KOMPARASI ALGORITMA NAIVE BAYES CLASSIFIER DAN SUPPORT VECTOR MACHINE (SVM) DALAM ANALISIS SENTIMEN OPINI MASYARAKAT TERHADAP POLUSI UDARA JAKARTA DI MEDIA SOSIAL TWITTER

SKRIPSI

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh Gelar Sarjana Sastra Satu (S-1) dalam Ilmu Teknologi Informasi

Diajukan Oleh :

ISTI NUR AZIZAH
NIM : 2008096041

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG

2024

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini:

Nama : Isti Nur Azizah

NIM : 2008096041

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

Komparasi Algoritma Naive Bayes Classifier Dan Support Vector Machine (SVM) Dalam Analisis Sentimen Opini Masyarakat Terhadap Polusi Udara Jakarta Di Media Sosial Twitter

Secara keseluruhan adalah hasil penelitian/karya saya sendiri, kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, Juni 2024

Pembuat Pernyataan,

Isti Nur Azizah

NIM : 2008096041

KEMENTRIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Prof. Dr. Hamka Ngaliyan Semarang
Telp.024-7601295 Fax.7615387

PENGESAHAN

Naskah skripsi berikut ini:

Judul	:	Komparasi Algoritma Naïve Bayes Classifier dan Support Vector Machine (SVM) Dalam Analisis Sentimen Opini Masyarakat Terhadap Polusi Udara Jakarta Di Media Sosial Twitter
Penulis	:	Isti Nur Azizah
NIM	:	2008096041
Jurusan	:	Teknologi Informasi

Telah diujikan dalam tugas akhir oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam bidang ilmu Teknologi Informasi.

Semarang, 7 Juni 2024

DEWAN PENGUJI

Penguji I
Masy Ari Ulinuha, M.T
NIP.198108122011011007

Penguji II
Siti Nur'aini, M.Kom
NIP. 198401312018012001

Penguji III
Wenty Dwi Yuniarti, S.Pd, M.Kom
NIP. 197706222006042005

Penguji IV
Heru Mustofa, M.Kom.
NIP. 198703172019031007

Pembimbing I,
Nur Cahyo Hendrowibowo, S.T.,M.Kom
NIP. 197312222006041001

Pembimbing II,
Siti Nur'aini, M.Kom
NIP. 198401312018012001

iii

NOTA PEMBIMBING

Semarang, 30/April/2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi UIN Walisongo
Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul	:	**Komparasi Algoritma Naive Bayes Classifier dan Support Vector Machine (SVM) Dalam Analisis Sentimen Opini Masyarakat Terhadap Polusi Udara Jakarta Di Media Sosial Twitter**
Nama	:	**Isti Nur Azizah**
NIM	:	2008096041
Jurusan	:	Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqsyah.

Wassalamu'alaikum. wr. wb.

Pembimbing I,

Nur Cahyo Hendro Wibowo, S.T.,M.Kom.
NIP : 19731222 200604 1 001

NOTA PEMBIMBING

Semarang, 30/April/2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi UIN Walisongo
Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul	:	**Komparasi Algoritma Naive Bayes Classifier dan Support Vector Machine (SVM) Dalam Analisis Sentimen Opini Masyarakat Terhadap Polusi Udara Jakarta Di Media Sosial Twitter**
Nama	:	**Isti Nur Azizah**
NIM	:	2008096041
Jurusan	:	Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqsyah.

Wassalamu'alaikum. wr. wb.

Pembimbing II,

Siti Nur'aini, M.Kom.
NIP : 19840131201801 2 001

ABSTRAK

Buruknya kualitas udara di Jakarta telah menjadi isu yang diperbincangkan sejak bulan Agustus 2023 di media sosial Twitter. Peneliti mengolah data tanggapan, opini serta keluhan pengguna Twitter terkait pro dan kontra akan polusi udara di Jakarta untuk dilakukan analisis sentimen opini masyarakat terhadap polusi udara Jakarta menggunakan dua model klasifikasi yaitu *Naive Bayes Classifier* dan *Support Vector Machine*. Tahapan awal penelitian ini dengan melakukan crawling data pada twitter menggunakan kata kunci "polusi udara jakarta" yang diambil dari periode 1 Agustus sampai 31 Agustus 2023 dan mendapatkan data sebanyak 510 data tweet. Selanjutnya melalui tahapan labelling data dibagi menjadi dua sentimen dan didapatkan hasil untuk sentimen positif sebanyak 208 data dan sentimen negatif sebanyak 284 data. Tahapan ketiga yaitu text preprocessing untuk mempermudah perhitungan pada tahap selanjutnya. Alur preprocessing yaitu *case folding, cleansing, remove duplicate, tokenization, normalization, stopword removal,* dan *stemming*. Setelah dilakukan pembersihan data diperoleh 492 data tweet yang kemudian data akan melalui proses pembobotan kata TFIDF.

Klasifikasi dengan metode NBC dan SVM dilakukan uji sebanyak 4 kali berdasarkan 4 perbandingan pembagian data latih dan data uji diantaranya 90:10, 80:20, 70:30, dan 60:40. Pengujian performa dilakukan dengan perhitungan akurasi, *precision, recall,* dan *f1 score* yang menghasilkan nilai performa tertinggi diperoleh metode NBC perbandingan data 90:10 dengan nilai akurasi 80%, *precision* 0,85, nilai *recall* sebesar 0,88 serta nilai *f1 score* sebesar 0,82. Sedangkan metode SVM tertinggi pada perbandingan 70:30 dengan nilai akurasi sebesar 77,027%. Pada penelitian ini diperoleh nilai rata-rata performa tertinggi pada metode NBC dengan perbandingan 90:10 dengan nilai *precision, recall,* dan *f1 score* secara berurutan sebesar 0,81, 0,80, dan 0,80.

Kata kunci : polusi udara, sentimen, Twitter, NBC, SVM

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, karna berkat rahmat dan hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan baik dan lancar. Penulisan skripsi ini merupakan salah satu persyaratan dalam menempuh gelar Sarjana Komputer di program studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang. Penelitian ini dilakukan untuk menggali pemahaman yang lebih dalam tentang subjek yang telah dipilih, yaitu Komparasi Algoritma Naïve Bayes Classifier dan Support Vector Machine (SVM) Dalam Analisis Sentimen Opini Masyarakat Terhadap Polusi Udara Jakarta Di Media Sosial Twitter. Dalam proses penulisan skripsi terdapat banyak pihak yang telah memberikan dukungan dan bantuan dalam penulisan skripsi ini. Penulis mengucapkan terima kasih kepada :

1. Bapak Prof. Dr. Nizar, M.Ag, selaku Rektor Universitas Islam Negeri Walisongo Semarang.
2. Bapak Dr. H. Musahadi, M.Ag, selaku Dekan Fakultas Teknologi Informasi Universitas Islam Negeri Walisongo Semarang.

3. Bapak Dr. Khotibul Umam S. T., M. Kom. selaku ketua program studi Teknologi Informasi Universitas Islam Negeri Walisongo Semarang
4. Bapak Nur Cahyo Hendro Wibowo, S.T., M.Kom selaku Dosen Pembimbing satu skripsi saya yang selalu memberikan bimbingan dan bantuannya dalam pembuatan skripsi ini.
5. Ibu Siti Nur'aini, M. Kom selaku dosen pembimbing dua skripsi saya yang selalu memberikan bimbingan dan bantuannya dalam pembuatan skripsi ini.
6. Orang tua dan keluarga tersayang yang selalu mendo'akan dan memberikan dukungan kepada penulis.
7. Teman-teman Teknologi Informasi, rekan seperjuangan pada saat kuliah yang selalu memberikan dukungan.
8. Semua pihak yang telah membantu hingga selesainya laporan ini yang tidak dapat penulis sebutkan satu per satu

Penulis menyadari bahwa penelitian ini belum sempurna, dan masih banyak ruang untuk pengembangan dan peningkatan di masa depan. Hal ini disebabkan keterbatasan kemampuan dan pengetahuan dari penulis. Oleh karena itu, penulis berharap bahwa penelitian ini dapat menjadi landasan untuk penelitian-penelitian selanjutnya yang lebih baik dan mendalam dan mampu memberikan manfaat dan kontribusi untuk pengembangan ilmu pengetahuan pembaca serta semua pihak yang terlibat.

Semarang, Juni 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

A. Latar Belakang

Perubahan keadaan alam digambarkan dengan adanya kerusakan yang ditimbulkan karena perbuatan manusia terhadap lingkungan sekitar. Dalam ajaran agama Islam terdapat suatu ayat yang mengingatkan akan tanggung jawab kaum Muslim untuk menjaga alam semesta, seperti dijelaskan dalam ayat Al-Qur'an Surah Ar- Rum ayat 41.

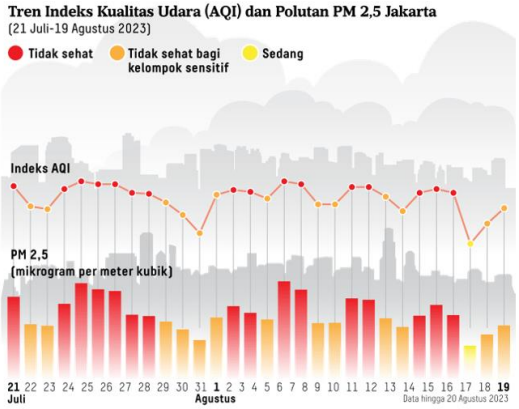
ظَهَرَ الْفَسَادُ فِى الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ اَيْدِى النَّاسِ لِيُذِيْقَهُمْ بَعْضَ الَّذِىْ عَمِلُوْا لَعَلَّهُمْ يَرْجِعُوْنَ

Artinya : Telah nampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia; Allah menghendaki agar mereka merasakan sebagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar) (QS Ar-Rum 30 ayat 41).

Pada ayat di atas Allah SWT menjelaskan tentang kerusakan di daratan dan di laut dikarenakan perbuatan manusia. Salah satunya terkait udara yang dihirup setiap harinya semakin memburuk. Manusia dalam usahanya untuk mencapai kenyamanan hidup seringkali melibatkan aktivitas yang menghasilkan polusi seperti emisi gas dari kendaraan bermotor, polusi pabrik industri dan lainnya. Polusi udara ini mengancam kualitas udara dan juga

berdampak negatif pada kesehatan manusia dan lingkungan.

Buruknya kualitas udara di Jakarta telah menjadi isu yang diperbincangkan selama empat dekade terakhir. Namun, pembahasan tentang masalah ini seringkali muncul dan tenggelam tanpa tindakan serius dan berkelanjutan. Isu buruknya udara di Jakarta muncul lagi dan tampaknya berpotensi hilang tanpa solusi tindakan yang kuat. Menurut laporan dari situs IQAir pada tanggal 11 Agustus 2023, kualitas udara di Jakarta berada pada kategori tidak sehat.



Gambar 1. 1 Kualitas Udara Jakarta (AQI) Bulan Juli - Agustus

Sejak bulan Agustus 2023, kondisi Jakarta yang diselimuti oleh kabut akibat polusi udara telah menjadi topik hangat dalam percakapan, seperti di media sosial bahkan dalam kehidupan sehari-hari masyarakat.

Informasi tentang Jakarta yang dinyatakan sebagai salah satu kota dengan kualitas udara terburuk di dunia tersebar luas di berbagai platform daring dan media sosial (Indraswari, 2023). Pengguna *Twitter* seringkali membagikan opini, perasaan, dan keluhan mereka tentang kualitas udara di Jakarta. Tidak diragukan, banyak pengguna *Twitter* memberikan respons mereka terkait polusi udara di Jakarta dalam bentuk komentar, termasuk tanggapan positif dan negatif. Oleh karena itu, kumpulan tanggapan, opini, serta keluhan dari respon pengguna *Twitter* terkait pro dan kontra polusi udara di Jakarta ini dapat klasifikasikan dan dimanfaatkan dalam penelitian (Herianto, 2018)

Klasifikasi data dalam jumlah besar dapat ditingkatkan dengan menerapkan berbagai algoritma. Algoritma-algoritma umum yang sering digunakan diantaranya Naïve Bayes dan Support Vector Machines yang memungkinkan pengolahan data yang cepat dan akurat (Wibisono dan Fahrurozi, 2019). Dalam algoritma klasifikasi, tingkat akurasi adalah salah satu faktor kunci untuk menilai keberhasilan algoritma. Semakin tinggi tingkat akurasi, semakin baik kemampuan algoritma dalam mengklasifikasikan data dengan tepat. Salah satu teknik klasifikasi probabilitas yang sangat sederhana dimiliki oleh

metode *Naïve Bayes Classifier* dan metode ini dikenal memiliki tingkat akurasi yang tinggi ketika diterapkan pada basis data besar. Demikian halnya dengan metode *Support Vector Machin*e (SVM), yang merupakan metode *supervised learning* yang digunakan untuk mengenali pola dan menganalisis data dalam konteks klasifikasi. Pemilihan metode *Support Vector Machine* (SVM) didasari oleh kemampuannya untuk memberikan hasil klasifikasi dengan akurasi yang tinggi, proses pembelajaran yang cepat, dan kemampuan untuk menemukan *hyperplane* terbaik sebagai pemisah. Metode SVM juga diakui sebagai salah satu metode klasifikasi teks yang paling efektif karena mampu mengelola ruang fitur yang besar dan memiliki tingkat generalisasi yang tinggi. (Hakim 2021).

Pendekatan klasifikasi dapat digunakan untuk mengelompokkan tanggapan ke dalam bentuk tanggapan positif, tanggapan negatif maupun tanggapan yang bersifat netral. Hal tersebut termasuk komponen penting dalam analisis sentimen(Sodik dan Kharisudin, 2021). Pada dasarnya analisis sentimen adalah proses mengidentifikasi dan mengelompokkan opini yang masih berupa teks ke dalam kategori sentimen yang sesuai, seperti positif, negatif atau netral. Analisis sentimen melibatkan evaluasi pendapat, sentimen, emosi, penilaian, dan sikap terhadap

berbagai entitas seperti produk, tokoh, organisasi, layanan, isu, atau peristiwa yang memengaruhi masyarakat. Analisis ini sangat erat kaitannya dengan masyarakat karena berfokus pada sumber informasi yang berasal dari media sosial, tempat di mana masyarakat secara aktif berpartisipasi dan berinteraksi.

Penggunaan analisis sentimen bisa sangat berguna dalam berbagai konteks seperti analisis produk, pengawasan merek, politik dan berita, media sosial, dan pelayanan pelanggan. Pada penelitian ini akan menganalisis pada konteks media sosial. Penggunaan internet di Indonesia saat ini mengalami pertumbuhan pesat. Salah satu platform media sosial yang populer adalah *Twitter*, yang memungkinkan pengguna untuk menyampaikan pendapat dan memantau trending topik yang sedang ramai. Peran media sosial dalam opini masyarakat khususnya *Twitter*, telah menjadi platform utama bagi masyarakat untuk berbicara tentang kampanye, tren, atau isu-isu lingkungan, termasuk polusi udara. Pertumbuhan cepat pengguna *Twitter* menjadi subjek menarik untuk menganalisis berbagai opini dan fenomena. Mayoritas informasi yang terdapat di *Twitter* adalah data teks, yang berisi opini dan pendapat dalam bentuk kalimat tidak terstruktur, sering kali mengandung

banyak noise. Diperlukan analisis yang tepat untuk mengolah data teks ini dan menghasilkan informasi yang akurat (Herianto, 2018).

Berdasarkan latar belakang di atas, evaluasi opini mengenai polusi udara di media sosial Twitter akan diolah melalui analisis sentimen dengan pendekatan klasifikasi. Dalam pengolahan ini, digunakan dua metode, yaitu *Support Vector Machine* (SVM) dan *Naive Bayes Classifier* (NBC). Hasil dari metode *Naive Bayes dan Support Vector Machine* dievaluasi menggunakan *Confusion Matrix* untuk menghitung akurasi, *precision, recall, f1 Score* (Meilany, 2022). Penelitian ini membandingkan kinerja dan performa antara kedua algoritma setelah data tersebut melewati tahap *preprocessing*.

Penelitian ini membandingkan algoritma klasifikasi *Naive Bayes Classifier* (NBC) dan *Support Vector Machine* (SVM) dengan menggunakan data tweet untuk analisis sentimen masyarakat terhadap polusi udara di Jakarta berdasarkan sentimen positif dan negatif.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka pokok dari rumusan masalah pada penelitian ini yaitu sebagai berikut :

1. Bagaimana analisis sentimen opini masyarakat terhadap polusi udara Jakarta di media sosial *Twitter* menggunakan algoritma *Naïve Bayes Classifier* dan *Support Vector Machine* (SVM) ?
2. Bagaimana performa yang diberikan oleh algoritma *Naive Bayes Classifier* dan *Support Vector Machine* (SVM) pada analisis sentimen opini Masyarakat terhadap polusi udara Jakarta di media sosial *Twitter*?

C. Tujuan Penelitian

Dari rumusan masalah yang telah diberikan, penelitian ini bertujuan untuk :

1. Mengetahui sentimen opini masyarakat terhadap polusi udara Jakarta di media sosial *Twitter* menggunakan algoritma *Naïve Bayes Classifier* dan *Support Vector Machine* (SVM)
2. Mengetahui performa dari hasil perhitungan algoritma *Naive Bayes Classifier* dan *Support Vector Machine* (SVM) pada analisis sentimen opini Masyarakat terhadap polusi udara Jakarta di media sosial *Twitter*

D. Batasan Masalah

Untuk menjalankan penelitian dengan obyektif dan jelas, peneliti mendefinisikan batasan-batasan yang diperlukan. Berikut adalah batasan-batasan yang diterapkan dalam penelitian ini::

1. Data yang digunakan adalah data dari media sosial *Twitter* yang berbahasa Indonesia
2. Sentiment analisis opini Masyarakat terhadap polusi udara Jakarta dilakukan dengan metode *Naive Bayes Classifier* dan *Support Vector Machine* (SVM)
3. Data *Twitter* polusi udara Jakarta tanggal 1 Agustus 2023 – 31 Agustus 2023.
4. Kata kunci pencarian yang diteliti pada data *Twitter* adalah "polusi udara Jakarta"
5. Jumlah data yang digunakan adalah 510 tweet
6. Penelitian ini melakukan klasifikasi sentiment menjadi dua kelas yaitu positif dan negatif

E. Manfaat Penelitian

1. Manfaat Teoritis :

a. Membantu menganalisis sentiment dengan algoritma *Naive Bayes* dan *Support Vector Machine* (SVM).

b. Sebagai acuan dan pertimbangan bagi penelitian selanjutnya khususnya berkaitan dengan analisis sentiment.

2. Manfaat Praktis :

a. Memudahkan pihak pemerintah untuk melihat informasi tentang sentiment Masyarakat mengenai opini Masyarakat terhadap polusi udara Jakarta, sehingga dapat fokus melakukan penanganan serta evaluasi kearah yang lebih baik

b. Membantu masyarakat mengetahui jumlah sentiment opini masyarakat terhadap polusi udara Jakarta lebih cenderung ke hal positif atau negatif

BAB II

LANDASAN TEORI

A. Kajian Pustaka

1. Text Mining

Teknik yang digunakan dalam klasifikasi dokumen mencakup *Text Mining* yang merupakan turunan dari Data Mining. Tujuannya adalah untuk mengidentifikasi pola-pola menarik dari berbagai data teks yang besar. (Nuari, 2018). *Text Mining* mengelola data yang lebih tidak terstruktur atau dikenal sebagai unstructured data, sedangkan data mining mengelola data yang lebih terstruktur (Sholekha et al., 2022). Menurut Fitriyah, Warsito, & Maruddani (2020) Data yang digunakan dalam *Text Mining* adalah data berupa teks yang bersifat tidak terstruktur, seperti yang ditemukan dalam *tweet* di *Twitter*, artikel, dan berbagai sumber lainnya. Data yang bersifat tidak terstruktur ini meliputi teks, video, audio, foto, serta berbagai jenis format data lainnya.

Text Mining berguna untuk menganalisis opini, sentimen, evaluasi, sikap, penilaian, serta emosi individu terkait dengan topik, layanan, organisasi, individu, atau aktivitas tertentu. Metode ini

memungkinkan pengelompokan, klasifikasi, pengambilan informasi, dan ekstraksi informasi yang diperlukan dalam analisis data teks. (Zidan, 2022).

