classification report* pada fitur *Classification*

Selanjutnya, pada fitur *Classification* dalam sistem juga menampilkan *classification report* untuk melihat performa model klasifikasi *Naïve Bayes* dalam melakukan analisis sentimen seperti pada gambar 4.26 diatas. Dalam *classification report* diatas akan ditampilkan beberapa nilai evaluasi model berdasarkan *confusion matrix* sebelumnya diantaranya sebagai berikut:

- *Precision* (Presisi), untuk mengukur sejauh mana prediksi positif yang dilakukan oleh sistem adalah benar. Misalnya, pada kelas negatif, presisinya sebesar 0.8200, yang berarti 82.00% dari prediksi negatif sistem pada kelas tersebut adalah benar. Sedangkan pada kelas positif, presisinya sebesar 0.7900, yang berarti 79.00% dari prediksi positif sistem pada kelas tersebut adalah benar.
- *Recall* (Sensitivitas), untuk mengukur sejauh mana sistem berhasil mengklasifikasikan semua *instance* positif yang sebenarnya. Misalnya, pada kelas negatif, recallnya sebesar 0.7600, yang berarti sistem berhasil mendeteksi 76.00% dari semua *instance* yang sebenarnya termasuk dalam kelas tersebut. Sedangkan pada kelas positif, recallnya adalah 0.8500, yang berarti sistem berhasil mendeteksi 85.00% dari semua *instance* yang sebenarnya termasuk dalam kelas tersebut.
- *F1-score*, merupakan nilai perbandingan rata-rata antara *recall* dan *precision*. *F1-score* memberikan keseimbangan antara presisi dan *recall*. Misalnya, pada kelas negatif, F1-scorenya adalah 0.7900, yang menunjukkan keseimbangan antara presisi dan *recall* pada kelas tersebut. Sedangkan pada kelas positif, F1-scorenya adalah 0.8200.

- Support, merupakan jumlah *instance* aktual dalam setiap kelas. Pada kelas negatif, terdapat 170 *instance*, dan pada kelas positif, terdapat 181 *instance*.
- *Accuracy* (Akurasi) adalah persentase keseluruhan prediksi yang benar dari semua *instance* dalam *dataset*. Dalam kasus ini, akurasi yang didapat sebesar 81.00%.
- *Macro Average* adalah rata-rata dari metrik presisi, *recall*, dan *F1-Score* untuk semua kelas.
- *Weighted Average* adalah rata-rata berbobot dari metrik presisi, *recall*, dan *F1-score* untuk semua kelas. *Weighted average* memberi bobot berdasarkan jumlah *instance* dalam masing-masing kelas.

Kemudian, data hasil klasifikasi yang dilakukan oleh sistem dapat di *download* dalam format *excel* dengan menekan link *Download Excel* untuk mendapatkan *dataset* hasil analisis sentimen opini publik terhadap kendaraan listrik menggunakan algoritma *Naïve Bayes Classifier*.

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan dan mengacu pada tujuan dalam penelitian ini, maka diperoleh beberapa kesimpulan dengan data yang digunakan dalam rentang waktu mulai dari 15 April 2021 – 25 Mei 2023 diantaranya sebagai berikut:

- 1) Dari 3509 data teks komentar yang digunakan melalui hasil *crawling* pada *twitter* dan *youtbe* yang telah melalui tahap *preprocessing*, yang dimana data tersebut juga dapat diakses pada link *github* (<https://github.com/ryasakbar060/Dataset-Komentar-Kendaraan-Listrik>) menunjukkan bahwa, dengan pembagian data pada perbandingan rasio 90:10. Model berhasil mengklasifikasikan sejumlah 192 sentimen Positif dan 159 sentimen Negatif dari 351 data uji atau 10% dari data latih model.
- 2) Model klasifikasi analisis opini publik terhadap kendaraan listrik menggunakan algoritma *Naïve Bayes Classifier* dengan ekstraksi fitur *Tf-Idf* pada penggunaan 2 kelas (Positif dan Negatif) menghasilkan nilai akurasi 81%, presisi 79%, *recall* 85% dan *f1-score* 82%. Sedangkan, model klasifikasi tanpa ekstraksi fitur *Tf-Idf* mendapatkan akurasi 77%, presisi 78%, *recall* 78% dan *f1-score* 78%. Kemudian pada penggunaan 3 kelas (Positif, Netral dan Negatif) *Naïve Bayes Classifier* dengan ekstraksi fitur *Tf-Idf* mendapat akurasi 64%, presisi 69%, *recall* 82% dan *f1-score* 75%. Sedangkan, klasifikasi tanpa ekstraksi fitur *Tf-Idf* mendapat akurasi 62%, presisi 71%, *recall* 74% dan *f1-score* 73%.
- 3) Penerapan model klasifikasi analisis menggunakan *Naïve Bayes Classifier* dilakukan dengan pengintegrasian model klasifikasi kedalam aplikasi berbasis *web* yang dibangun menggunakan *framework streamlit*. Aplikasi ini akan menerapkan beberapa proses dalam perancangan model yang telah dibuat sebelumnya menjadi beberapa fitur seperti *crawling*, *preprocessing*, klasifikasi, visualisasi, dan evaluasi performa model berdasarkan data teks komentar opini publik terhadap kendaraan listrik.