2. Analisis Sentimen

Analisis sentimen adalah tindakan mengkategorikan suatu emosi dalam polaritas teks dalam frasa sehingga dapat di nilai menjadi sentiment positif, negatif, atau netral (Samsir et al. 2021). Analisis sentiment yang juga dikenal sebagai *opinion mining* memiliki peran penting dalam bidang pengelolaan Bahasa alami, komputasi linguistic, dan *Text Mining*. Tujuannya adalah untuk mengidentifikasi dan menganalisis sentimen masyarakat, seperti sikap, opini, emosi terhadap berbagai elemen seperti produk, individu, topik, organisasi, dan layanan, sebagaimana dijelaskan oleh Sonawanse(2016).

Analisis sentimen melibatkan klasifikasi dokumen teks ke dalam berbagai kategori, termasuk sentimen positif dan negatif. Signifikansi dan manfaat dari analisis sentimen telah mendorong pertumbuhan penelitian dan aplikasi di bidang ini. Perkembangan penelitian dalam analisis sentimen mengalami perkembangan pesat, bahkan di Amerika Serikat, lebih

dari 20 hingga 30 perusahaan telah memfokuskan pada layanan analisis sentimen. Meskipun pada dasarnya analisis sentimen adalah bentuk klasifikasi, dalam praktiknya, ini tidaklah sederhana karena berhubungan dengan kompleksitas bahasa, termasuk ambiguitas dalam penggunaan kata, ketiadaan intonasi dalam teks, dan perkembangan bahasa itu sendiri. (Nomleni, 2015)

Analisis sentimen memiliki berbagai kegunaan bagi sektor bisnis, seperti melacak produk, jasa, merek, dan audiens target di pasar. Selain itu, analisis sentimen juga memungkinkan penilaian atas kelebihan dan kekurangan produk serta jasa tertentu. Pada dasarnya, penggunaan analisis sentimen mencakup deteksi keluhan, persepsi terhadap produk atau layanan baru, dan citra merek tertentu (Hakim, 2021).

3. Media Sosial *Twitter*

Platform media sosial *Twitter* yang populer di Indonesia telah digunakan secara luas untuk memposting dan mempublikasikan opini masyarakat mengenai berbagai aspek, termasuk hiburan, layanan, dan banyak hal lainnya. Pengguna *Twitter* memiliki kemampuan untuk menyampaikan pendapat pribadi mereka, bahkan menggunakannya sebagai platform

bisnis. *Twitter* dirancang dengan tujuan untuk memfasilitasi interaksi sosial, memudahkan pencarian informasi, memperoleh berita, melakukan dokumentasi, dan berbagi informasi, seperti yang dikemukakan dalam penelitian yang dilakukan oleh Fikri, Sabrila, dan Azhar pada tahun 2020.

Twitter adalah platform media sosial yang memungkinkan pengguna untuk berbagi pemikiran dan pendapat melalui pesan tweet, yang dapat berupa teks, gambar, atau video. Yang membedakan *Twitter* dari platform lain adalah pembatasan karakter pesan hingga 280 karakter. Meskipun bersifat publik, tweet dapat dibagikan hanya kepada teman atau pengikut. *Twitter* memiliki keunggulan dalam jangkauan, mencapai figur publik, promosi yang lebih luas, banyak jaringan, dan kemampuan pengukuran yang lebih baik. Berikut adalah beberapa fitur utama *Twitter* yaitu *tranding topic, hastag, retweet, following* (I. Taufik dan S.A.Pamungkas, 2018)

Sejak Elon Musk mengambil alih Twitter, platform media sosial ini telah mengalami transformasi besar. Tujuannya adalah mengubah Twitter dari jaringan mikroblogging menjadi 'super aplikasi,' tetapi perubahan ini tidak selalu disambut baik oleh

pengguna. Saat ini, Twitter kembali menjadi perhatian media karena perubahan merek drastis yang mencakup nama dan logo baru. Hanya dalam satu hari, Elon Musk berhasil melakukan rebranding yang monumental untuk platform tersebut, yang jauh dari sekadar perubahan minor pada merek; itu merupakan transformasi lengkap. Dia menggantikan merek Twitter dengan yang benar-benar baru: X. Pada tanggal 23 Juli 2023, Elon Musk mengusulkan melalui beberapa cuitan bahwa nama dan logo Twitter mungkin akan segera berubah. Pada tanggal 24 Juli 2023, nama merek dan logo baru telah diumumkan secara publik. Merek baru "X" secara tiba-tiba menggantikan merek Twitter yang terkenal dan burung biru yang sudah dikenal. Nama domain situs web "x.com" juga mengarahkan ke "twitter.com." Dan tampaknya "tweet" sekarang akan disebut "X's." (Andrivet, 2023).

Data *Twitter* dapat diambil menggunakan aplikasi yang dihubungkan dan *Twitter*. Keterbatasan platform media sosial lainnya adalah mereka memiliki kebijakan privasi yang beragam dan tidak memfasilitasi akses mudah ke data mereka. Mengumpulkan data secara terbuka dan otomatis dari platform semacam itu bukanlah tugas yang sedehana.

Opini yang terdapat dalam tweet umumnya terdapat dalam teks bebas yang tidak terstruktur dan tidak memiliki format standar. Dalam pemilihan platform untuk *Text Mining*, *Twitter* dipilih karena terdapat tiga poin utama yang mendukungnya.(Wandani, 2021)

1. Format data *Twitter* yang nyaman bagi peneliti dan cocok untuk analisis.
2. Peraturan *Twitter* tergolong fleksibel dalam hal data jika dibandingkan dengan API lainnya.
3. *Twitter* dirancang dengan desain antarmuka yang mudah diakses bagi penggunanya.

Perubahan banyak terjadi pada Twitter sejak diakuisisi oleh Elon Musk pada tahun 2022, salah satunya yaitu pada perubahan fungsi utama. Pada 3 Juli 2023 Elon Musk mengumumkan penetapan batas jumlah tweet yang dapat dilihat pengguna setiap hari. Pengguna tidak dapat melihat tweet tanpa masuk ke platform X. Tujuan dilakukannya pembatasan adalah membantu mengatasi pengambilan data dari Twitter dalam jumlah besar dan manipulasi sistem yang ekstrim oleh hampir semua orang termasuk dari perusahaan AI dan Starup hingga raksasa teknologi. Batasan data yang ditetapkan untuk akun terverifikasi adalah dapat membaca 10000 postingan per hari, akun

yang belum terverifikasi 1000 postingan, dan 500 untuk akun baru yang belum terverifikasi (Andrivet,2023). Maka dari itu, pada penelitian ini peneliti mengambil batasan untuk data *tweet* yang diambil dari *twitter* sebanyak 500 data.

4. Crawling Data

Proses crawling data di *Twitter* melibatkan pengambilan data dari server *Twitter* menggunakan Application Programming Interface (API) *Twitter*, yang mencakup data pengguna dan data tweet (Mas Pintoko & Muslim, 2018). Crawling data ini bertujuan untuk mengumpulkan data dari *Twitter* yang diperlukan dalam penelitian ini. Prosesnya melibatkan pembuatan program yang memanfaatkan kata kunci untuk mencari tweet yang relevan sesuai kebutuhan penelitian (Zidan, 2022). Pada penelitian ini, *crawling data* dilakukan dengan menggunakan kata kunci "polusi udara Jakarta", program akan mengambil tweet yang berkaitan dengan kata kunci tersebut dan hasil dari tweet yang diperoleh merupakan data yang akan digunakan.

5. Data Labelling

Data *labelling* adalah tahap penting dalam pengolahan data, yang melibatkan identifikasi data

mentah dan pemberian satu atau lebih label informatif yang memberikan konteks. Dalam konteks *machine learning*, data labelling membantu model untuk memahami dan mempelajari pola dalam data. Proses data labelling diperlukan dalam berbagai bidang, seperti *computer vision, speech recognition*, pemrosesan bahasa alami, dan banyak lagi. Dengan memberikan label pada data, kita membantu model untuk mengenali pola-pola yang ada dalam data tersebut, sehingga model dapat membuat prediksi atau klasifikasi yang lebih akurat. Data labelling juga memainkan peran kunci dalam pelatihan model *machine learning*. (Qorita, 2022).

6. *Text Preprocessing*

Preprocessing adalah langkah awal dalam menyiapkan data mentah sebelum langkah-langkah berikutnya. Proses preprocessing melibatkan penghapusan data yang tidak relevan atau transformasi data menjadi format yang lebih mudah untuk diproses oleh sistem. Dalam konteks analisis sentimen, preprocessing sangat penting, terutama ketika berhadapan dengan data dari media sosial seperti *Twitter*. Media sosial seringkali berisi teks yang

tidak formal, tidak terstruktur, dan memiliki tingkat noise yang tinggi (Syakuro, 2017).

Secara umum proses yang dilakukan dalam tahapan preprocessing adalah sebagai berikut:

a. Case Folding

Case folding adalah proses yang digunakan untuk mengubah huruf besar menjadi huruf kecil dalam sebuah dokumen teks (*lowercase*). Dalam proses ini, semua kata yang mengandung huruf besar diubah menjadi huruf kecil, sehingga membuat teks menjadi lebih konsisten dan memudahkan dalam analisis dan pemrosesan data. *Case folding* juga berguna untuk menghindari perbedaan penulisan yang tidak konsisten yang sering muncul dalam teks, sehingga mempermudah pengolahan data yang lebih lanjut. (Mustofa et al., 2019).

b. Cleansing

Cleansing adalah proses krusial dalam pemrosesan dokumen teks yang bertujuan untuk membersihkan dokumen dari kata-kata yang tidak relevan atau tidak diperlukan. Dalam tahap ini, beberapa komponen yang sering menjadi sumber noise dan tidak berperan dalam analisis sentimen akan dihilangkan. Komponen yang dihapus mencakup

karakter HTML, kata kunci, ikon emosi, hashtag (#), username (@username), URL (http://situs.com), dan alamat email (nama@situs.com). Menghilangkan komponen-komponen tersebut dapat meningkatkan kualitas data teks dan membantu proses analisis sentimen yang lebih akurat dan efisien. (Faradhillah, 2016).

c. Remove Duplicate

Remove duplicate yaitu proses penghapusan data berulang untuk menghilangkan data tweet yang memiliki opini yang sama.

d. Tokenization

Tokenization merupakan proses menguraikan teks yang awalnya berupa kalimat-kalimat menjadi beberapa bagian-bagian kata. Tekonisisasi adalah proses pemotongan string berdasarkan kata-kata yang membentuknya. Proses ini mengandalkan karakter spasi dalam teks dokumen untuk melakukan pemisahan antara kata-kata. Proses ini akan menghilangkan *whitespace* (Hakim, 2021).

e. Normalization

Normalisasi adalah langkah penting dalam proses pemrosesan teks yang bertujuan untuk memastikan konsistensi kata-kata dalam dokumen sesuai dengan

aturan bahasa yang berlaku. Pada tahap ini, dilakukan substitusi atau koreksi kata-kata, terutama untuk mengatasi kata-kata yang merupakan singkatan atau memiliki kesalahan ejaan. Kata-kata yang bersifat tidak baku atau singkatan akan diubah menjadi bentuk kata baku sesuai dengan pedoman yang terdapat dalam Kamus Besar Bahasa Indonesia (KBBI). Normalisasi ini diperlukan untuk memastikan konsistensi dan keseragaman dalam teks yang akan diolah, sehingga lebih mudah diproses oleh algoritma analisis.

f. Stopword Removal

Stopword removal adalah proses penyaringan dengan menghilangkan kata-kata yang memiliki banyak penggunaannya tetapi tidak mempengaruhi sentimen suatu kalimat atau tidak memiliki arti penting. Proses tersebut dilakukan dengan cara membandingaknnya dengan *stopword list* yang ada. Contoh dari stopword misalnya, kata sambung, artikel dan preposisi (Nomleni, 2015).

g. Stemming

Stemming adalah metode pengambilan kata dasar dari kata-kata yang telah mengalami imbuhan dan kata perulangan, dengan asumsi bahwa kata-kata tersebut memiliki makna yang setara. Algoritma ini beroperasi

berdasarkan struktur morfologi Bahasa Indonesia, yang mencakup awalan, akhiran, sisipan, serta kombinasi awalan dan akhiran. (Hakim, 2021). Tahap stemming memiliki dua tujuan utama yaitu untuk meningkatkan efisiensi dengan mengurangi jumlah kata dalam dokumen, sehingga dapat menghemat ruang penyimpanan dan mempercepat proses pencarian. Selain itu juga untuk meningkatkan efektivitas dengan mengurangi variasi kata, yang pada gilirannya dapat mengurangi recall. Contoh dari efektivitas ini adalah mengubah kata-kata seperti "duduk-lah" menjadi "duduk," "minum-lah" menjadi "minum," "jika-pun" menjadi "jika," dan sebagainya.

Akurasi *stemming* adalah metrik yang mengukur sejauh mana algoritma *stemming* berhasil dalam menghasilkan kata dasar dari berbagai bentuk kata. Berdasarkan tujuan proses *stemming* yang dilakukan, pengukuran akurasi *stemming* memiliki fungsi untuk mengevaluasi seberapa baik algoritma dapat memahami struktur kata dan menghasilkan kata dasar yang konsisten dan benar. Tingkat keberhasilan algoritma dalam mengidentifikasi kata-kata dasar dapat digunakan untuk menilai kualitas algoritma dan memperbaikinya jika diperlukan. Sehingga akurasi

stemming menjadi penting dalam mengukur kualitas dan efektivitas algoritma *stemming* dalam pemrosesan teks.

7. *Term Frequency-Inverse Document Frequency*

Menurut Amalia & Yustanti (2021), tahapan penting dalam pengolahan data teks adalah ekstraksi fitur, karena komputer secara prinsip hanya mampu mengolah data numerik. Ekstraksi fitur memungkinkan representasi kata-kata dalam bentuk vektor, dan salah satu teknik yang umum digunakan adalah pembobotan Term Frequency-Inverse Document Frequency (TF-IDF). Pembobotan ini adalah metode algoritma yang memberikan bobot pada teks, yang tiap katanya diberikan bobot (Fikri et al., 2020). Konsep pembobotan Term Frequency (TF) melibatkan perhitungan frekuensi munculnya kata dalam satu dokumen, sedangkan Document Frequency (DF) adalah jumlah dokumen yang memuat kata tersebut. Inverse Document Frequency (IDF) adalah nilai kebalikan dari Document Frequency (DF). Metode ini menghasilkan matriks output yang berisi nilai dan kata-kata unik (Fridayati, 2023). Perhitungan bobot

tiap token t di dokumen dirumuskan dalam persamaan 2.1 sebagai berikut:

$$W_{dt} = TF_{dt} * IDF_t \quad (2.1)$$

Keterangan :

W_{dt} yaitu bobot dari kata (t) dalam satu dokumen (d)

tf_{dt} yaitu frekuensi kemunculan kata dalam dokumen

IDF_t yaitu *Inversed Document Frequency*

Berdasarkan persamaan di atas, untuk menghitung frekuensi kemunculan kata dalam dokumen (tf_{dt}) dapat dihitung dengan persamaan 2.2 berikut:

$$TFdt = \frac{jumlah\ kata\ (t)\ pada\ dokumen\ (d)}{total\ kata\ (t)\ pada\ dokumen\ (d)} \quad (2.2)$$

Nilai *IDF* (*Inversed Document* Frequency) merupakan jumlah dokumen yang mengandung kata tersebut yang didapatkan dari perhitungan dalam persamaan 2.3 sebagai berikut: (Zidan, 2022)

$$IDFt = \log \frac{total\ dokumen}{jumlah\ dokumen\ yang\ mengandung\ kata\ (t)} \quad (2.3)$$

Setelah mendapatkan bobot (W) untuk setiap kata dalam dokumen, langkah selanjutnya adalah penyusunan berdasarkan urutan bobot. Semakin besar nilai bobotnya, semakin tinggi tingkat kesamaan dokumen terhadap kata kunci tersebut, dan sebaliknya (Fridayati, 2023).

8. *Split Validation Data*

Split Validation Data adalah Teknik data dibagi secara acak menjadi dua bagian, bagian pertama digunakan untuk melatih model bisa disebut data latih(*training*) dan bagian kedua digunakan untuk menguji model bisa disebut data uji(*testing*) (Turmudi Zy et al., 2021). Data *testing* adalah iterasi dari setiap kelompok. Data *training* adalah kelompok yang tidak berperan di dalam data uji. Data latih berperan sebagai model dalam proses ini, dan data uji digunakan untuk mengklasifikasikan data, serta menghitung akurasi model atau kesalahan berdasarkan model yang telah dilatih (Meilany, 2022). Dalam penelitian ini, dataset akan dipartisi dengan 4 (empat) perbandingan data latih dan data uji yaitu 90:10, 80:20, 70:30, dan 60:40

9. *Naïve Bayes Classifier*

Algoritma *Naive Bayes* merupakan metode machine learning yang berasal dari teorema Bayes yang ditemukan oleh Thomas Bayes pada abad ke-18 (Suyanto, 2018). *Naive Bayes Classifier* memanfaatkan perhitungan sederhana berdasarkan teorema Bayes, yang memungkinkan komputasi yang lebih efisien melalui perkalian probabilitas (Arifin & Sasongko,

2018*). Naive Bayes* adalah pendekatan sederhana namun sangat popular dalam pengklasifikasian probabilitas. *Naive Bayes Classifier* (NBC) merupakan metode klasifikasi probabilitas yang simple, berdasarkan teorema Bayes. Dalam *Naive Bayes Classifier*, teorema Bayes digunakan dengan asumsi tingkat independensi yang tinggi (Meilany, 2022; Hakim, 2021; Zidan, 2022).

Dalam analisis sentimen, metode klasifikasi *Naive Bayes Classifier* digunakan untuk melakukan prediksi dalam suatu kasus berdasarkan hasil klasifikasinya (Raharjo et al., 2022). Metode klasifikasi Naive Bayes mengaplikasikan teori Bayes dengan asumsi bahwa atribut objek adalah saling bebas. Dalam proses klasifikasi, metode ini membandingkan probabilitas kemunculan kata kunci dalam data latih dan data uji. Hal ini memungkinkan penentuan keberadaan atau ketiadaan fitur dalam kelas yang tidak berhubungan. Proses klasifikasi dengan Naive Bayes memerlukan data pelatihan awal sebagai dasar pengambilan keputusan. Probabilitas yang dihitung dalam metode ini berkaitan dengan kemunculan kata-kata dalam dokumen (Sodik and Kharisudin, 2021). *Naive Bayes Classifier* dapat digunakan untuk

mengklasifikasi sebuah opini positif maupun negatif. Hasil dari penelitian-penelitian yang ada menunjukkan efektivitas metode *Naive Bayes Classifier* sebagai salah satu metode yang paling baik untuk pelatihan. Selain itu hasil klasifikasinya juga memiliki tingkat akurasi yang baik (Ambasador Flores et al. 2020)

Persamaan 2.4 dari teorema Bayes adalah sebagai berikut (Darwis et al., 2021)

$$P\ (\ Y\ |\ X\ )\ =\ \frac{P\ (X|Y).\ P(Y)}{P\ (X)} \qquad (2.4)$$

Berikut uraian persamaan 2.4 sebagai berikut :

Y = Hipotesis data X dari kelas yang spesifik

X = Data dengan kelas yang belum diketahui

P(Y|X) = Probabilitas hipotesis H berdasarkan kondisi X (posterior)

P(X|Y) = Probabilitas X berdasarkan kondisi tersebut (Likelihood)

P(Y) = Probabilitas hipotesis H (prior)

P(X) = Probabilitas dari X (Evidance)

Kunci perbedaan antara teorema Bayes dan metode lainnya adalah bahwa dalam teorema ini, parameter Bayes dianggap sebagai variable acak, sementara dalam statistic klasik, parameter dianggap tidak dapat diketahui. Teorema Bayes merujuk pada hubungan peluang bersyarat antara kejadian H dan X.

Hasilnya, tingkat akurasi yang dicapai oleh metode ini lebih tinggi dibandingkan dengan metode lainnya(Sains, 2019).