5.2 Saran

Berdasarkan penelitian ini, terdapat beberapa kekurangan dan potensi pengembangan yang dapat menjadi referensi bagi penelitian selanjutnya, diantaranya sebagai berikut.

- 1) Data pada penelitian ini didapatkan dengan *crawling* data pada media sosial *twitter* dan *youtube*. Diharapkan pada penelitian selanjutnya menambahkan data melalui media atau *platform* lainnya agar mendapat data yang lebih bervariasi.
- 2) Dalam penelitian ini data yang digunakan adalah data berbahasa indonesia dan *labelling* data dilakukan secara manual. Diharapkan pada penelitian selanjutnya menggunakan metode *labelling* otomatis dengan *library* yang *support* untuk proses *laeblling* data berbahasa indonesia.
- 3) Diharapkan pada penelitian selanjutnya menggunakan metode lainnya seperti algoritma *machine learning* yang berbeda dan penggunaan *feature extraction* lainnya yang dapat meningkatkan performa model semakin baik.
- 4) Diharapkan pada penelitian selanjutnya, dalam perancangan sistem aplikasi sentimen analisis dapat menggunakan *framework* yang lainnya dalam mengintegrasikan model *machine learning* sehingga mendapatkan tampilan yang lebih menarik.

DAFTAR PUSTAKA

- Agustian, A., Tukino, & Nurapriani, F. (2022). Penerapan Analisis Sentimen Dan Naïve Bayes Terhadap Opini Penggunaan Kendaraan Listrik Di Twitter. *Jurnal Tika Fakultas Ilmu Komputer Universitas Almuslim*, 7(3), 243-249.
- Anwar, M., & Permana, D. (2023). Analisis Sentimen Masyarakat Indonesia Terhadap Produk Kendaraan Listrik Menggunakan VADER. *Jurnal Teknik Informatika dan Sistem Informasi*, 10(1), 783-792.
- Arianto, M. A., & Solichin, A. (2022). Analisis Sentimen Motogp Mandalika Pada Twitter Menggunakan Metode Naïve Bayes. *Jurnal Ticom: Technology of Information and Communication*, 11(1), 20-25.
- Attabi, A. W. (2018). Penerapan Analisis Sentimen Untuk Menilai Suatu Produk Pada Twitter Berbahasa Indonesia Dengan Metode Naïve Bayes Classifier Dan Information Gain [Skripsi, Universitas Brawijaya]. *Repository Universitas Brawijaya*.
- Fatihin, A. (2022). Analisis Sentimen Terhadap Ulasan Aplikasi Mobile Menggunakan Metode Support Vector Machine [Skripsi, Universitas Islam Negeri Syarif Hidayatullah]. *Repository Universitas Islam Negeri Syarif Hidayatullah*.
- Febriyani, E., & Februariyanti, H. (2022). Analisis Sentimen Terhadap Program Kampus Merdeka Menggunakan Algoritma Naive Bayes Classifier Di Twitter. *Jurnal Tekno Kompak*, 17(1), 25-38.
- Imron, A. (2019). Analisis Sentimen Terhadap Wisata Di Kabupaten Rembang Menggunakan Metode Naïve Bayes Classifier [Skripsi, Universitas Islam Indonesia]. *Repository Universitas Islam Indonesia*.
- Irawansyah, R. S. (2022). Analisis Sentimen Terhadap Program Merdeka Belajar-Kampus Merdeka Pada Twitter Menggunakan Algoritma Naive Bayes Classifier (NBC) [Skripsi, Universitas Mataram]. *Repository Universitas Mataram*.
- Karim, M. (2022). Analisis Sentimen Pada Twitter Menggunakan Support Vetor Machine Dengan Modifikasi Lexicon Inset Dan Sentis-Trength_Id [Skripsi, Universitas Islam Negeri Syarif Hidayatullah]. *Repository Universitas Islam Negeri Syarif Hidayatullah*.