10. *Support Vector Machine*

Support Vector Machine (SVM) adalah metode pembelajaran yang menganalisis data dan mengenali pola yang digunakan untuk klasifikasi serta merupakan model yang berasal dari teori pembelajaran statistika (luthfanida, 2022). *Support Vector Machine* (SVM) adalah salah satu algoritma klasifikasi yang umum digunakan dalam analisis sentimen. Tujuan dari metode SVM adalah menemukan *hyperplane* yang paling optimum, yaitu *hyperplane* yang berada di tengah-tengah kedua kelas dengan jarak paling jauh ke data-data terluar di kedua kelas (Suyanto,2018 ; Sodik et al. 2021). Dalam metode ini, langkah awal adalah mencari hyperplane terbaik atau batas keputusan yang dapat memisahkan dua kelas di ruang input. Contohnya, dalam hal ini, metode bertujuan untuk memisahkan tweet yang bersentimen positif dari yang bersentimen negatif. Proses pencarian nilai hyperplane melibatkan penggunaan vektor pendukung dan perhitungan margin. Hyperplane merujuk pada bidang

pemisah antara dua kelas yang berbeda. Sedangkan margin adalah jarak antara *Support Vector Machine* dan hyperplane tersebut (Santoso, 2021).

Manualisasi perhitungan klasifikasi dalam model SVM, diperlukan menentukan parameter yang akan digunakan. Dalam proses pembuatan model, parameter yang digunakan akan berbeda sesuai dengan jenis kernel yang digunakan. Kernel linear SVM merupakan fungsi kernel yang baik digunakan ketika data sudah terpisah secara linear. Dalam kasus model SVM dengan kernel linear, untuk perhitungan prediksi dilakukan dengan menggunakan persamaan berikut.(Santoso, 2021)

$$[(w^T.x_i) + b\ ] \geq 1 \text{ untuk } y_i = +1\ (positif) \quad (2.5)$$

$$[(w^T.x_i) + b\ ] \leq 1 \text{ untuk } y_i = -1\ (negatif) \quad (2.6)$$

Berikut uraian persamaan 2.5 dan 2.6 di atas :

W = vektor bobot (weight)

x_i = nilai fitur ke – *i*

b = nilai bias

Perhitungan nilai bobot (weight) adalah vektor yang menentukan orientasi *hyperplane* (batas keputusan) dalam ruang fitur. Sedangkan nilai bias (b) adalah konstanta yang menentukan posisi *hyperplane*

dalam ruang fitur. Dalam model SVM menggunakan kernel linear, bobot (w) diperoleh dengan persamaan 2.7 dan nilai bias (b) diperoleh dari persamaan 2.8 berikut. (Taufik dan Pamungkas, 2018)

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad (2.7)$$

$$b = -\frac{w.x_i + w.x_j}{2} \qquad (2.8)$$

Berdasarkan uraian persamaan 2.7 dan 2.8 di atas dimana α_i adalah koefisien Lagrange dan y_i adalah label kelas, serta x_i dan x_j adalah dua titik pada batas keputusan. Dengan menggunakan nilai bobot dan bias yang optimal, SVM dapat menghasilkan batas keputusan yang efektif dan memisahkan kelas-kelas dengan baik dalam ruang fitur.

Support Vector Machine (SVM) mencari hyperplane terbaik yang terletak di tengah-tengah pembatas kelas, dan memaksimalkan margin atau jarak antara dua kumpulan objek yang berbeda kelas. Metode *Quadratic Programming* (QP) untuk mengatasi permasalahan tersebut menggunakan fungsi *Dualitis Langrage Multipier* sebagai berikut. (Santoso, 2021)

$Max\ L_D = \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j\ K\ (x_i, x_j)$

Syarat : $0 \leq \alpha_i \leq C$ dan $\sum_{i=1}^{N} \alpha_i y_i = 0$

SVM memiliki beberapa keunggulan, seperti kemampuan bekerja dengan baik pada data, kemampuan untuk mencapai solusi optimal, rasio konvergensi yang rendah, dan kecenderungan untuk menghindari kesulitan dalam masalah. Di sisi lain, kelemahan SVM terletak pada tantangan dalam pemilihan parameter yang cocok, yang dapat memengaruhi tingkat akurasi pada algoritma SVM. Studi yang dilakukan oleh Kristiyanti et al. pada tahun 2019 menunjukkan bahwa SVM memiliki tingkat akurasi yang lebih tinggi dibandingkan dengan Naive Bayes (Meilany, 2022).

11. Evaluasi

Dalam menilai akurasi data, validasi model menjadi suatu langkah penting. Evaluasi bertujuan untuk menilai kegunaan dari model yang telah dipelajari sebelumnya. Ini dicapai dengan perbandingan melalui penggunaan Confusion Matrix, yang membagi dataset menjadi dua kelas yaitu positif dan negatif (Widowati dan Sadikin, 2021).

Dalam penelitian ini, data *tweet* dikategorikan menjadi 2 (dua) kategori sentimen yaitu sentimen positif dan negatif. Maka dari itu, pada penelitian ini

proses evaluasi dihitung menggunakan metode *Confusion matrix 2x2. Confusion matrix* atau sering disebut sebagai *error matrix*, adalah metode umum yang digunakan untuk mengevaluasi kinerja model klasifikasi. Confusion matrix memberikan informasi tentang seberapa sering perilaku tertentu diprediksi dengan benar dan seberapa sering perilaku tersebut salah diklasifikasikan sebagai perilaku lain. Tabel confusion matrix dapat digambarkan seperti yang ditunjukkan pada Tabel 2.1.

Tabel 2. 1 *Confusion Matrix*

Nilai Prediksi	**Nilai Aktual**	
	Positive	*Negative*
Positive	*True Positive*	*False Positive*
Negative	*False Negative*	*True Negative*

Tabel di atas adalah confusion matrix dengan penjelasan sebagai berikut:

1. TP (*True Positive*) : nilai aktual yang bernilai positif dan terprediksi benar
2. FP (*False Positive*) : nilai aktual yang bernilai negatif dan terprediksi positif
3. TN (*True Negative*) : nilai aktual yang bernilai negatif dan terprediksi benar

4. FN (False Negative) : nilai aktual yang bernilai positif dan terprediksi negatif

Hasil dari metode *Naive Bayes* dan *Support Vector Machine* dievaluasi menggunakan *Confusion Matrix* untuk menghitung akurasi, *precision, recall, f1 Score* (Meilany, 2022)

Akurasi adalah suatu formula untuk memprediksi sejauh mana kesesuaian (True) dengan nilai aktual dari keseluruhan data tersebut. Untuk menghitung nilai akurasi dapat dilihat pada persamaan 2.5 berikut.

$$akurasi = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \qquad (2.5)$$

Precision adalah ukuran yang mengindikasikan tingkat ketepatan sistem dalam mengidentifikasi data positif dan negatif dengan benar. Nilai presisi diperoleh dari perbandingan jumlah hasil positif yang diprediksi dengan jumlah data aktual yang positif. Perhitungan nilai *precision* dapat dilihat pada persamaan 2.6 berikut.

$$precison = \frac{TP}{TP+FP} \qquad (2.6)$$

Recall adalah suatu rumus dalam mengukur prediksi True Positive dengan nilai keseluruhan data yang diprediksi negatif. Perhitungan *recall* dihitung dengan persamaan 2.7 berikut.

$$recall = \frac{TP}{TP+FN} \qquad (2.7)$$

F1 Score adalah metrik yang menggabungkan nilai presisi (precision) dan recall menjadi satu angka tunggal. Ini memberikan perbandingan rata-rata antara presisi dan recall, dan berguna untuk mengukur performa suatu model klasifikasi. Semakin tinggi nilai *F1 score*, semakin baik performa modelnya dalam mengklasifikasikan data. Persamaan 2.8 digunakan untuk menghitung nilai *f1 score*

$$f1\ score = \ 2\ x\ \frac{Precision\ x\ Recall}{Precision+Recall} \qquad (2.8)$$

Setelah mendapatkan performa dari model klasifikasi *Naive Bayes Classifier* dan *Support Vector Machine*, peneliti menetapkan ambang batas performa dengan nilai AUC (*Area Under Curve*) untuk mengukur perbedaan performansi metode yang digunakan. Pada penelitian ini nilai AUC yang digunakan diberi nilai dalam bentuk angka dengan menggunakan perhitungan persentase (1-100%). Nilai AUC dapat dibagi menjadi beberapa kelompok yaitu (Nuari,2018).

a. Akurasi bernilai 90 – 100 % = klasifikasi sangat baik
b. Akurasi bernilai 80 – 90 % = klasifikasi baik
c. Akurasi bernilai 70 – 80 % = klasifikasi cukup
d. Akurasi bernilai 60 – 70 % = klasifikasi buruk
e. Akurasi bernilai 50 – 60 % = klasifikasi salah

B. Kajian Penelitian Yang Relevan

Adapun sumber penelitian terdahulu sebagai rujukan untuk memperkuat penelitian ini berupa skripsi maupun jurnal yang membahas tentang analisis sentimen menggunakan algoritma *Naive Bayes* dan *Support Vector Machine* (SVM).

Tabel 2. 2 *Kajian Pustaka*

No	Penulis (tahun)	Judul	Hasil Penelitian
1	Muhammad Zidan (2022)	Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial Twitter Di Indonesia Menggunakan Metode Naïve Bayes	Pada penelitian ini menganalisis sentimen kenaikan harga BBM dengan dataset 1483 tweet yang diambil dari Twitter kemudian telah melalui tahap *preprocessing* dan mengklasifikasikannya menjadi 3 yaitu positif, netral dan negatif dengan perbandingan data latih dan data uji 80:20 yang diambil acak. Hasil analisis sentimen dengan metode Naïve Bayes

			didapatkan nilai performa dengan tingkat akurasi sebesar 81%, presicion sebesar 83%, recall sebesar 81% dan f1 score sebesar 79%. Serta didapatkan sentimen negatif memiliki nilai persentase tertinggi.
2	Muhammad Hadiyan Rasyadi (2017)	Analisis Sentimen Pada Twitter Menggunakan Metode Naïve Bayes (Studi Kasus Pemilihan Gubernur DKI Jakarta 2017)	Penelitian ini menganalisis prediksi sentimen masyarakat kepada pasangan calon Gubernur dan Wakil Gubernur DKI Jakarta putaran kedua menggunakan Naïve Bayes dengan membagi sentiment menjadi 3 kelas yaitu positif, negative dan netral. Pelabelan dilakukan manual dengan Data yang

			diambil menggunakan keyword @AhokDjarot sebanyak 400 data tweet sebagai data latih dan @JktMajubersama 200 data tweet sebagai data uji dengan metode naïve bayes memperoleh predikdi hasil setiap data pada data uji memiliki akurasi 60,60%. Hasil dari sensitifitas terbesar pada kelas positif, disusul kelas negatif dan netral. Sedangkan untuk spesifisitas terdapat pada kelas netral, lalu kelas negatif dan positif.
3	Gading Teguh	Analisis Sentimen Pada Tweet Dengan Tagar	Pada penelitian ini menganalisis sentimen pada tweet

	Santoso (2021)	#BPJSRASARENTENIR Menggunakan Metode Support Vector Machine (SVM)	menggunakan SVM dengan mengklasifikasikan sentimen kedalam kelas positif atau negatif dengan menggunakan 300 data dengan 200 sebagai data latih dan 100 data sebagai data uji, mendapatkan hasil tingkat akurasi sentimen terhadap tagar #BPJSrasarentenir adalah 94%. Hasil dari analisis ini menyatakan bahwa sentimen masyarakat tentang #BPJSrasarentenir lebih cenderung beranggapan negatif.
4	Sulton Nur Hakim (2021)	Analisis Sentimen Persepsi Pengguna MYINDIHOME	Penelitian ini dilakukan dengan 2539 ulasan sebagai

		Menggunakan Metode Support Vector Machine (SVM) Dan Naïve Bayes Classifier (NBC)	sampelnya. Hasil pelabelan setiap setimen dianalisis didapatkan jumlah sentiment positif 1379 ulasan(54,3%) dan negative sebanyak 1160 (46,7%). Berdasarkan tiga perbandingan data training dan testing yaitu 70%:30%, 80%:20%, 90%:10% dilakukan percobaan sebanyak lima kali setiap perbandingan menggunakan metode NBC dan SVM. Hasil percobaan didapatkan rata-rata akurasi dengan metode SVM sebesar 86,54% dan dengan metode NBC sebesar 84,69%.

5	Nur Adinda Salsabila (2022)	Analisis Sentimen Pada Media Sosial Twitter Terhadap Tokoh Gus Dur Menggunakan Metode Naïve Bayes Dan Support Vector Machine (SVM)	Penelitian ini menganalisis sentimen tokoh Gus Dur dengan algoritma Naïve Bayes dan SVM. Pada klasifikasi metode Naïve Bayes didapatkan 71 sentimen positif dan 31 negatif, sedangkan pada SVM diperoleh 86 sentimen positif dan 16 sentimen negatif. Pada penelitian ini dengan rasio test dan train 30:70% didapat pengguna twitter lebih banyak memberi opini positif untuk topik tokoh Gus Dur karena dilihat dari nilai akurasi yang didapatkan pada Naïve Bayes 78,36% dan SVM 84,27%

			sehingga metode yang diambil adalah SVM .
6	Arfina Shella Meilany (2022)	Analisis Sentimen Opini Masyarakat Pengguna Twitter Terhadap Pariwisata Lampung Menggunakan Support Vector Machine dan Naïve Bayes	Dalam penelitian ini, dengan membandingkan algoritma Support Vector Machine dan Naive Bayes untuk menganalisis sentimen data dari Twitter terkait Pariwisata Lampung ke dalam kelas positif dan negative dengan menggunakan perbandingan data latih dan data uji 90:10% dilakukan berulang sebanyak 10 kali. Hasil penelitian dengan menggunakan SVM menunjukkan hasil kinerja yang lebih baik dengan nilai akurasi sebesar 94,11% dibandingkan

			dengan Naive Bayes yang memiliki nilai akurasi sebesar 93,78% .
7	Rizki Anom Raharjo, I Made Gede Sunarya, Dewa Gede Hendra Divayana (2022)	Perbandingan Metode Naïve Bayes Classifier dan Support Vector Machine Pada Kasus Analisis Sentimen Terhadap Data Vaksin Covid-19 Di Twitter	Penelitian mengukur kinerja algoritma Naïve Bayes Classifier dan Support Vector Machine (SVM) pada analisis sentimen data vaksin Covid-19 menggunakan data tweet periode 1 April 2021 sampai 31 Agustus 2021 dengan keyword Vaksin Covid-19 yaitu berjumlah 1106 data akan dibagi data latih dan data testing 80:20. Hasil penelitian dengan mengklasifikasikan data ke dalam kelas positif dan negatif mendapatkan hasil

			untuk metode *Naive Bayes* accuracy sebesar 81%, precission sebesar 80%, recall sebesar 99%, dan f1_score sebesar 89%. Sedangkan metode SVM accuracy sebesar 87%, precission sebesar 88%, recall sebesar 96%, dan f1_score sebesar 92%.
8	Hizkia Yotant Pradana. Isnandar Slamet, Etik Zukhronah (2022)	Analisis Sentimen Kinerja Pemerintahan Menggunakan Algoritma NBC, KNN, dan SVM	Penelitian ini mengklasifikasikan sentimen masyarakat Indonesia mengenai kinerja pemerintahan dengan 5874 data twitter yang dibagi menjadi data latih dan data uji 70%:30% serta sentimen diberi label positif dan negatif. Hasil penelitian

			menunjukkan bahwa klasifikasi dengan algoritma SVM lebih unggul dengan nilai akurasi 85,47%. Disusul algoritma NBC dengan akurasi 80,24% dan KNN dengan akurasi 75,37%.
9	Yusuf Ansori, Khadijah Fahmi Hayati Holle (2022)	Perbandingan Metode Machine Learning dalam Analisis Sentimen Twitter	Pada penelitian ini dilakukan perbandingan 4 algoritma klasifikasi yaitu Support Vector Machine, K-Nearest Neighbor, Naïve Bayes Classifier, dan Logistic Regression. Data penelitian diambil dari twitter dengan memberikan 2 label pada sentimen yaitu positif dan negatif. Hasil penelitian menunjukkan urutan

			nilai akurasi dari yang paling tinggi yaitu svm, nbc, k-nn, dan logistic regression.
10	Dewi Nurmalasari, Teguh Iman Hermanto, Imam Ma'ruf Nugroho (2023)	Perbandingan Algortima SVM, KNN dan NBC Terhadap Analisis Sentimen Aplikasi Loan Service	Penelitan ini dilakukan untuk menganalisis perbandingan akurasi dari algoritma SVM, K-NN, dan NBC dalam analisis sentimen terhadap aplikasi kredit digital (Loan Service) yaitu Kredivo, Akulaku dan Indodana dengan membaginya menjadi dua kelas, positif dan negatif. Data keseluruhan dari review pengguna aplikasi di google playstore dan total diperoleh sebanyak 42.174 kemudian data dibagi menjadi 70 data latih dan 30 data uji.

			Dalam hasil penelitiannya pada setiap aplikasi yang diteliti algoritma SVM memiliki nilai akurasi lebih tinggi dibandingkan lainnya. Algoritma yang lebih baik selanjutnya adalah NBC lalu diikuti algoritma K-NN.

Berdasarkan penelitian yang telah dilakukan beberapa penulis di atas yang dapat diajukan acuan dalam melakukan penelitian. Hasil penelitian terdahulu menunjukkan performansi penggunaan metode *Support Vector Machine* dan *Naive Bayes Classifier* yang menghasilkan performansi yang cukup baik. Namun, belum ada penelitian yang menerapkan metode *Support Vector Machine* (SVM) dan *Naive Bayes Classifier* dalam analisis sentiment mengenai pandangan atau opini masyarakat terhadap polusi udara di Jakarta. Maka dari itu, peneliti tertarik untuk melakukan penelitian mengenai komparasi algoritma *Support Vector Machine* (SVM) dan *Naive Bayes Classifier* (NBC) terhadap opini Masyarakat mengenai polusi udara di Jakarta pada media sosial *Twitter*.

BAB III

METODOLOGI PENELITIAN

A. Metode Pengumpulan Data

Adapun teknik untuk pengumpulan data adalah sebagai berikut :

a. Pengamatan (Observasi)

Observasi yaitu metode pengumpulan data dengan cara mengadakan tinjauan secara langsung ke objek yang diteliti. Data primer yang diperoleh penulis pada metode ini didapatkan secara langsung dengan pengamatan terhadap segala aktivitas yang sedang *trending topic* di media sosial *Twitter* tentang polusi udara Jakarta.

b. Studi Pustaka

Untuk mendapatkan data-data sekunder yang bersifat teoritis maka penulis melakukan pengumpulan materi dengan memanfaatkan buku, jurnal, skripsi, website dan sejenisnya yang berhubungan dengan masalah yang diteliti oleh penulis baik teori-teori dasar untuk melakukan analisis maupun dokumen laporan yang berhubungan dengan masalah tersebut.

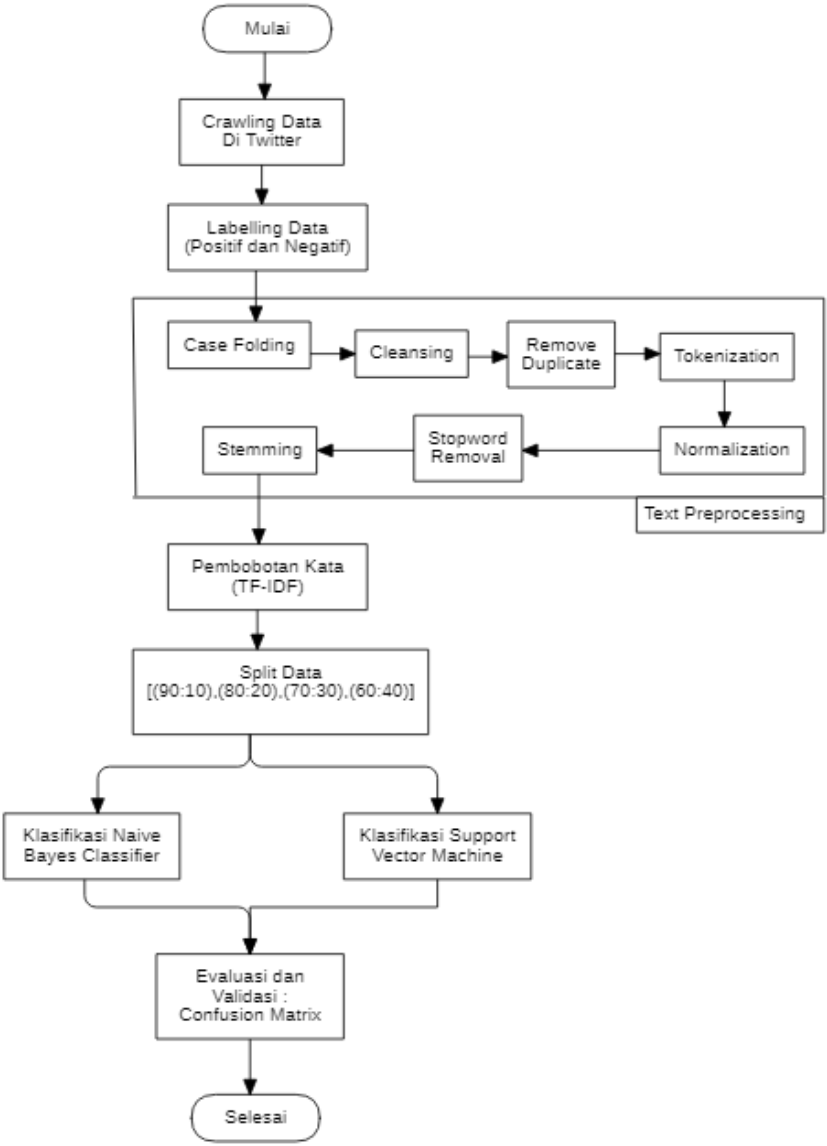
B. Perangkat Penelitian

Spesifikasi perangkat yang digunakan pada penelitian ini membutuhkan beberapa perangkat keras dan perangkat lunak yang harus dipenuhi sebagai alat pendukung dalam berjalannya proses sistem. Beberapa spesifikasi kebutuhan yang diuraikan dalam proses sistem ini sebagai berikut:

1. Spesifikasi kebutuhan perangkat keras (*hardware*)
   a. Device : Laptop ACER
   b. Processor : Intel Celeron N4120 CPU @1.10GHz
   c. Memori (RAM) : 4,00 GB
   d. Monitor : 14 inch
2. Spesifikasi kebutuhan perangkat lunak (*software*)
   a. Sistem Operasi : Windows 11 Home Single Language 64bit
   b. Bahasa Pemrograman : Python
   c. Ms. Office : Ms. Word, Ms. Excel 2019
   d. Google Drive : Google Colab

C. Alur Penelitian

Adapun tahapan pengerjaan penelitian yang membahas mengenai gambaran umum alur penelitian yang dilakukan peneliti bisa dilihat pada gambar 3.1 berikut (Wenty, 2019).

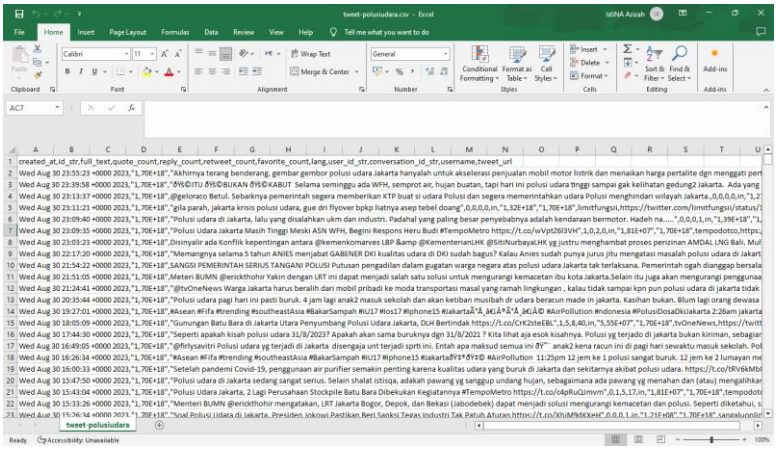


Gambar 3. 1 Alur Penelitian

Berdasarkan *flowchart* penelitian berikut merupakan uraian alur pengerjaan penelitian di atas sebagai berikut.

1. Crawling Data Di *Twitter*

   Proses pengambilan data dari media sosial *Twitter* dilakukan menggunakan teknik crawling dengan bantuan tweet harvest. Library harvest biasanya berisi alat-alat dan fungsi yang memudahkan pengembangan perangkat lunak untuk mengambil dan mengolah data dari berbagai sumber dengan cara yang lebih terstruktur dan efisien. Tweet harvest dapat dilakukan menggunakan API *Twitter* untuk mengumpulkan tweet dari *Twitter* dalam volume besar, dimana proses ini disesuiakan dengan kata kunci yang ditentukan. Pada penelitian ini pengambilan tweet disesuiakan dengan kata kunci "polusi udara jakarta" mendapatkan data sebanyak 510 tweet yang diambil dari periode 1 Agustus 2023 sampai 31 Agustus 2023. Data teks dari *Twitter* dengan proses menggunakan library harvest dengan built menggunakan Node.js yang hasilnya akan dikonversikan menjadi file csv. Berikut hasil *crawling data*



Gambar 3. 2 Data Polusi Udara Jakarta

2. Labelling Data

Proses selanjutnya adalah proses pelabelan data. Setelah melakukan *crawling data* lalu didapatkan file dalam bentuk csv. Pada penelitian ini proses pelabelan dilakukan secara manual oleh ahli bahasa untuk mengklasifikasikan tweet tersebut menjadi dua sentimen yaitu positif dan negatif. Pelabelan data menjadi dua sentimen didasarkan dengan kriteria sebagai berikut.

a. Sentimen Positif

Data tweet yang termasuk ke dalam klasifikasi sentimen positif yaitu data tweet yang menyampaikan pandangan, perasaan positif

terkait penanganan pemerintah terhadap polusi udara. Contoh dari sentimen positif dapat berupa pendapat mengenai langkah pemerintahan atau kebijakan dalam menangani masalah polusi udara di Jakarta.

b. Sentimen Negatif

Data tweet yang tergolong dalam sentimen negatif yaitu tweet yang menyampaikan pendapat tentang ketidakpuasan, kekhawatiran, menjelekkan dan berisi komentar negatif terkait polusi udara di Jakarta. Contoh dari sentimen negatif dapat berupa dampak buruk atau ketidaknyamanan masyarakat mengenai masalah polusi udara.

Berikut contoh proses labelling data menjadi sentimen positif dan negatif ditampilkan pada tabel 3.1 berikut.

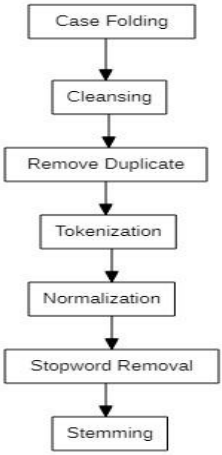
Tabel 3. 1 Contoh labelling data

Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/CYo2wjibsP	Positif

Ajakan #TransportasiUmumLebihEnak! Gunakan transportasi umum untuk mengurangi polusi dan kemacetan. Kita semua bisa berkontribusi, serta himbauan #JanganLupaUjiEmisi kendaraan yang merupakan langkah kecil yang berdampak besar untuk kualitas udara Jakarta. https://t.co/LJCXI1Elcz	Positif
Polusi Udara Mengancam, Kualitas Udara jakarta Belum Penuhi Hak Ekologis Anak! https://t.co/BIyTQ4Dt2W	Negatif
@budibongg Di Jakarta pohon bnyk yg ditebang gedung dimana mana jd polusi udara sgt mengancam	Negatif

3. Text Preprocessing

Preprocessing data adalah tahap ketiga dalam penelitian ini, yang bertujuan untuk mengorganisir data menjadi lebih terstruktur, sehingga mempermudah perhitungan dan tahap penelitian selanjutnya. Tahapan alur *preprocessing* dapat dilihat pada gambar 3.3 berikut.



Gambar 3. 3 Alur Preprocessing

Berikut metode – metode *preprocessing* yang akan dilakukan sebagai berikut:

a. Case Folding

Pada tahap ini, data yang akan diteliti dilakukan penyeragaman bentuk huruf dan penghapusan angka dan tanda baca. Penyeragaman bentuk huruf dilakukan dengan mengubah huruf kapitan menjadi huruf kecil (*lower case*). Berikut hasil contoh data tweet dari diproses case folding.

Tabel 3. 2 *Contoh Case Folding*

Teks Input	Teks Output
Tepat sekali nih gerakan #GotongRoyongBoyong Pohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/CYo2wjibsP	tepat sekali nih gerakan #gotongroyongboyong pohon ini dilakukan. salah satu tindakan nyata yang tepat untuk ngatasi polusi udara jakarta ya ngelakuin gerakan gotong royong boyong pohon begini. ðÿ‘ https://t.co/cyo2wjibsp

b. Cleansing

Pada tahap *cleansing* dilakukan proses pembersihan dokumen dengan menghilangkan karakter HTML, kata kunci, ikon emosi, hashtag (#), username (@username), URL (http://situs.com), dan alamat email (nama@situs.com). Dalam data *Twitter* sering muncul komponen tersebut yang tidak

berpengaruh pada sentimen maka dari itu diperlukan proses *cleansing* untuk menghilangkan atribut tidak berpengaruh dan menggantinya dengan karakter spasi agar dapat mengurangi *noise.* Contoh proses *cleansing*.

Tabel 3. 3 *Contoh Cleansing*

Teks Input	Teks Output
Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/CYo2wjibsP	tepat sekali nih gerakan pohon ini dilakukan salah satu tindakan nyata yang tepat untuk ngatasi polusi udara jakarta ya ngelakuin gerakan gotong royong boyong pohon begini

c. Remove Duplicate

Dalam data *Twitter*, seringkali terdapat pengguna yang mungkin melakukan spam dengan tujuan mengirimkan tweet opini atau tanggapan secara berulang untuk mendapatkan perhatian. Hal tersebut jika tidak dilakukan pembersihan tentu akan mempengaruhi proses perhitungan, maka tahap *remove duplicate* melakukan penghapusan data berulang dan data yang memiliki opini yang sama. Berikut contoh tweet dengan opini yang sama.

Tabel 3. 4 *Contoh Remove Duplicate*

Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4

Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4

d. Tokenization

Pada tahap ini, terjadi proses menguraikan teks yang awalnya berupa kalimat-kalimat menjadi beberapa bagian-bagian kata yang bertujuan untuk menghilangkan *whitespace*. Contoh proses *tokenization.*

Tabel 3. 5 *Contoh Tokenization*

Teks Input	Teks Output
Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/CYo2wjibsP	['tepat', 'sekali', 'nih', 'gerakan', 'pohon', 'ini', 'dilakukan', 'salah', 'satu', 'tindakan', 'nyata', 'yang', 'tepat', 'untuk', 'ngatasi', 'polusi', 'udara', 'jakarta', 'ya', 'ngelakuin', 'gerakan', 'gotong', 'royong', 'boyong', 'pohon', 'begini']

e. Normalization

Pada tahap *normalization* dilakukan pengubahan kata yang tidak sesuai dengan EYD, sehingga dapat mengurangi hasil sentimen dokumen. Tahap ini dibagi menjadi tiga langkah yaitu konversi kata singkatan, konversi kata baku, dan konversi kata inggris. Pada *Twitter* terdapat ketentuan hanya dapat melakukan posting 280 karakter saja, sehingga banyak pengguna yang menulis dengan kata singkatan agar tulisannya dapat tersampaikan. Contoh berikut merupakan contoh proses mengubah kata singkatan menjadi kata normal.

Tabel 3. 6 *Contoh Normalization*

Teks Input	Teks Output
Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini.	['tepat', 'sekali', 'nih', 'gerakan', 'pohon', 'ini', 'dilakukan', 'salah', 'satu', 'tindakan', 'nyata', 'yang', 'tepat', 'untuk', 'mengatasi', 'polusi', 'udara', 'jakarta', 'ya', 'melakukan', 'gerakan',

ðŸ‘ https://t.co/CYo2wjibsP	'gotong', 'royong', 'boyong', 'pohon', 'begini']

f. Stopword Removal

Stopword removal adalah proses eliminasi kata-kata yang tidak relevan dalam data, bertujuan untuk meningkatkan akurasi hasil. Kata yang dihilangkan dihimpun dalam database kata stopword. Sebagai gambaran dari proses stopword removal berikut contoh tweet pada tahap ini.

Tabel 3. 7 *Contoh Stopword Removal*

Teks Input	Teks Output
Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/CYo2wjibsP	['tepat', 'sekali', 'nih', 'gerakan', 'pohon', 'dilakukan', 'salah', 'tindakan', 'nyata', 'tepat', 'mengatasi', 'polusi', 'udara', 'jakarta', 'ya', 'mengatasi', 'gerakan', 'gotong', 'royong', 'boyong', 'pohon', 'begini']

g. Stemming

Dalam tahap stemming, kata-kata dengan imbuhan diubah menjadi bentuk kata dasar, dan kata-kata yang berulang direduksi menjadi satu kata dasar. *Library Sastrawi* Indonesia merupakan salah satu *library* yang digunakan untuk melakukan *stemming.* Dengan menggunakan library sastrawi Indonesia, proses stemming dapat dilakukan dengan cepat dan efisien dalam bahasa Indonesia.

Berikut adalah aturan yang digunakan dalam proses stemming dalam bahasa Indonesia. Prefiks adalah imbuhan yang ditambahkan di awal kata, seperti "se-", "ke-", "me-", dan lain-lain, contohnya "memasak" yang memiliki prefiks "me-" pada kata dasar "masak". Suffiks adalah imbuhan yang ditambahkan di akhir kata, misalnya "-lah", "-kah", "-pun", dan sebagainya, seperti pada kata "marilah". Konfiks adalah kombinasi dari prefiks dan suffiks yang ditambahkan di awal dan akhir kata, contohnya "perasaan" yang terdiri dari prefiks "per-" dan suffiks "-an" dengan kata dasar "rasa". Infiks adalah imbuhan yang disisipkan di tengah kata, seperti "kemilau" yang terbentuk dari infiks "-em-" dalam kata dasar "kilau". Terakhir,

ada perulangan kata, yang merupakan pengulangan bentuk kata dasar, misalnya "anak-anak"(Zidan, 2022).

Berikut contoh dari proses *stemming* pada salah satu tweet yang dihasilkan.

Tabel 3. 8 *Contoh Stemming*

Teks Input	Teks Output
Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/Cyo2wjibsP	[‘tepat’, ‘sekali’, ‘nih’, ‘gerak’, ‘pohon’, ‘laku’, ‘salah’, ‘tindak’, ‘nyata’, ‘tepat’, ‘atasi’, ‘polusi’, ‘udara’, ‘jakarta’, ‘ya’, ‘atasi’, ‘gerak’, ‘gotong’, ‘royong’, ‘boyong’, ‘pohon’, ‘begini’]

Setelah proses *stemming* dilakukan menggunakan *library sastrawi* Indonesia pada data pengujian, langkah berikutnya adalah membandingkan hasil stemmming dengan label asli atau yang telah diproses secara manual. Setiap kata dalam teks hasil stemming kemudian dibandingkan dengan kata yang telah di-stem secara manual atau kata dalam kamus baku. Jika

kedua versi kata tersebut cocok, maka proses stemming dianggap berhasil dan dihitung sebagai prediksi yang benar.

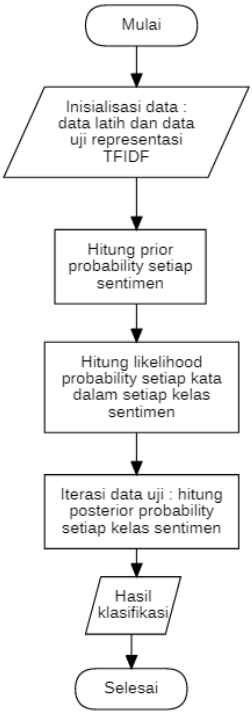
Akurasi stemming kemudian dihitung sebagai proporsi antara jumlah kata yang berhasil di-stem dengan total kata dalam dataset pengujian. Semakin tinggi nilai akurasi stemming, semakin baik kualitas stemming yang dilakukan pada dataset. Akurasi stemming penting untuk mengevaluasi kinerja proses pemrosesan teks, terutama dalam konteks pengelolaan dan analisis teks besar di berbagai aplikasi seperti pengolahan bahasa alami, analisis sentimen, dan klasifikasi teks.

4. Pembobotan TF-IDF

Setelah tahap preprocessing selesai, langkah selanjutnya adalah melakukan proses pembobotan kata. Proses pembobotan kata dengan menggunakan metode TFIDF (Term Frequency Invers Document Frequency). Melalui proses TFIDF, setiap kata dalam setiap kalimat atau dokumen diberi bobot, dan dataset tersebut sudah siap untuk digunakan dalam pelatihan menggunakan metode Naive Bayes dan *Support Vector Machine*.

5. Klasifikasi *Naive Bayes Classifier*

Pada tahap klasifikasi ini, metode Naive Bayes dibagi menjadi dua proses yaitu proses pelatihan (training) dan proses pengujian (testing). Tahap pelatihan dilakukan terlebih dahulu menggunakan dataset training untuk mengestimasi parameter yang dibutuhkan. Selanjutnya, tahap pengujian dilakukan dengan merujuk pada probabilitas yang diperoleh dari dataset pelatihan. Metode Naive Bayes memiliki keunggulan karena memerlukan jumlah data pelatihan yang relatif kecil untuk mengestimasi parameter. Perbedaan mendasar antara teorema Bayes dengan metode lainnya adalah parameter Bayes dianggap sebagai variabel acak, sedangkan dalam statistik klasik, parameter dianggap sebagai nilai yang pasti. Teorema Bayes menjelaskan hubungan antara probabilitas bersyarat dari peristiwa H dan X. Metode ini mencapai tingkat akurasi yang lebih tinggi dibandingkan dengan metode lainnya(Zidan, 2022). Berikut *flowchart* untuk perhitungan klasifikasi menggunakan metode Naive Bayes Classifier ditunjukkan pada gambar 3.4 berikut.

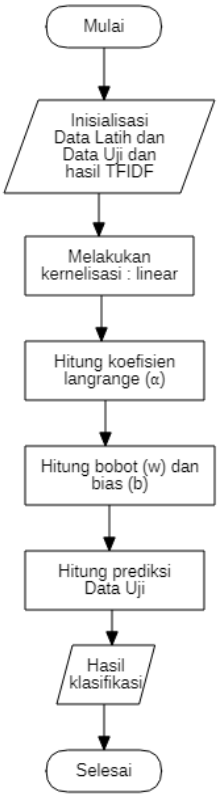


Gambar 3. 4 Flowchart Klasifikasi NBC

6. Klasifikasi *Support Vector Machine*

SVM adalah metode klasifikasi yang dilakukan dengan cara mencari term atau kondisi serupa dengan parameter tertentu agar dapat menentukan kategori akhirnya. Kelebihan menggunakan metode *Support Vector Machine* (SVM) adalah kemampuannya untuk menghasilkan model klasifikasi yang baik dengan formulasi yang jelas dan hanya sedikit parameter yang perlu diatur. Selain itu, SVM mudah dalam

penerapannya karena penentuan SVM dapat ditentukan menggunakan metode Quadratic Programming (QR), dan SVM memiliki kemampuan generalisasi yang tinggi. Alur metode SVM melibatkan langkah-langkah berikut: Melakukan transformasi data ke dalam ruang fitur yang sesuai, mengukur margin antarkelas dalam data, memahami dan menyesuaikan parameter dalam kernel yang digunakan, dan menemukan hyperplane terbaik yang dapat memisahkan kelas-kelas data dengan baik. Dalam proses klasifikasi, sistem hanya memperhatikan titik dan ruang dalam dokumen untuk memodelkan ruang vektor yang digunakan. Setiap kata dalam dokumen yang akan diproses diberikan bobot. Dalam klasifikasi, Support Vector Machine (SVM) bertujuan menemukan garis pemisah terbaik untuk membagi dokumen menjadi dua kelas, dan dokumen uji diklasifikasikan berdasarkan sisi di mana garis tersebut berada. Jika data yang dilatih dapat dipisahkan secara linear, pemilihan hyperplane dari margin dapat dilakukan dengan menghilangkan nilai antara hyperplane dan margin, kemudian memaksimalkan jaraknya(Taufik dan Pamungkas,2018). Berikut tampilan flowchart metode SVM ditunjukkan pada gambar 3.5 berikut.