- Khasanah, L. U. (2022). Tiga Metode Machine Learning yang Wajib Diketahui. *DQLab*. Retrieved from <https://dqlab.id/tiga-metode-machine-learning-yang-wajib-diketahui>
- Khomsah, S., & Aribowo, A. S. (2020). Model Text-Preprocessing Komentar Youtube Dalam Bahasa Indonesia. *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)*, 4(4), 648-654.
- Mustaqim, T. (2020). Sentiment Analysis Opini Pelantikan Kabinet Pemerintah Indonesia Tahun 2019 Menggunakan Vader dan Random Forest [Skripsi, Universitas Negeri Semarang]. *Repository Universitas Negeri Semarang*.
- Pangestu, D. A. (2020). Analisis Sentimen Terhadap Opini Publik Tentang Kesehatan Mental Selama Pandemi Covid-19 Di Media Sosial Twitter Menggunakan Naïve Bayes Classifier Dan Support Vector Machine [Skripsi, Universitas Islam Indonesia]. *Repository Universitas Islam Indonesia*.
- Pratama, P., Murdiansyah, D., & Lhaksana, K. (2023). Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma Logistic Regression dan Principal Component Analysis. *Jurnal Media Informatika Budidarma*, 7(1), 592-535.
- Pratama, Y., Murdiansyah, D., & Lhaksana, K. (2023). Analisis Sentimen Kendaraan Listrik Pada Media Sosial Twitter Menggunakan Algoritma Logistic Regression dan Principal Component Analysis. *Jurnal Media Informatika Budidarma*, 7(1), 529-535.
- Qinanda, M. D., Nilogiri, A., & Timur, T. W. (2022). Sentimen Pada Komentar Youtube Tentang Pencegahan Dan Penanganan Kekerasan Seksual Pada Permendikbud Berbasis Naive Bayes Dan Support Vector Machine. *Jurnal Sistem dan Teknologi Informasi (JUSTINDO)*, 7(2), 114-121.
- Riyadi, A. F., Rahman, F. R., Pratama, M. A., Khafidli, M. K., & H, P. (2022). Pengukuran Sentimen Sosial Terhadap Teknologi Kendaraan Listrik: Bukti Empiris di Indonesia. *Jurnal Manajemen Sistem Informasi Dan Teknologi*, 12(2), 141-149.
- Santoso, A., Nugroho, A., & Sunge, A. S. (2022). Analisis Sentimen Tentang Mobil Listrik Dengan Metode Support Vector Machine Dan Feature Selection Particle Swarm Optimization. *Jurnal Of Practical Computer Science*, 2(1), 24-31.

- Sasmita, A., Pradnyana, G. A., & Divayana, D. G. (2022). Sistem Analisis Sentimen Untuk Evaluasi Kinerja Dosen Dengan Metode Naive Bayes. *Jurnal Sains dan Teknologi*, 11(2), 451-462.
- Siswandi, A. (2019). Implementasi Data Mining Dengan Metode Klasifikasi Naive Bayes Untuk Memprediksi Stok Bahan Jadi. *Jurnal Teknologi Pelita Bangsa*, 10(2), 1-10.
- Toy, K. V., Sari, Y. A., & Cholissodin, I. (2021). Analisis Sentimen Twitter Menggunakan Metode Naive Bayes Dengan Relevance Frequency Feature Selection (Studi Kasus: Opini Masyarakat Mengenai Kebijakan New Normal). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 5(11), 5068-5074.
- Zhafira, D. F., Rahayudi, B., & Indriati. (2021). Analisis Sentimen Kebijakan Kampus Merdeka Menggunakan Naive Bayes Dan Pembobotan Tf-Idf Berdasarkan Komentar Pada Youtube. *Jurnal Sistem Informasi, Teknologi Informasi dan Edukasi Sistem Informasi*, 3(1), 55-63.