Gambar 3. 5 Flowchart klasifikasi SVM

7. Evaluasi

Setelah proses pemodelan *Naive Bayes Classifier* dan *Support Vector Machine* selesai dilakukan, langkah selanjutnya adalah menghitung confusion matrix untuk menentukan 4 macam performa machine learning yaitu

akurasi, presisi, recall, dan f1 score yang didapatkan dari analisis sentimen dengan metode *Naive Bayes* dan *Support Vector Machine* terhadap data opini masyarakat mengenai polusi udara jakarta di *Twitter*. Peneliti menghitung hasil metode analisis sentimen kedua metode tersebut, agar dapat mengetahui hasil akurasi yang paling tinggi diantara kedua metode itu.

BAB IV

HASIL DAN PEMBAHASAN

A. Crawling Data Twitter

Proses *Crawling Data* dari media sosial *Twitter* yang sekarang berubah nama menjadi X dilakukan menggunakan tools *tweet-harvest.* Proses ini merupakan proses yang digunakan untuk mengumpulkan dan mengindeks informasi dari berbagai sumber, seperti situs web, *database,* atau dokumen. *Tweet-harvest* adalah salah satu *tools* yang dapat mengumpulkan data pada media sosial *Twitter* yang dapat dijalankan dengan *Application Programming Interface* (API) dan dapat mengakses data dengan memasukkan *user valid auth_token.* Pada penelitian ini dibutuhkan data dari media sosial *Twitter* mengenai sentimen masyarakat mengenai polusi udara yang terjadi Jakarta pada bulan Agustus tahun 2023. Software yang digunakan oleh peneliti dalam melakukan penelitian mengenai analisis sentimen yaitu dengan menggunakan *google colab* atau *colaboratory* sebagai *tools* pendukung yang berbentuk *cloud* dan dapat dijalankan dalam *browser* yang hanya memerlukan satu akun google sudah dapat mengaksesnya untuk menulis, menyimpan, dan membagikan program yang telah ditulis melalui *google drive* .

Proses pengambilan data perlu dilakukan *input user Auth Token* dari sebuah akun *Twitter* yang sebelumnya sudah

dilakukan *Login* terlebih dahulu untuk menggunakan fitur pencarian pada media sosial Twitter. Selanjutnya dilakukan *instalasi tweet-harvest* dengan melakukan import *package* yang diperlukan seperti melakukan instalasi pandas dan *Node.js* untuk menggunakan *tweet-harvest* seperti Gambar 4.1 di bawah ini.

```
# Import required Python package
!pip install pandas

# Install Node.js (because tweet-harvest built using Node.js)
!sudo apt-get update
!sudo apt-get install -y ca-certificates curl gnupg
!sudo mkdir -p /etc/apt/keyrings
!curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

!NODE_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee /etc/apt/source

!sudo apt-get update
!sudo apt-get install nodejs -y

!node -v
```

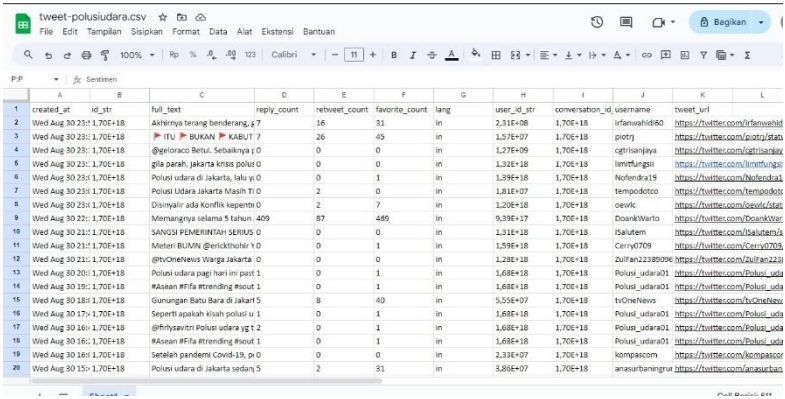
Gambar 4. 1 Install pandas dan node.js untuk menggunakan tweet-harvest

Pada proses *crawling data* dilakukan dengan menggunakan kata kunci "polusi udara jakarta" yang kembali menjadi *trending topic* pada bulan Agustus 2023. Data *tweets* yang diambil mulai dari tanggal 1 Agustus 2023 hingga 31 Agustus 2023. Setelah proses ini dilakukan berdasarkan input yang sudah ditentukan maka data *tweets* yang akan ditampilkan dalam bentuk file *Command Separated Values* (CSV) yang disimpan dalam dataframe yang diberi nama "tweet-polusiudara.csv". Berikut pada Gambar 4.2 *source code* untuk dilakukan proses *crawling data.*

```
filename = 'tweet-polusiudara.csv'
search_keyword = 'polusi udara jakarta until:2023-08-31 since:2023-08-01'
limit = 500

!npx --yes tweet-harvest@2.2.8 -o "{filename}" -s "{search_keyword}" -l {limit} --token {twitter_auth_token}
```

Gambar 4. 2 Source code Crawling Data

Hasil dari *Crawling data* yang dilakukan dengan menjalankan program pada gambar 4.2 di atas akan diperoleh sebuah DataFrame awal yang belum diolah dapat dilihat pada Gambar 4.3 berikut ini.



	A	B	C	D	E	F	G	H	I	J	K	L
1	created_at	id_str	full_text	reply_count	retweet_count	favorite_count	lang	user_id_str	conversation_id	username	tweet_url	
2	Wed Aug 30 23:(	1,70E+18	Akhirnya terang benderang, ş	7	16	31	in	2,31E+08	1,70E+18	irfanwahidi80	https://twitter.com/irfanwahid	
3	Wed Aug 30 23:(	1,70E+18	ITU BUKAN KABUT	7	26	45	in	1,57E+07	1,70E+18	piotrj	https://twitter.com/piotrj/stat	
4	Wed Aug 30 23:(	1,70E+18	@geloraco Betul. Sebaiknya ţ	0	0	0	in	1,27E+09	1,70E+18	cgtrisanjaya	https://twitter.com/cgtrisanjay	
5	Wed Aug 30 23:(	1,70E+18	gila parah, jakarta krisis polus	0	0	0	in	1,32E+18	1,70E+18	limitfungsii	https://twitter.com/limitfungsi	
6	Wed Aug 30 23:(	1,70E+18	Polusi udara di Jakarta, lalu yi	0	0	1	in	1,59E+18	1,70E+18	Nofendra19	https://twitter.com/Nofendra1	
7	Wed Aug 30 23:(	1,70E+18	Polusi Udara Jakarta Masih TI	0	2	0	in	1,81E+07	1,70E+18	tempodotco	https://twitter.com/tempodotc	
8	Wed Aug 30 23:(	1,70E+18	Disinyalir ada Konflik kepentin	0	2	7	in	1,20E+18	1,70E+18	ceowlc	https://twitter.com/ceowlc/stat	
9	Wed Aug 30 22:(	1,70E+18	Memangnya selama 5 tahun :	409	87	469	in	9,39E+17	1,70E+18	DoankWarto	https://twitter.com/DoankWar	
10	Wed Aug 30 21:(	1,70E+18	SANGSI PEMERINTAH SERIUS	0	0	0	in	1,31E+18	1,70E+18	ISalutem	https://twitter.com/ISalutem/s	
11	Wed Aug 30 21:(	1,70E+18	Meteri BUMN @erickthohir Y	0	0	1	in	1,59E+18	1,70E+18	Cerry0709	https://twitter.com/Cerry0709	
12	Wed Aug 30 21:(	1,70E+18	@tvOneNews Warga Jakarta	0	0	0	in	1,28E+18	1,70E+18	ZulFan22389096	https://twitter.com/ZulFan223	
13	Wed Aug 30 20:(	1,70E+18	Polusi udara pagi hari ini past	1	0	1	in	1,68E+18	1,70E+18	Polusi_udara01	https://twitter.com/Polusi_uda	
14	Wed Aug 30 19:(	1,70E+18	#Asean #Fifa #trending #sout	1	0	1	in	1,68E+18	1,70E+18	Polusi_udara01	https://twitter.com/Polusi_uda	
15	Wed Aug 30 18:(	1,70E+18	Gunungan Batu Bara di Jakart	5	8	40	in	5,55E+07	1,70E+18	tvOneNews	https://twitter.com/tvOneNew	
16	Wed Aug 30 17:(	1,70E+18	Seperti apakah kisah polusi u	1	0	1	in	1,68E+18	1,70E+18	Polusi_udara01	https://twitter.com/Polusi_uda	
17	Wed Aug 30 16:(	1,70E+18	@fhrlysavitri Polusi udara yg t	2	0	1	in	1,68E+18	1,70E+18	Polusi_udara01	https://twitter.com/Polusi_uda	
18	Wed Aug 30 16:(	1,70E+18	#Asean #Fifa #trending #sout	1	0	1	in	1,68E+18	1,70E+18	Polusi_udara01	https://twitter.com/Polusi_uda	
19	Wed Aug 30 16:(	1,70E+18	Setelah pandemi Covid-19, pi	0	0	0	in	2,33E+07	1,70E+18	kompascom	https://twitter.com/kompascor	
20	Wed Aug 30 15:(	1,70E+18	Polusi udara di Jakarta sedanj	5	2	31	in	3,86E+07	1,70E+18	anasurbaningrur	https://twitter.com/anasurban	

Gambar 4. 3 Hasil data awal dari proses crawling data

B. Labelling Data Twitter

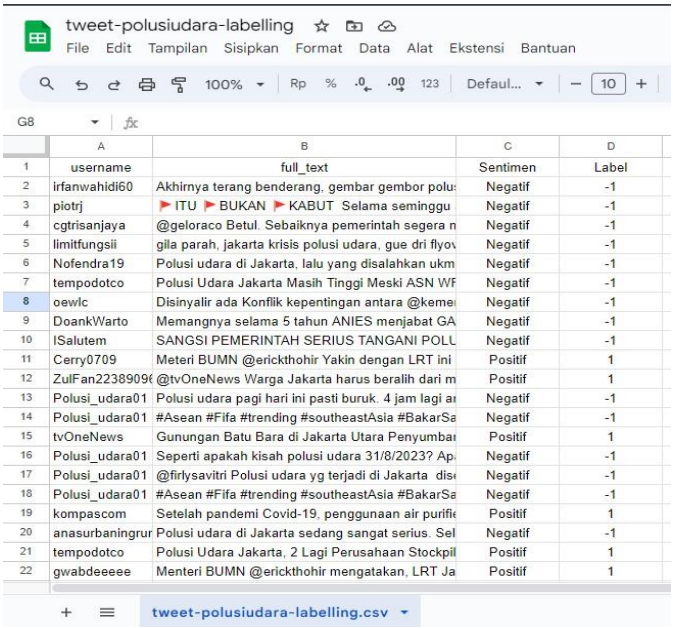
Proses selanjutnya setelah dilakukan proses *crawling data* selesai dan didapatkan data *tweets* yang dibutuhkan untuk penelitian ini yaitu proses pelabelan data *tweet.* Dalam proses ini peneliti melakukan pelabelan sentimen yang dikategorikan menjadi dua kategori yaitu positif dan negatif. Karakteristik dari kedua kategori dapat dijadikan penentu bagi peneliti dalam melakukan pelabelan data tweet ini. Sentimen yang memiliki nilai positif artinya tweet yang menyampaikan pandangan, perasaan positif terkait penanganan polusi udara Jakarta, contohnya berupa pendapat mengenai langkah pemerintah/kebijakan.

Sentimen yang bernilai negatif merupakan tweet yang menyampaikan ketidakpuasan, kekhawatiran, menjelekkan atau berisi komentar negatif, contohnya dapat berupa dampak buruk atau ketidaknyamanan masyarakat terkait kondisi polusi udara. Pada proses pelabelan data, peneliti memberikan label untuk sentimen positif akan diberi angka 1 (positif satu), sentimen negatif diberikan nilai -1 (negatif satu).

Dalam proses labelling data twitter ini setidaknya membutuhkan dua orang atau lebih untuk menghindari perbedaan pendapat yang sama serta menghindari subjektivitas dalam menentukan sentimen tweet yang terdapat dari hasil crawling data. Pada penelitian ini proses labelling data dilakukan dengan tiga orang yaitu mahasiswa dari jurusan Pendidikan Bahasa dan Sastra Indonesia. Proses pelabelan data dengan jumlah data sebanyak 510 tweet yang dilakukan secara manual oleh tiga pemberi label akan menimbulkan perbedaan pendapat dalam pengklasifikasian sentimen karena pelabelan tersebut berdasarkan persepsi dari pemberi label. Hasil dari proses labelling yang dilakukan oleh tiga mahasiswa ditampilkan pada lampiran 2.

Pada data *tweet* awal memiliki banyak *atribute* yang tidak diperlukan dalam penelitian ini, maka dari itu peneliti akan menghilangkan beberapa *atribute* yang tidak digunakan dan hanya akan menampilkan beberapa *atribute* seperti username

dan *full_text*. Berikut pada gambar 4.4 pada kolom sentimen dan label menunjukkan bahwa data tweet awal sudah diberi label .



	A	B	C	D
1	username	full_text	Sentimen	Label
2	irfanwahidi60	Akhirnya terang benderang, gembar gembor polus	Negatif	-1
3	piotrj	ITU BUKAN KABUT Selama seminggu	Negatif	-1
4	cgtrisanjaya	@geloraco Betul. Sebaiknya pemerintah segera n	Negatif	-1
5	limitfungsii	gila parah, jakarta krisis polusi udara, gue dri flyov	Negatif	-1
6	Nofendra19	Polusi udara di Jakarta, lalu yang disalahkan ukm	Negatif	-1
7	tempodotco	Polusi Udara Jakarta Masih Tinggi Meski ASN WF	Negatif	-1
8	oewlc	Disinyalir ada Konflik kepentingan antara @kemei	Negatif	-1
9	DoankWarto	Memangnya selama 5 tahun ANIES menjabat GA	Negatif	-1
10	ISalutem	SANGSI PEMERINTAH SERIUS TANGANI POLL	Negatif	-1
11	Cerry0709	Meteri BUMN @erickthohir Yakin dengan LRT ini	Positif	1
12	ZulFan2238909(	@tvOneNews Warga Jakarta harus beralih dari m	Positif	1
13	Polusi_udara01	Polusi udara pagi hari ini pasti buruk. 4 jam lagi ai	Negatif	-1
14	Polusi_udara01	#Asean #Fifa #trending #southeastAsia #BakarSa	Negatif	-1
15	tvOneNews	Gunungan Batu Bara di Jakarta Utara Penyumbai	Positif	1
16	Polusi_udara01	Seperti apakah kisah polusi udara 31/8/2023? Ap	Negatif	-1
17	Polusi_udara01	@firlysavitri Polusi udara yg terjadi di Jakarta dis	Negatif	-1
18	Polusi_udara01	#Asean #Fifa #trending #southeastAsia #BakarSa	Negatif	-1
19	kompascom	Setelah pandemi Covid-19, penggunaan air purifi	Positif	1
20	anasurbaningrur	Polusi udara di Jakarta sedang sangat serius. Sel	Negatif	-1
21	tempodotco	Polusi Udara Jakarta, 2 Lagi Perusahaan Stockpil	Positif	1
22	gwabdeeeee	Menteri BUMN @erickthohir mengatakan, LRT Ja	Positif	1

Gambar 4. 4 Hasil Labelling Data

C. Text Preprocessing

Pada tahap *Text Preprocessing* ini bertujuan untuk menghilangkan noise dalam data *tweet* agar memudahkan perhitungan pembobotan kata dan analisis pada tahap berikutnya. Langkah awal dalam tahap ini yaitu memasukkan data csv ke dalam *directory python* yang terdapat di Google Collaboratory Python. Berikut adalah *source code* untuk memasukkan data csv yang berisi data tweet yang sudah melewati proses labelling ditampilkan pada gambar 4.5.

```
df = pd.read_csv('tweet-polusiudara-labelling.csv')
df
```

Gambar 4. 5 Source code memasukkan data csv

Memasukkan data csv dengan dibantu memanggil *library pandas* yang disingkat menjadi *pd* untuk membantu penulisan dalam memasukkan data dari sumber luar. Setelah itu memasukkan file data *tweet-polusiudara-labelling.csv* ke dalam *directory python* yang dibantu dengan menggunakan *pd* yang disimpan ke dalam *dataframe* yang merupakan *array* 2 dimensi yang dinamakan *df.*

Adapun tahapan-tahapan yang dilakukan pada proses *text preprocessing* akan melalui 7 proses secara urut, diantaranya:

1. Case Folding

Pada tahap case folding ini berproses dengan mengubah suatu teks yang memiliki huruf kapital menjadi huruf kecil. Adapun *source code* untuk proses *case folding* dapat dilihat pada gambar 4.6 berikut

```
# Mengubah huruf besar menjadi huruf kecil
def casefolding(text):
  text = text.lower()
  return text

df['CaseFolding'] = df['full_text'].apply(casefolding)
df
```

Gambar 4. 6 Source code proses case folding

Berikut hasil dari penerapan proses *case folding* yang akan ditunjukkan pada gambar 4.7 di bawah ini.

	username	full_text	Sentimen	Label	CaseFolding
0	irfanwahidi60	Akhirnya terang benderang, gembar gembor polus...	Negatif	-1	akhirnya terang benderang, gembar gembor polus...
1	piotrj	ITU BUKAN KABUT Selama seminggu ada WFH, s...	Netral	0	itu bukan kabut selama seminggu ada wfh, s...
2	cgtrisanjaya	@geloraco Betul. Sebaiknya pemerintah segera m...	Negatif	-1	@geloraco betul. sebaiknya pemerintah segera m...
3	limitfungsii	gila parah, jakarta krisis polusi udara, gue d...	Negatif	-1	gila parah, jakarta krisis polusi udara, gue d...
4	Nofendra19	Polusi udara di Jakarta, lalu yang disalahkan ...	Negatif	-1	polusi udara di jakarta, lalu yang disalahkan ...
...	...	...	...	...	...
505	Polusi_udara01	#Asean #Fifa #trending #southeastAsia #BakarSa...	Netral	0	#asean #fifa #trending #southeastasia #bakarsa...
506	5ulam_4ja	Nah jokowi mengumpulkan menteri untuk membahas...	Positif	1	nah jokowi mengumpulkan menteri untuk membahas...
507	radarbangsa_com	Polusi Udara di Jabodetabek Menparekraf Sebut ...	Negatif	-1	polusi udara di jabodetabek menparekraf sebut ...
508	KAI_THE_BRAND	yang masih nyemprot parfum biar ga bau badan m...	Netral	0	yang masih nyemprot parfum biar ga bau badan m...
509	detikfinance	Upaya pemerintah mengatasi polusi udara di DKI...	Positif	1	upaya pemerintah mengatasi polusi udara di dki...

510 rows × 5 columns

Gambar 4. 7 Hasil dari proses case folding

2. Cleansing

Proses *cleansing* bertujuan untuk membersihkan data pada dokumen yang masih memiliki berbagai atribut yang tidak diperlukan pada data tweet dengan menghapusnya dan diganti dengan spasi. Tahapan *cleansing* dilakukan dengan mengolah data tweet yang telah melalui proses *case folding.* Proses ini dilakukan untuk mengurangi kesalahan yang terjadi. Berikut langkah-langkah yang akan dilakukan pada proses *cleaning* yaitu sebagai berikut.

a. Melakukan instalasi modul emoji

Modul emoji adalah pustaka yang menyediakan berbagai fitur dan fungsi terkait dengan penggunaan emoji dalam pemrograman Python.

```
#sumber : https://pypi.org/project/emoji/

!pip install emoji
```

Gambar 4. 8 Install modul emoji

b. Melakukan import Regex library dan emoji

```
import re, emoji
```

Gambar 4. 9 Import Regex Library dan emoji

c. Menjalankan fungsi 'cleansing' untuk menghapus semua mention yang ditandai dengan simbol "@", kata hastag "#hastag", kata "RT" (*Retweet), URL (http://* atau *https:///),* menggabungkan beberapa spasi menjadi satu spasi serta menghapus spasi di awal dan di akhir, mengganti kata yang berulang menjadi kata tunggal, mengubah tanda baca berulang menjadi tunggal, menghapus kata tunggal, menghapus angka

```
def cleansing(text):
  text = re.sub(r'@[A-Za-z0-0_]+', '', text)
  text = re.sub(r'#\w+', '', text)
  text = re.sub(r'RT[\s]+', '', text)
  text = re.sub(r'(https|https):\/\/\S+', '', text)
  text = re.sub(r'[^A-Za-z0-9 ]', '', text)
  text = re.sub(r'\s+', ' ', text).strip()
  text = re.sub(r'(.)\1+', r'\1\1', text)
  text = re.sub(r'[\?\.\!]+(?=[\?.\!])', '', text)
  text = re.sub(r'\b[a-zA-Z]\b', '', text)
  text = re.sub(r'\d+', '', text)
  return text
```

Gambar 4. 10 Source Code Cleansing

d. Menjalankan fungsi '*remove_emojis*' untuk menghapus semua emoji dari teks

```
def remove_emojis(data):
    emoj = re.compile("["
        u"\U00002700-\U000027BF"  # Dingbats
        u"\U0001F600-\U0001F64F"  # Emoticons
        u"\U00002600-\U000026FF"  # Miscellaneous Symbols
        u"\U0001F300-\U0001F5FF"  # Miscellaneous Symbols And Pictographs
        u"\U0001F900-\U0001F9FF"  # Supplemental Symbols and Pictographs
        u"\U0001FA70-\U0001FAFF"  # Symbols and Pictographs Extended-A
        u"\U0001F680-\U0001F6FF"  # Transport and Map Symbols
                          "]+", re.UNICODE)
    return re.sub(emoj, '', data)
```

Gambar 4. 11 Code fungsi remove emoji

e. Menerapkan fungsi '*cleansing*' dan '*remove_emojis*' terhadap kolom 'CaseFolding' dan memasukkannya ke dalam kolom baru '*Cleansing*'

```
df['Cleansing'] = df['CaseFolding'].apply(cleansing, remove_emojis)
```

Gambar 4. 12 Source code menerapkan proses cleansing

Berikut hasil dari data tweet setelah dilakukan proses *cleansing* yang ditunjukkan pada gambar 4.13.

```
0      akhirnya terang benderang gembar gembor polusi...
1      itu bukan kabut selama seminggu ada wfh sempro...
2      betul sebaiknya pemerintah segera memberikan k...
3      gila parah jakarta krisis polusi udara gue dri...
4      polusi udara di jakarta lalu yang disalahkan u...
                           ...
505       selasa udara klp gading kembali tercemar ol...
506    nah jokowi mengumpulkan menteri untuk membahas...
507    polusi udara di jabodetabek menparekraf sebut ...
508    yang masih nyemprot parfum biar ga bau badan m...
509    upaya pemerintah mengatasi polusi udara di dki...
Name: Cleansing, Length: 510, dtype: object
```

Gambar 4. 13 Hasil dari proses cleansing

3. Remove Duplicate

Tahapan *Remove Duplicate* bertujuan menghapus *tweet* yang sama atau terduplikat untuk digunakan satu tanggapan saja. Berikut *source code* untuk menghapus baris yang terduplikasi seperti ditunjukkan pada gambar 4.14.

```
df = df.drop_duplicates(subset=['full_text'])
df = df.reset_index(drop =True)
df['full_text']
```

Gambar 4. 14 Source Code Proses Remove Duplicate

Adapun hasil dari penerapan tahapan *remove duplicate* menunjukkan bahwa terdapat perubahan terhadap jumlah data tweet yang semula 510 tweets berubah menjadi 492 tweets seperti ditunjukkan pada gambar 4.15 berikut.

```
0      Akhirnya terang benderang, gembar gembor polus...
1      🚩ITU 🚩BUKAN 🚩KABUT  Selama seminggu ada WFH, s...
2      @geloraco Betul. Sebaiknya pemerintah segera m...
3      gila parah, jakarta krisis polusi udara, gue d...
4      Polusi udara di Jakarta, lalu yang disalahkan ...
                             ...
487    #Asean #Fifa #trending #southeastAsia #BakarSa...
488    Nah jokowi mengumpulkan menteri untuk membahas...
489    Polusi Udara di Jabodetabek Menparekraf Sebut ...
490    yang masih nyemprot parfum biar ga bau badan m...
491    Upaya pemerintah mengatasi polusi udara di DKI...
Name: full_text, Length: 492, dtype: object
```

Gambar 4. 15 Hasil dari proses Remove duplicate

Pada proses remove duplicate data yang sama atau terduplikat akan di hilangkan kecuali data pertama. Berikut 18 data tweets yang terdeteksi sebagai data duplikat ditampilkan pada tabel 4.1 di bawah ini.

Tabel 4. 1 Data yang di remove

Indeks	Full_text
49	Untuk Atasi Polusi Udara di Jakarta, Pemerintah Disarankan Adopsi Jurus Anies https://t.co/nWFjEHZRoe
86	Untuk Atasi Polusi Udara di Jakarta, Pemerintah Disarankan Adopsi Jurus Anies https://t.co/nWFjEHZRoe
90	Untuk Atasi Polusi Udara di Jakarta, Pemerintah Disarankan Adopsi Jurus Anies https://t.co/nWFjEHZRoe
92	Untuk Atasi Polusi Udara di Jakarta, Pemerintah Disarankan Adopsi Jurus Anies https://t.co/nWFjEHZRoe
96	Untuk Atasi Polusi Udara di Jakarta, Pemerintah Disarankan Adopsi Jurus Anies https://t.co/nWFjEHZRoe
116	BPJS Habiskan Rp10 T untuk Penyakit Napas, Menkes: Terutama Akibat Polusi Udara Menkes menyarankan warga Jakarta mengenakan masker saat beraktivitas di luar ruangan. https://t.co/tWmVK7YXmA
117	BPJS Habiskan Rp10 T untuk Penyakit Napas, Menkes: Terutama Akibat Polusi Udara Menkes menyarankan warga

	Jakarta mengenakan masker saat beraktivitas di luar ruangan. https://t.co/tWmVK7YXmA
119	Sudah WFH Tapi Polusi Udara Jakarta Masih Tinggi? Heru Budi: Berarti dari Industi dan PLTU https://t.co/c1ugSY3jER
131	Guru Besar Ilmu Lingkungan Hidup Universitas Diponegoro Prof Sudharto P Hadi menyoroti membeludaknya jumlah kendaraan pribadi di Ibu Kota Jakarta yang berakibat pada peningkatan polusi udara. https://t.co/kYOHXn9FRH
148	Mengatasi Polusi Udara di Jakarta, Peran PLTU dan Teknologi Ramah Lingkungan https://t.co/SeizjLJzDJ
153	Menanti Pertamax Dikasih Subsidi Demi Kurangi Polusi Udara di Jakarta https://t.co/3hTrS3n6Hw
271	Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
274	Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
278	Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
282	Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
283	Halo teman teman semua,ternyata bukan PLTU ya penyebab polusi udara di jakarta memburuk. @https://apnews.com/article/indonesia-jakarta-air-pollution-dry-season-vehicles-ef97483d1c3de48207619562635710c2 https://t.co/sgSeU7asSe
290	Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4
297	Kurniasih: BPJS Kesehatan Wajib Lindungi dan Tanggung Pasien ISPA Akibat Polusi Udara di DKI Jakarta https://t.co/N74qgJDUN4

4. Tokenization

Proses *tokenization* merupakan suatu proses memecah teks menjadi beberapa bagian-bagian kata untuk mempermudah dalam pengolahan data lebih lanjut. Pada proses *tokenization* memerlukan *library* NLTK (*Natural Language Toolkit)* dengan modul '*punkt*' dengan menggunakan kode *nltk.download('punkt')* untuk dapat menggunakan fungsi *word_tokenize* yang berfungsi untuk membagi teks menjadi token . Berikut kode program yang digunakan dalam tahapan *tokenization* ditunjukkan pada gambar 4.16 berikut ini.

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
def word_tokenize_wrapper(text):
  return word_tokenize(text)def word_tokenize_wrapper(text):
  return word_tokenize(text)

df['Tokenize'] = df['Cleansing'].apply(word_tokenize_wrapper)
df['Tokenize']
```

Gambar 4. 16 Source code proses tokenization

Hasil dari penerapan tahapan *tokenization* diperlihatkan pada gambar 4.17 berikut ini.

```
0      [akhirnya, terang, benderang, gembar, gembor, ...
1      [itu, bukan, kabut, selama, seminggu, ada, wfh...
2      [betul, sebaiknya, pemerintah, segera, memberi...
3      [gila, parah, jakarta, krisis, polusi, udara, ...
4      [polusi, udara, di, jakarta, lalu, yang, disal...
                            ...
487    [selasa, udara, klp, gading, kembali, tercemar...
488    [nah, jokowi, mengumpulkan, menteri, untuk, me...
489    [polusi, udara, di, jabodetabek, menparekraf, ...
490    [yang, masih, nyemprot, parfum, biar, ga, bau,...
491    [upaya, pemerintah, mengatasi, polusi, udara, ...
Name: Tokenize, Length: 492, dtype: object
```

Gambar 4. 17 Hasil setelah dilakukan tokenization

Berikut pada gambar 4.18 ditunjukkan hasil proses tokenize pada indeks ke- 10 hingga indeks ke -15 untuk memperlihatkan perubahan yang terjadi secara jelas pada proses selanjutnya.

	Tokenize
10	[warga, jakarta, harus, beralih, dari, mobil, ...
11	[polusi, udara, pagi, hari, ini, pasti, buruk,...
12	[am, jakarta, rank, dunia, ada, lokasi, yg, po...
13	[gunungan, batu, bara, di, jakarta, utara, pen...
14	[seperti, apakah, kisah, polusi, udara, apakah...
15	[polusi, udara, yg, terjadi, di, jakarta, dise...

Gambar 4. 18 Hasil tokenize indeks 10 – 15

5. Normalization

Proses *normalization* merupakan suatu proses untuk mengubah kata singkatan atau *slang words* menjadi bahasa baku sesuai KBBI. Langkah awal yang perlu dipersiapkan pada proses ini yaitu menyiapkan kamus yang menyediakan kumpulan kata-kata yang akan digunakan sebagai kamus normalisasi. Kamus ini berisi kumpulan kosa kata yang tidak formal menjadi formal dalam bentuk *.txt* dan .csv. Jadi diperlukan beberapa *library* untuk dapat mengimplementasikan proses *normalization* ini yaitu dengan *import* data *pandas, csv,* dan *string.* Berikut tampilan *source code* yang diperlukan untuk dapat menggunakan *library* berikut ditunjukkan pada Gambar 4.19 berikut.

```
import pandas as pd
import csv
import string
```

Gambar 4. 19 Library proses normalisasi

Pada penelitian ini, proses normalization dilakukan dengan berdasarkan kamus yang diinputkan. Kamus pendukung yang digunakan sebanyak 3 file sehingga data akan melalui proses normalisasi sebanyak 3 tahapan. Kamus pertama dengan nama file "*collaquial-indonesian-lexicon.csv" (Aliyah et al., 2018)*. File yang kedua diberi nama "*kbba.txt" (Rio, 2018).* File yang ketiga diberi nama *"slangword.txt" (Rio,*

2018). Berikut *source code* untuk mengimplementasikan proses normalisasi tahap 1 ditampilkan pada Gambar 4.20.

```
normalized_word = pd.read_csv('colloquial-indonesian-lexicon.csv')

normalized_word_dict= {}
for index, row in normalized_word.iterrows():
  if row[0] not in normalized_word_dict:
    normalized_word_dict[row[0]]=row[1]

def normalized_term(document):
  return [normalized_word_dict[term] if term in normalized_word_dict else term for term in document]

df['Normalisasi'] = df['Tokenize'].apply(normalized_term)
df['Normalisasi']
```

Gambar 4. 20 Source code normalization 1

Adapun tahapan 2 dan 3 ditunjukkan pada Gambar 4.21 dan 4.22 dibawah ini.

```
normalized_word1 = pd.read_csv('kbba.txt', sep='\t')

normalized_word_dict1= {}
for index, row in normalized_word1.iterrows():
  if row[0] not in normalized_word_dict1:
    normalized_word_dict1[row[0]]=row[1]

def normalized_term1(document):
  return [normalized_word_dict1[term] if term in normalized_word_dict1 else term for term in document]

df['Normalisasi'] = df['Normalisasi'].apply(normalized_term1)
df['Normalisasi']
```

Gambar 4. 21 Source code normalization 2

```
normalized_word2 = pd.read_csv('slangword.txt', sep=':')

normalized_word_dict2= {}
for index, row in normalized_word2.iterrows():
  if row[0] not in normalized_word_dict2:
    normalized_word_dict2[row[0]]=row[1]

def normalized_term2(document):
  return [normalized_word_dict2[term] if term in normalized_word_dict2 else term for term in document]

df['Normalisasi'] = df['Normalisasi'].apply(normalized_term2)
df['Normalisasi']
```

Gambar 4. 22 Source code normalization 3

Hasil setelah dilakukan proses normalization ditunjukkan pada gambar 4.23 berikut ini.

```
0      [akhirnya, terang, benderang, gembar, gembor, ...
1      [itu, bukan, kabut, selama, seminggu, ada, ker...
2      [betul, sebaiknya, pemerintah, segera, memberi...
3      [gila, parah, jakarta, krisis, polusi, udara, ...
4      [polusi, udara, di, jakarta, lalu, yang, disal...
                             ...
487    [selasa, udara, kelapa, gading, kembali, terce...
488    [nah, jokowi, mengumpulkan, menteri, untuk, me...
489    [polusi, udara, di, jabodetabek, kementrian pa...
490    [yang, masih, semprot, parfum, biar, tidak, ba...
491    [upaya, pemerintah, mengatasi, polusi, udara, ...
Name: Normalisasi, Length: 492, dtype: object
```

Gambar 4. 23 Hasil proses normalisasi

Berikut tampilan perubahan pada indeks 10 hingga indeks 15 setelah dilakukan proses normalisasi ditunjukkan pada gambar 4.24 berikut.

	Normalisasi
10	[warga, jakarta, harus, beralih, dari, mobil, ...
11	[polusi, udara, pagi, hari, ini, pasti, buruk,...
12	[sama, jakarta, rank, dunia, ada, lokasi, yang...
13	[gunungan, batu, bara, di, jakarta, utara, pen...
14	[seperti, apakah, kisah, polusi, udara, apakah...
15	[polusi, udara, yang, terjadi, di, jakarta, di...

Gambar 4. 24 Hasil normalisasi indeks 10 - 15

6. Stopword Removal

Proses *stopword removal* adalah suatu langkah dalam proses *text pre-processing* yang bertujuan untuk

menghilangkan kata-kata yang tidak memiliki makna dan banyak informasi atau bisa disebut *stopword*. Kamus *stopwords* yang digunakan pada penelitian ini adalah *stopwords* bahasa indonesia. Berikut source code untuk mengimplementasikan proses *stopword removal* seperti ditunjukkan pada gambar 4.25 di bawah ini.

```
stopw = set(stopwords.words('indonesian'))

def sw_removal (words):
  return [word for word in words if word not in stopw]
# Menghapus stopwords dari setiap teks dalam DataFrame
df['Stopword_Removal'] = df['Normalisasi'].apply(sw_removal)
df['Stopword_Removal']
```

Gambar 4. 25 Source code proses stopword removal

Adapun hasil dari penerapan proses *stopword removal* ditunjukkan pada gambar 4.26 berikut.

```
0      [terang, benderang, gembar, gembor, polusi, ud...
1      [kabut, seminggu, kerja dari rumah, semprot, a...
2      [pemerintah, ktp, sih, udara, polusi, memerint...
3      [gila, parah, jakarta, krisis, polusi, udara, ...
4      [polusi, udara, jakarta, disalahkan, ukm, indu...
                             ...
487    [selasa, udara, kelapa, gading, tercemar, sese...
488    [jokowi, mengumpulkan, menteri, membahas, polu...
489    [polusi, udara, jabodetabek, kementrian pariwi...
490    [semprot, parfum, biar, bau, badan, menambah, ...
491    [upaya, pemerintah, mengatasi, polusi, udara, ...
Name: Stopword_Removal, Length: 492, dtype: object
```

Gambar 4. 26 Hasil dari proses stopword removal

Berikut tampilan perubahan pada indeks 10 hingga 15 setelah dilakukan proses stopword removal ditunjukkan pada gambar 4.27 berikut ini.

	Stopword_Removal
10	[warga, jakarta, beralih, mobil, pribadi, moda...
11	[polusi, udara, pagi, buruk, jam, anak, masuk,...
12	[jakarta, rank, dunia, lokasi, polusi, udara, ...
13	[gunungan, batu, bara, jakarta, utara, penyumb...
14	[kisah, polusi, udara, buruknya, lihat, esok, ...
15	[polusi, udara, jakarta, disengaja, maksud, an...

Gambar 4. 27 Hasil stopword removal indeks 10 -15

7. Stemming

Proses *stemming* bertujuan untuk mengubah kata yang ber-imbuhan menjadi sebuah kata dasar. Pada proses ini *library* yang akan digunakan adalah *library sastrawi.* Maka untuk dapat menggunakan *library sastrawi,* langkah awal yaitu dengan menginstall library sastrawi terlebih dahulu selanjutnya mengimport *package stemmer* yang diperlukan untuk proses *stemming* seperti ditunjukkan pada gambar 4.28 berikut.

```
!pip install Sastrawi
```

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

```
# Membuat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()
```

Gambar 4. 28 Library proses Stemming

Selanjutnya menjalankan fungsi *stemming* dan menerapkannya ke dalam data di kolom *Stopword_removal* seperti gambar 4.29 berikut.

```
def stemming(text):
  return [stemmer.stem(token) for token in text]

df['Stemming'] = df['Stopword_Removal'].apply(stemming)
df['Stemming']
```

Gambar 4. 29 Source code stemming

Pada gambar 4.30 dibawah ini merupakan hasil dari penerapan proses *stemming* yang telah dilakukan.

```
0      [terang, benderang, gembar, gembor, polusi, ud...
1      [kabut, minggu, kerja dari rumah, semprot, air...
2      [perintah, ktp, sih, udara, polusi, perintah, ...
3      [gila, parah, jakarta, krisis, polusi, udara, ...
4      [polusi, udara, jakarta, salah, ukm, industri,...
                             ...
487    [selasa, udara, kelapa, gading, cemar, segera ...
488    [jokowi, kumpul, menteri, bahas, polusi, udara...
489    [polusi, udara, jabodetabek, tri pariwisata da...
490    [semprot, parfum, biar, bau, badan, tambah, po...
491    [upaya, perintah, atas, polusi, udara, dki, ja...
Name: Stemming, Length: 492, dtype: object
```

Gambar 4. 30 Hasil proses Stemming

Pada gambar 4.31 ditampilkan perubahan hasil setelah dilakukan proses stemming pada indeks 10 hingga indeks 15 berikut ini.

	Stemming
10	[warga, jakarta, alih, mobil, pribadi, moda, t...
11	[polusi, udara, pagi, buruk, jam, anak, masuk,...
12	[jakarta, rank, dunia, lokasi, polusi, udara, ...
13	[gunung, batu, bara, jakarta, utara, sumbang, ...
14	[kisah, polusi, udara, buruk, lihat, esok, kis...
15	[polusi, udara, jakarta, sengaja, maksud, anak...

Gambar 4. 31 Hasil stemming indeks 10 - 15

Adapun dilakukan perhitungan untuk mengetahui *accuray* dari proses *stemming* yang telah dilakukan dengan perhitungan berikut.

$$\text{accuracy} = \frac{total\ token\ sama}{total\ token}\ x\ 100\%$$

Berikut tampilan *source code* untuk menghitung *accuracy* dari *stemming* yang ditunjukkan pada gambar 4.32 dibawah ini.

```
# Menghitung akurasi Stemming

# Token asli
token_asli = df['Stopword_Removal']
# Token setelah stemming
stemmed_tokens = df['Stemming']

# Inisialisasi list untuk menyimpan token yang sama
same_tokens_list = []
# Menyimpan token yang sama
for original, stemmed in zip(token_asli, stemmed_tokens):
    same_tokens = [token for token in original if token in stemmed]
    same_tokens_list.append(same_tokens)

# Menghitung jumlah token yang sama antara token asli dan token setelah stemming
total_token_sama = sum(sum(1 for original, stemmed in zip(original, stemmed)
if original == stemmed) for original, stemmed in zip(token_asli, stemmed_tokens))
# Menghitung total token
total_tokens = sum(len(tokens) for tokens in token_asli)

# Menghitung akurasi stemming
accuracy = total_token_sama / total_tokens * 100

print("Jumlah token yang sama:", total_token_sama)
print("Jumlah total token:", total_tokens)
print("Akurasi stemming:", accuracy, "%")
```

Gambar 4. 32 Source code menghitung accuracy stemming

Hasil yang didapatkan dari perhitungan *accuracy stemming* yang telah dilakukan menunjukkan bahwa nilai *accuracy* dari proses *stemming* sebesar 75 % seperti terlihat pada gambar 4.33 berikut ini.

```
Jumlah token yang sama: 5434
Jumlah total token: 7205
Akurasi stemming: 75.41984732824427 %
```

Gambar 4. 33 Hasil dari accuracy stemming

Sebuah nilai akurasi stemming sebesar 75% menunjukkan bahwa algoritma stemming berhasil mengidentifikasi kata-kata yang mirip atau terkait dengan

kata dasar dengan benar dalam 75% kasus yang diuji. Namun, masih ada 25% kasus dimana stemming tidak berfungsi dengan baik karena kata dasar yang tidak dikenali atau pengenalan yang tidak tepat terhadap kata-kata tertentu. Data *tweet* telah melalui proses *text preprocessing.* Maka dari itu, hasil data *tweet* berikut dapat digunakan untuk melakukan penelitian dengan menyimpan data tersebut ke dalam bentuk file *csv* baru yang diberi nama *"tweet-polusiudara—prossesing.csv"*. Hasil dari data *tweet* yang sudah melalui proses *preprocessing* ditampilkan pada gambar 4.34 di bawah ini.

	username	full_text	Sentimen	Label	CaseFolding	Cleansing	Tokenize	Normalisasi	Stopword_Removal	Stemming	Tweet_bersih
0	irfanwahidi60	Akhirnya terang benderang, gembar gembor polus...	Negatif	-1	akhirnya terang benderang, gembar gembor polus...	akhirnya terang benderang gembar gembor polusi...	[akhirnya, terang, benderang, gembar, gembor, ...	[akhirnya, terang, benderang, gembar, gembor, ...	[terang, benderang, gembar, gembor, polusi, ud...	[terang, benderang, gembar, gembor, polusi, ud...	terang benderang gembar gembor polusi udara ja...
1	piotrj	▶ ITU ▶ BUKAN ▶ KABUT Selama seminggu ada WFH, s...	Negatif	-1	▶ itu ▶ bukan ▶ kabut selama seminggu ada wfh, s...	itu bukan kabut selama seminggu ada wfh sempro...	[itu, bukan, kabut, selama, seminggu, ada, wfh...	[itu, bukan, kabut, selama, seminggu, ada, ker...	[kabut, seminggu, kerja dari rumah, semprot, a...	[kabut, minggu, kerja dari rumah, semprot, air...	kabut minggu kerja dari rumah semprot air huja...
2	cgtrisanjaya	@geloraco Betul. Sebaiknya pemerintah segera m...	Negatif	-1	@geloraco betul. sebaiknya pemerintah segera m...	betul sebaiknya pemerintah segera memberikan k...	[betul, sebaiknya, pemerintah, segera, memberi...	[betul, sebaiknya, pemerintah, segera, memberi...	[pemerintah, ktp, sih, udara, polusi, memerint...	[perintah, ktp, sih, udara, polusi, perintah, ...	perintah ktp sih udara polusi perintah udara p...
3	limitfungsii	gila parah, jakarta krisis polusi udara, gue d...	Negatif	-1	gila parah, jakarta krisis polusi udara, gue d...	gila parah jakarta krisis polusi udara gue dri...	[gila, parah, jakarta, krisis, polusi, udara, ...	[gila, parah, jakarta, krisis, polusi, udara, ...	[gila, parah, jakarta, krisis, polusi, udara, ...	[gila, parah, jakarta, krisis, polusi, udara, ...	gila parah jakarta krisis polusi udara fiyover...
4	Nofendra19	Polusi udara di Jakarta, lalu yang disalahkan ...	Negatif	-1	polusi udara di jakarta, lalu yang disalahkan ...	polusi udara di jakarta lalu yang disalahkan u...	[polusi, udara, di, jakarta, lalu, yang, disal...	[polusi, udara, di, jakarta, lalu, yang, disal...	[polusi, udara, jakarta, disalahkan, ukm, indu...	[polusi, udara, jakarta, salah, ukm, industri,...	polusi udara jakarta salah ukm industri sebab ...
...	...	...	...	...	...	...	...	...	...	...	...

Gambar 4. 34 Hasil akhir setelah melalui text preprocessing

D. TFIDF

Data *tweet* yang telah dilakukan proses *text preprocessing* masih berupa sebuah teks atau kata-kata. Namun untuk melakukan analisis klasifikasi data yang diperlukan harus

berbentuk numerik atau angka. Maka dari itu, terlebih dahulu data *tweet* dilakukan konversi kedalam bentuk numerik angka dengan melakukan pembobotan kata menggunakan TF-IDF.

Langkah awal yang perlu dipersiapkan dalam penerapan proses pembobotan kata dengan menginstall *library* yang diperlukan. Dalam hal ini, proses TFIDF dibantu menggunakan *library scikit learn* yang perlu dilakukan instalasi dengan source *code "!pip3".* Proses pembobotan kata akan dilakukan perhitungan menggunakan file data *tweet* yang sudah dilakukan *preprocessing.* Untuk dapat membaca file dilakukan mengunakan kode 'pd.read_csv' untuk membaca file CSV dengan nama "tweet-polusiudara—processing.csv" dan menyimpannya ke dalam sebuah DataFrame yang disebut 'df' dengan didukung *pandas* dan *numpy.* Pada hal ini kolom yang akan digunakan hanya kolom 'Tweet_bersih' yang merupakan hasil akhir tweet yang telah melalui proses *preprocessing.* Dalam hal ini, dikarenakan hanya kolom "Tweet_bersih", "Sentimen", dan "Label" yang dipilih peneliti untuk ditampilkan, maka peneliti menggunakan parameter '*usecols*' untuk menentukan kolom-kolom mana yang akan digunakan dari file CSV tersebut. Nama kolom untuk "Tweet_bersih" akan diganti nama kolom nya menjadi "Tweet_join".

Berikut kode program yang diperlukan untuk membaca dataset yang akan diolah seperti ditunjukkan pada gambar 4.35 dan hasil

yang akan muncul dari kode program tersebut ditampilkan pada gambar 4.36 berikut ini.

```
!pip3 install -U scikit-learn
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.4.2)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.10/dist-packages (from scikit-
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-l
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from
```

```
import pandas as pd
import numpy as np

df = pd.read_csv("tweet-polusiudara--prossesing.csv", usecols=["Sentimen", "Label", "Tweet_bersih"])
df.columns = ["Sentimen", "Label", "Tweet_join"]

df
```

Gambar 4. 35 Kode program membaca data untuk proses TFIDF

	Sentimen	Label	Tweet_join
0	Negatif	-1	terang benderang gembar gembor polusi udara ja...
1	Negatif	-1	kabut minggu kerja dari rumah semprot air huja...
2	Negatif	-1	perintah ktp sih udara polusi perintah udara p...
3	Negatif	-1	gila parah jakarta krisis polusi udara flyover...
4	Negatif	-1	polusi udara jakarta salah ukm industri sebab ...
...	...	...	...
487	Negatif	-1	selasa udara kelapa gading cemar segera mungki...
488	Positif	1	jokowi kumpul menteri bahas polusi udara jakar...
489	Negatif	-1	polusi udara jabodetabek tri pariwisata dan ek...
490	Negatif	-1	semprot parfum biar bau badan tambah polusi ud...
491	Positif	1	upaya perintah atas polusi udara dki jakarta g...

492 rows × 3 columns

Gambar 4. 36 Data yang akan dilakukan proses TFIDF

Pada *library scikit-learn* ini perhitungan *Term Frequency-Invers Document Frequency* (TF-IDF) dilakukan dalam beberapa tahapan. Pertama, menghitung *Term Frequency (TF)* menggunakan metode *'fit_transform()'* dari kelas *CountVectorizer.* Selanjutnya, *Inverse Document Frequency* (IDF) dihitung menggunakan metode *'fit()'* dari kelas *TfidTransformer*. Terakhir, TF-IDF dihitung menggunakan metode *'transform()'* dari kelas *TfidfTransformer(Santoso,2021)*. Pada penelitian ini, digunakan kelas TfidfVectorizer yang disediakan *library Scikit-Learn* yang secara internal menggabungkan perhitungan dari *CountVectorizer* untuk menghitung frekuensi kemunculan kata-kata dalam dokumen (TF) dan kemudian mengalikannya dengan bobot IDF yang dihitung dari kumpulan dokumen tersebut menggunakan *TfidfTransformer.* perbedaan utama antara keduanya adalah *TfidVectorizer* digunakan untuk mengubah teks menjadi matriks TFIDF secara keseluruhan mulai dari awal, sementara *TfidfTransformer* digunakan untuk memodifikasi matriks TFIDF yang sudah ada. Maka dari itu, peneliti menggunakan modul *TfidfVectorizer* pada proses perhitungan TFIDF. Selain itu, dari modul '*sklearn.preprocessing*' juga mengimpor fungsi *'normalize'* yang digunakan untuk melakukan normalisasi data *vector* sehingga menghasillkan representasi data yang seragam. Berikut tampilan kode program untuk

menambahkan fungsi yang diperlukan dalam proses TFIDF ditunjukkan pada gambar 4.37 berikut.

```
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.preprocessing import normalize
```

Gambar 4. 37 Library TFIDF

Peneliti dalam menghitung matriks TFIDF dari teks yang akan diproses dengan membatasi jumlah fitur yang akan digunakan dalam vektorisasi sebanyak 2000 dengan menggunakan *code 'max_features'.* Kemudian, untuk menghasilkan matriks TFIDF didefinisikan pada sebuah fungsi *'generate_tfidf_mat(min_gram, max_gram)'* yang menerima dua argumen yaitu *min_gram* dan *max_gram* yang digunakan untuk menentukan rentang n-gram yang akan digunakan dalam vektorisasi. Selanjutnya, untuk perhitungan TF menggunakan *CountVectorizer* dengan parameter *'max_features'* dan *'ngram_range'* yang hasilnya adalah matriks frekuensi term (TF_vector). Vektor TF dinormalisasikan menggunakan fungsi *'normalize'* agar panjang vektor menjadi 1. Perhitungan bobot IDF dilakukan menggunakan *TfidfVectorizer* dengan parameter *'max_features'* dan *'ngram_range'* hasil dari perhitungan tersebut yaitu matriks IDF_vector. Berikut tampilan *source code* yang digunakan untuk menghitung TF dan IDF, ditunjukkan pada gambar 4.38 berikut.

```
max_features = 2000

# calc TF VECTOR
def generate_tfidf_mat(min_gram, max_gram):
    cvect = CountVectorizer(max_features=max_features, ngram_range=(min_gram, max_gram))
    TF_vector = cvect.fit_transform(df["Tweet_join"])

    normalized_TF_vector = normalize(TF_vector, norm='l1', axis=1)

# calc IDF
    tfidf = TfidfVectorizer(max_features=max_features, ngram_range=(min_gram, max_gram), smooth_idf=False)
    tfs = tfidf.fit_transform(df["Tweet_join"])
    IDF_vector = tfidf.idf_
```

Gambar 4. 38 Source Code menghitung TF dan IDF

Perhitungan TFIDF dihitung dengan mengalikan matriks TF dengan IDF dengan fungsi *multiply()* menghasilkan matriks TFIDF (tfidf_mat). Hasil dari matriks TF, IDF, dan TFIDF disimpan dalam variabel 'TF', 'IDF', dan 'TF_IDF' secara berturut-turut. Pengambilan daftar term (kata-kata) yang telah diekstrak dari teks dilakukan menggunakan fungsi *'get_features_names_out()'.* Peneliti hanya melibatkan satu term saja dalam menghitung TFIDF hal ini disebut TFIDF unigram. Pemanggilan fungsi *'generate_tfidf_mat(1,1)'* dilakukan untuk menghitung matriks TFIDF menggunakan unigram. Selanjutnya, peneliti melakukan cek data sparse untuk memeriksa nilai vektor TF, IDF, dan TFIDF yang tidak nol dari setiap indeks dan dicetak dalam bentuk tabel untuk diperiksa. Tampilan *source code* menghitung matriks TFIDF dan menampilkan nilai vektor TF, IDF dan TFIDF setiap indeks ditunjukkan pada gambar 4.39 berikut.

```
# calc TF -IDF
    tfidf_mat = normalized_TF_vector.multiply(tfidf.idf_).toarray()

    TF = normalized_TF_vector.toarray()
    IDF = tfidf.idf_
    TF_IDF = tfidf_mat
    return TF, IDF, TF_IDF, tfidf.get_feature_names_out()

# ngram_range (1, 1) hanya menghitung unigram
tf_mat_unigram, idf_mat_unigram, tfidf_mat_unigram, terms_unigram = generate_tfidf_mat(1,1)

# ---------- check sparse data -------------------
idx_sample = 0

print("Show TFIDF sample ke-" + str(idx_sample), "\n")
print(df["Tweet_join"][idx_sample], "\n")

print("\t\t\t", "TF", "\t\t", "IDF", "\t\t", "TF-IDF", "\t", "Term\n")
for i, item in enumerate(zip(tf_mat_unigram[idx_sample], idf_mat_unigram, tfidf_mat_unigram[idx_sample], terms_unigram)):
    if(item[2] != 0.0):
        print ("array position " + str(i) + "\t",
               "%.6f" % item[0], "\t",
               "%.6f" % item[1], "\t",
               "%.6f" % item[2], "\t",
               item[3])
```

Gambar 4. 39 Script menghitung TFIDF

Hasil dari cek data sparse yang dilakukan peneliti untuk mengetahui nilai pembobotan kata setiap indeks. Pada gambar 4.40 ditunjukkan tampilan hasil cek data sparse pada indeks ke-0 (nol).

```
Show TFIDF sample ke-0

terang benderang gambar gembor polusi udara jakarta akselerasi jual mobil motor listrik naik harga pertalite menggati pertalite green harga harga pertalite

                        TF              IDF             TF-IDF          Term

array position 37       0.047619        7.198479        0.342785        akselerasi
array position 196      0.047619        7.198479        0.342785        benderang
array position 429      0.047619        6.505332        0.309778        gambar
array position 430      0.047619        6.099866        0.290470        gembor
array position 454      0.047619        7.198479        0.342785        green
array position 481      0.142857        6.505332        0.929333        harga
array position 576      0.047619        1.039383        0.049494        jakarta
array position 607      0.047619        5.406719        0.257463        jual
array position 804      0.047619        3.940382        0.187637        listrik
array position 881      0.047619        7.198479        0.342785        menggati
array position 901      0.047619        4.895894        0.233138        mobil
array position 912      0.047619        4.153956        0.197807        motor
array position 929      0.047619        6.099866        0.290470        naik
array position 1048     0.142857        6.505332        0.929333        pertalite
array position 1083     0.047619        1.050010        0.050000        polusi
array position 1420     0.047619        6.505332        0.309778        terang
array position 1493     0.047619        1.010215        0.048105        udara
```

Gambar 4. 40 Hasil proses TFIDF indeks 0

Selanjutnya untuk mengekstrak nilai – nilai TF, IDF, dan TFIDF yang telah didapatkan dari setiap indeks ke dalam bentuk daftar dan ditambahkan sebagai kolom baru ke dalam DataFrame ‘df’. Dengan demikian, nilai-nilai tersebut setiap baris akan

disimpan dalam bentuk daftar tabel. Berikut tampilan *source code* untuk mengekstrak nilai-nilai tersebut ke dalam bentuk tabel yang ditunjukkan dalam Gambar 4.41 di bawah ini.

```
def get_TF(row):
    idx = row.name
    return [tf for tf in tf_mat_unigram[idx] if tf != 0.0]

df["TF"] = df.apply(get_TF, axis=1)

def get_IDF(row):
    idx = row.name
    return [item[1] for item in zip(tf_mat_unigram[idx], idf_mat_unigram) if item[0] != 0.0]

df["IDF"] = df.apply(get_IDF, axis=1)

def get_TFIDF(row):
    idx = row.name
    return [tfidf for tfidf in tfidf_mat_unigram[idx] if tfidf != 0.0]

df["TFIDF"] = df.apply(get_TFIDF, axis=1)
```

Gambar 4. 41 Source code mengekstrak TFIDF ke dalam tabel

Hasil akhir dari perhitungan pembobotan kata yang telah dilakukan pada keseluruhan kata dalam dokumen ditampilkan pada Gambar 4.42 berikut ini.

	Sentimen	Label	Tweet_join	TF	IDF	TFIDF
0	Negatif	-1	terang benderang gembar gembor polusi udara ja...	[0.047619047619047616, 0.047619047619047616, 0...	[7.198478716492308, 7.198478716492308, 6.50533...	[0.34278470078534795, 0.34278470078534795, 0.3...
1	Negatif	-1	kabut minggu kerja dari rumah semprot air huja...	[0.043478260869565216, 0.043478260869565216, 0...	[3.5875608038480835, 4.490428515390098, 4.6335...	[0.15598090451513405, 0.19523602240826513, 0.2...
2	Negatif	-1	perintah ktp sih udara polusi perintah udara p...	[0.09090909090909091, 0.09090909090909091, 0.0...	[6.099866427824199, 1.0393833280003753, 7.1984...	[0.5545333116203818, 0.09448939345457957, 0.65...
3	Negatif	-1	gila parah jakarta krisis polusi udara flyover...	[0.09090909090909091, 0.09090909090909091, 0.0...	[7.198478716492308, 4.800583443693938, 7.19847...	[0.6544071560447553, 0.43641667669944895, 0.65...
4	Negatif	-1	polusi udara jakarta salah ukm industri sebab ...	[0.09090909090909091, 0.09090909090909091, 0.0...	[6.5053315359323625, 4.308106958596143, 1.0393...	[0.5913937759938511, 0.39164608714510396, 0.09...
...	...	...	...	...	...	...

Gambar 4. 42 Hasil akhir perhitungan TF, IDF, dan TFIDF

Penelitian menggunakan 3 data *tweet* untuk dilakukan perhitungan manualisasi pada proses pembobotan kata dengan TFIDF sebagai berikut.

D1	Polusi Udara Mengancam, Kualitas Udara jakarta Belum Penuhi Hak Ekologis Anak! https://t.co/BIyTQ4Dt2W	Negatif
D2	@budibongg Di Jakarta pohon bnyk yg ditebang gedung dimana mana jd polusi udara sgt mengancam	Negatif
D3	Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/CYo2wjibsP	Positif

Setelah melalui text preprocessing data tweet di atas akan berubah menjadi seperti berikut:

D1 = “polusi udara ancam kualitas udara jakarta penuh hak ekologis anak”

D2 = “jakarta pohon tebang gedung mana polusi udara ancam”

D3 = “gerak salah tindak nyata atasi polusi udara jakarta gerak gotong royong pindah pohon”

Proses selanjunya pada perhitungan manualisasi pembobotan kata dengan menghitung TF, IDF dan TFIDF. Menghitung nilai TF (*Term Frequency*) adalah jumlah tiap-tiap kata yang ada dalam dokumen. Berikut tabel proses

perhitungan TF pada 3 (tiga) dokumen di atas ditunjukkan pada tabel 4.2 berikut.

Tabel 4. 2 Hasil perhitungan TF

Kata (Term)	TF		
	D1	D2	D3
polusi	1	1	1
udara	2	1	1
ancam	1	1	0
kualitas	1	0	0
jakarta	1	1	1
penuh	1	0	0
hak	1	0	0
ekologis	1	0	0
anak	1	0	0
gerak	0	0	2
salah	0	0	1
tindak	0	0	1
nyata	0	0	1
atasi	0	0	1
gotong	0	0	1
royong	0	0	1
pindah	0	0	1
pohon	0	1	1
tebang	0	1	0
gedung	0	1	0
mana	0	1	0

Selanjutnya, pada proses perhitungan DF dilakukan perhitungan jumlah dokumen yang memiliki kata (term). Ketika sebuah kata ditemukan dalam satu dokumen, nilai df

akan bertambah satu hingga mencapai jumlah total dokumen, yang mencerminkan frekuensi kemunculan kata tersebut di semua dokumen. Pada tabel 4.3 ditampilkan hasil perhitungan df berikut.

Tabel 4. 3 Hasil perhitungan DF

Kata (Term)	TF			DF
	D1	D2	D3	
polusi	1	1	1	3
udara	2	1	1	4
ancam	1	1	0	2
kualitas	1	0	0	1
jakarta	1	1	1	3
penuh	1	0	0	1
hak	1	0	0	1
ekologis	1	0	0	1
anak	1	0	0	1
gerak	0	0	2	2
salah	0	0	1	1
tindak	0	0	1	1
nyata	0	0	1	1
atasi	0	0	1	1
gotong	0	0	1	1
royong	0	0	1	1
pindah	0	0	1	1
pohon	0	1	1	2
tebang	0	1	0	1
gedung	0	1	0	1
mana	0	1	0	1

Setelah nilai DF diperoleh, langkah selanjutnya adalah menghitung IDF. Dalam perhitungan ini, jumlah total dokumen (D) yang digunakan adalah tiga. Proses perhitungan IDF dilakukan sesuai pada persamaan 2.3 di atas sehingga nilai IDF yang didapatkan ditampilkan pada tabel 4.4 di bawah ini.

Tabel 4. 4 Hasil perhitungan IDF

Kata (Term)	DF	D/DF (D=3)	IDF
polusi	3	1	0
udara	4	1.3	0.113
ancam	2	1.5	0.176
kualitas	1	3	0.477
jakarta	3	1	0
penuh	1	3	0.477
hak	1	3	0.477
ekologis	1	3	0.477
anak	1	3	0.477
gerak	2	1.5	0.176
salah	1	3	0.477
tindak	1	3	0.477
nyata	1	3	0.477
atasi	1	3	0.477
gotong	1	3	0.477
royong	1	3	0.477
pindah	1	3	0.477
pohon	2	1.5	0.176
tebang	1	3	0.477
gedung	1	3	0.477
mana	1	3	0.477

Berdasarkan pada tabel 4.1 dan 4.3 telah didapatkan nilai TF dan IDF. Selanjutnya melakukan perhitungan pembobotan kata sesuai pada persamaan 2.2. Pada tabel 4.5 ditunjukkan hasil dari perhitungan TF-IDF berikut.

Tabel 4. 5 Hasil perhitungan TFIDF (W)

Kata (Term)	**TF*IDF**		
	D1	**D2**	**D3**
polusi	0	0	0
udara	0,226	0,113	0,113
ancam	0,176	0,176	0
kualitas	0,477	0	0
jakarta	0	0	0
penuh	0,477	0	0
hak	0,477	0	0
ekologis	0,477	0	0
anak	0,477	0	0
gerak	0	0	0,352
salah	0	0	0,477
tindak	0	0	0,477
nyata	0	0	0,477
atasi	0	0	0,477
gotong	0	0	0,477
royong	0	0	0,477
pindah	0	0	0,477
pohon	0	0,176	0,176
tebang	0	0,477	0
gedung	0	0,477	0
mana	0	0,477	0

E. Split Validation Data

Tahapan setelah dilakukan proses perhitungan TF IDF atau pembobotan kata. Selanjutnya, data *tweet* yang akan diklasifikasikan dilakukan pembagian data menjadi data latih dan data uji terlebih dahulu. Pada penelitian ini dilakukan perhitungan klasifikasi sentimen dengan membagi data latih dan data uji dengan menggunakan perbandingan 90:10, 80:20, 70:30, dan 60:40. Maka dari itu, tahapan perhitungan klasifikasi pada setiap model akan dilakukan sebanyak 4 (empat) kali. Pada proses ini *library* yang digunakan yaitu *library sklearn,* untuk dilakukan pembagian data menggunakan kelas *train_test_split.* Pembagian data yang dilakukan memisahkan data untuk digunakan sebagai fitur dan label direpresentassikan menggunakan variabel X dan Y.

Pada penelitian ini data yang digunakan sebagai fitur atau atribut yang akan digunakan sebagai masukan untuk model dengan menggunakan data *tweet* yang telah dilakukan pembersihan atau *text preprocessing* yaitu data yang terdapat pada kolom "Tweet_bersih" yang telah melalui proses pembobotan kata sehingga data yang digunakan diambil dari *"tfidf_mat_unigram"*. Sedangkan untuk variabel Y yang digunakan untuk menyimpan label atau target yang ingin diprediksi oleh model. Pada analisis ini, variabel Y berisi label atau klasifikasi dari data *tweet* tersebut yang pada hal ini terdapat pada kolom

"Label". Berikut *source code* yang digunakan untuk memisahkan data menjadi fitur dan label dengan perbandingan data 90:10 yang ditunjukkan pada Gambar 4.43 berikut.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(tfidf_mat_unigram, df['Label'], test_size=0.1, random_state=42)
```

Gambar 4. 43 Source code split validation data

Pada penelitian ini, peneliti melakukan percobaan perhitungan klasifikasi dengan melakukan uji pada 4 (empat) perbandingan yang telah ditentukan. Perbedaan *script* pembagian data terdapat pada data test dengan melihat fungsi *test_size* yang diambil, berikut ditunjukkan penentuan pembagian data sesuai perbandingannya ditampilkan pada tabel 4.6 berikut.

Perbandingan	*test_size*
90 : 10	0.1
80 : 20	0.2
70 : 30	0.3
60 : 40	0.4

Tabel 4. 6 Pembagian data

F. Klasifikasi Naïve Bayes Classifier

Model klasifikasi pertama yang dilakukan pada analisis ini setelah data *tweet* telah melalui tahapan *text preprocessing* serta pembagian data latih dan data uji yaitu dilakukan perhitungan klasifikasi sentimen menggunakan Naive Bayes Classifier. Pada penelitian ini pengujian yang dilakukan untuk

menguji ketepatan sistem dalam proses klasifikasi data dengan menghitung nilai probabilitas yang lebih tinggi diantara kedua sentimen untuk menentukan label sentimen pada data tweet tersebut. Pada penelitian ini dilakukan perhitungan sentimen secara manual menggunakan metode Naive Bayes Classifier, proses perhitungan dengan mengambil salah satu data tweet untuk dianalisis.

Tweet = "Polusi Jakarta itu memang parah , sekarang mau salahkan siapa? Ruang hijau sudah minim sekali dijakarta, sudah banyak bangunan beton, populasi kendaraan terus bertambah terutama sepeda motor, home industri dan pabrik banyak menyebabkan polusi air dan udara"

Langkah perhitungan manual penentuan label sentimen pada metode Naive Bayes sebagai berikut.

1. Menganalisis data latih
    a. Polusi : kategori kata negatif
    b. Parah : kategori kata negatif
    c. Ruang hijau : kategori kata positif
    d. Minim : kategori kata negatif
2. Melakukan proses perhitungan probabilitas *prior* setiap kategori sentimen
    a. Jumlah *prior* positif = 1

       Jumlah *prior* negatif = 3

b. Rumus probabilitas positif sebagai berikut.

$$P\ (Positif) = \frac{Jumlah\ prior\ positif}{Total\ dokumen} = \frac{1}{1+3} = \frac{1}{4}$$

c. Rumus probabilitas negatif sebagai berikut.

$$P\ (Negatif) = \frac{Jumlah\ prior\ negatif}{Total\ dokumen} = \frac{3}{1+3} = \frac{3}{4}$$

3. Perhitungan probabilitas Likelihood untuk setiap kata dalam setiap kategori.

a. P("Polusi"|Positif)= $\frac{Jumlah\ kata\ kategori\ positif}{Jumlah\ prior\ positif} = \frac{0}{1} = 0$

P("Polusi"|Negatif)= $\frac{Jumlah\ kata\ kategori\ negatif}{Jumlah\ prior\ negatif} = \frac{1}{3}$

b. P("Parah"|Positif)= $\frac{Jumlah\ kata\ kategori\ positif}{Jumlah\ prior\ positif} = \frac{0}{1} = 0$

P("Parah"|Negatif)= $\frac{Jumlah\ kata\ kategori\ negatif}{Jumlah\ prior\ negatif} = \frac{1}{3}$

c. P("Ruang Hijau"|Pos)= $\frac{Jumlah\ kata\ kategori\ positif}{Jumlah\ prior\ positif} = \frac{1}{1} = 1$

P("RuangHijau"|Neg)= $\frac{Jumlah\ kata\ kategori\ negatif}{Jumlah\ prior\ negatif} = \frac{0}{3} = 0$

d. P("Minim"|Positif)= $\frac{Jumlah\ kata\ kategori\ positif}{Jumlah\ prior\ positif} = \frac{0}{1} = 0$

P("Minim"|Negatif)= $\frac{Jumlah\ kata\ kategori\ negatif}{Jumlah\ prior\ negatif} = \frac{1}{3}$

4. Membandingkan probabilitas secara keseluruhan

a. P(Positif | "Polusi Parah Ruang Hijau Minim") α P(Positif)*P("Polusi"|Positif)* P("Parah"|Positif)*

P(“Ruang Hijau”|Positif)* P(“Minim”|Positif) $\alpha \frac{1}{4} *$

$0 * 0 * 1 * 0$

b. P(Negatif | “Polusi Parah Ruang Hijau Minim”) α

P(Negatif)*P(“Polusi”|Negatif)* P(“Parah”|Negatif)*

P(“Ruang Hijau”|Negatif)* P(“Minim”|Negatif) $\alpha \frac{3}{4} *$

$\frac{1}{3} * \frac{1}{3} * 0 * \frac{1}{3}$

Berdasarkan hasil dari perbandingan probabilitas secara keseluruhan P(Negatif |”Polusi parah ruang hijau minim”) lebih tinggi nilainya dibandingkan keseluruhan P(Positif |”Polusi parah ruang hijau minim”). Maka dari itu dapat dikatakan dari perhitungan tersebut bahwa tweet “Polusi Jakarta itu memang parah , sekarang mau salahkan siapa? Ruang hijau sudah minim sekali dijakarta, sudah banyak bangunan beton, populasi kendaraan terus bertambah terutama sepeda motor, home industri dan pabrik banyak menyebabkan polusi air dan udara” termasuk dalam sentimen negatif.

Library yang akan digunakan untuk proses perhitungan klasifikasi yaitu *library sklearn* yang diantaranya dipakai untuk mengimport fungsi *MultinomialNB.* Ini merupakan model klasifikasi Naive Bayes yang digunakan untuk melatih dan menguji model. Selanjutnya, library ini juga digunakan untuk menyediakan berbagai matrix evaluasi untuk mengukur kinerja model klasifikasi diantaranya yaitu *accuracy_score, classification_report,* dan *confusion_matrix.*

Berikut *source code* yang diperlukan untuk mengimport *library-library* yang akan digunakan pada proses klasifikasi ditunjukkan pada gambar 4.44

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

Gambar 4. 44 Library Klasifikasi Naive Bayes

Proses yang dilakukan selanjutnya untuk klasifikasi menggunakan algritma Naive Bayes Classifier yaitu menginisialisasi model klasifikasi Naive Bayes menggunakan kelas *'MultinomialNB()'.* Kemudian, menggunakan metode *'fit()'* melatih model tersebut menggunakan data latih yaitu *X_train* dan *y_train*. Setelah model dilatih, dilakukan prediksi menggunakan data uji (*X_test)* menggunakan metode *predict().* Hasil prediksi yang telah didapatkan masuk dimasukan ke dalam variabel *y_pred.* Peneliti menghitung akurasi prediksi dengan membandingkan hasil prediksi dengan label sebenarnya dari data uji (*'y_test').* Akurasi dihitung menggunakan fungsi *accuracy_score()* yang hasilnya akan ditampilkan dalam bentuk presentase. Selanjutnya, mencetak laporan klasifikasi yang berisi berbagai metrik evaluasi seperti presisi, rexall, dan f1-score dengan memanggil fungsi *'classification_report()'* yang menerima argumen berupa label sebenarnya dan label prediksi dari data uji. Laporan klasifikasi ini memberikan gambaran tentang performa model Naive Bayes dalam melakukan klasifikasi data uji.

Peneliti melakukan perhitungan klasifikasi data uji menggunakan model Naive Bayes Classifier sebanyak 4 kali dengan membandingkannya berdasarkan pembagian data yang telah ditentukan. Pada prosesnya *source code* yang digunakan dalam perhitungan klasifikasi Naive Bayes menggunakan *script* yang sama. Perbedaan *source code* dari proses perhitungan tersebut hanya pada bagian awal yaitu pembagian data sesuai dengan perbandingan yang akan dilakukan perhitungan. Berikut tampilan *script* untuk metode klasifikasi dengan model Naive Bayes Classifier ditunjukkan pada gambar 4.45 berikut.

```
# Inisialisasi model klasifikasi Naive Bayes
nb_classifier = MultinomialNB()

# Latih model menggunakan data latih
nb_classifier.fit(X_train, y_train)

# Lakukan prediksi menggunakan data uji
y_pred = nb_classifier.predict(X_test)

# Hitung akurasi prediksi
accuracy = accuracy_score(y_test, y_pred)
print("Akurasi Naive Bayes:", accuracy*100)
```

Gambar 4. 45 Source code klasifikasi naive bayes

Perbandingan Data Latih & Data Uji	Hasil Akurasi NBC
90 : 10	80 %
80 : 20	71,7171 %
70 : 30	77,7027 %
60 : 40	76.6497 %

Tabel 4. 7 Hasil akurasi NBC

Pada tabel 4.7 di atas menunjukkan bahwa perhitungan klasifikasi menggunakan model Naive Bayes Classifier yang mendapatkan nilai akurasi tertinggi yaitu pada perbandingan 90 : 10 sebesar 80 %.

G. Klasifikasi Support Vector Machine

Model klasifikasi selanjutnya yang digunakan yaitu Support Vector Machine (SVM). Langkah awal yang dilakukan adalah dengan mengimport kelas 'SVC' (Support Vector Classifier) yang didapatkan dari modul 'sklearn.svm' untuk dilakukan proses inisialisasi pada model SVM. Berikut tampilan *script* yang diperlukan dalam perhitungan klasifikasi Support Vector Machine (SVM) ditunjukkan pada gambar 4.46 di bawah ini.

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

Gambar 4. 46 Library Klasifikasi SVM

Selanjutnya, proses inisialisasi dilakukan dengan pembentukan model SVM. Parameter yang digunakan dalam penelitian ini diantaranya yaitu parameter 'kernel' yang pada perhitungan ini menggunakan 'kernel linear' untuk memisahkan kelas-kelas yang berbeda. Setelah dilakukan inisialisasi dilanjutkan melatih model SVM menggunakan data latih pada variabel *X_train dan y_train* dengan memanggil menggunakan metode *'fit()'* . Setelah model dilatih, dengan menggunakan metode *'predict()'* dilakukan prediksi pada data uji (X_test). Hasil

dari prediksi yang didapatkan dimasukkan pada variabel *'y_pred'*. Perhitungan akurasi prediksi dilakukan dengan membandingkan hasil prediksi dengan label sebenarnya dari data uji atau *y_test* menggunakan fungsi *'accuracy_score()'* dan hasilnya akan ditampilkan dalam bentuk presentase. *Script* perhitungan klasifikasi menggunakan model SVM ditampilkan pada gambar 4.47 dibawah.

```
# Inisialisasi model SVM
svm_model = SVC(kernel='linear')

# Melatih model SVM
svm_model.fit(X_train, y_train)

# Membuat prediksi menggunakan model yang telah dilatih
y_pred = svm_model.predict(X_test)

# Menghitung akurasi prediksi
accuracy = accuracy_score(y_test, y_pred)
print("Akurasi:", accuracy*100)
```

Gambar 4. 47 Source code klasifikasi SVM

Perbandingan Data Latih & Data Uji	Hasil Akurasi SVM
90 : 10	72 %
80 : 20	71,7171 %
70 : 30	77,0270 %
60 : 40	73.6041 %

Tabel 4. 8 Hasil akurasi SVM

Dari *script* perhitungan klasifikasi menggunakan model SVM didapatkan nilai akurasi pada berbagai perbandingan

seperti pada tabel 4.8 tersebut yang menunjukkan bahwa klasifikasi SVM nilai akurasi tertinggi dengan perbandingan 70 : 30 yaitu sebanyak 77,0270 %

Pada klasifikasi SVM untuk perhitungan manualisasi menggunakan tiga data tweet pada tabel 4.9 sebagai berikut.

D1	Polusi Udara Mengancam, Kualitas Udara jakarta Belum Penuhi Hak Ekologis Anak! https://t.co/BIyTQ4Dt2W	[“polusi”, “udara”, “ancam” “kualitas”, “udara”, “jakarta”, “penuh”, “hak”, “ekologis”, “anak”]
D2	@budibongg Di Jakarta pohon bnyk yg ditebang gedung dimana mana jd polusi udara sgt mengancam	[“jakarta”, “pohon”, “tebang”, “gedung”, “mana”, “polusi”, “udara”, “ancam”]
D3	Tepat sekali nih gerakan #GotongRoyongBoyongPohon ini dilakukan. Salah satu tindakan nyata yang tepat untuk ngatasi polusi udara Jakarta ya ngelakuin gerakan Gotong Royong Boyong Pohon begini. ðŸ‘ https://t.co/CYo2wjibsP	[“gerak”, “salah”, “tindak”, “nyata”, “atasi”, “polusi”, “udara”, “jakarta”, “gerak”, “gotong”, “royong”, “pindah”, “pohon”]

Tabel 4. 9 Data Perhitungan Manual SVM

1. Data Latih :

   L1 = [“polusi”, “udara”, “ancam” “kualitas”, “jakarta”, “hak”, “ekologis”, “anak”]

   L2 = [“tindak”, “nyata”, “atasi”, “polusi”, “udara”, “jakarta”, “gerak”, “gotong”, “royong”, “pindah”, “pohon”]

   Data Uji :

   T1 = [“jakarta”, “pohon”, “tebang”, “gedung”, “mana”, “polusi”, “udara”, “ancam”]

2. Representasi data dengan pembobotan kata (TF-IDF)

   Pembobotan kata Data Latih :

Term	**TF-IDF**		**Data Uji**
	L1	**L2**	**T1**
polusi	0	0	0
udara	0,226	0,113	0,113
ancam	0,176	0	0,176
kualitas	0,477	0	0
jakarta	0	1	0
hak	0,477	0	0
ekologis	0,477	0	0
anak	0,477	0	0
tindak	0	0,477	0
nyata	0	0,477	0
atasi	0	0,477	0
gerak	0	0,352	0
gotong	0	0,477	0
royong	0	0,477	0

pindah	0	0,477	0
pohon	0	0,176	0,176
tebang	0	0	0,477
gedung	0	0	0,477
mana	0	0	0,477

Tabel 4. 10 TFIDF Perhitungan manual SVM

3. Melakukan kernelisasi menggunakan fungsi kernel linear yang didefinisikan $K(x_i, x_j) = (x_i . x_j)$

   X_1 =[0 0,22 0,17 0,47 0 0,47 0,47 0,47 0 0 0 0 0 0 0 0 0 0]

   X_2 =[0 0,11 0,17 0 0 0 0 0 0,47 0,47 0,47 0,35 0,47 0,47 0,47 0,17 0 0 0]

   $X_1 . X_1' = 0{,}960$

   $X_1 . X_2' = 0{,}0509$

   $X_2 . X_1' = 0{,}0531$

   $X_2 . X_2' = 1{,}951$

   Berdasarkan perhitungan di atas diperoleh untuk matriks kernel K adalah

$$K = \begin{bmatrix} 0{,}960 & 0{,}0590 \\ 0{,}0531 & 1{,}951 \end{bmatrix}$$

4. Substitusikan ke dalam persamaan berikut dengan memaksimalkan fungsi Dualitas Langrage Multipier (Ld) terhadap α

   $Max\ L_D = \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j\ K\ (x_i, x_{j,})$

   Syarat : $0 \leq \alpha_i \leq C$ dan $\sum_{i=1}^{N} \alpha_i y_i = 0$

   Vektor target : $y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ dengan $\alpha_1 - \alpha_2 = 0$

sehingga $\alpha_1 = \alpha_2 = \alpha$

Subtitusi ke persamaan

$L_D = (a_1 + a_2) - \frac{1}{2}(a_1 a_1 . 0{,}960 + a_1 a_2 . 0{,}0590 + a_2 a_1 . 0{,}0531 + a_2 a_2 . 1{,}951)$

$L_D = 2a - \frac{1}{2}(a^2 . 0{,}960 + a^2 . 0{,}0590 + a^2 . 0{,}0531 + a^2 . 1{,}951)$

$L_D = 2a - \frac{1}{2}(a^2 (0{,}960 + 0{,}0590 + 0{,}0531 + 1{,}951)$

$L_D = 2a - \frac{1}{2}(a^2 (3{,}0231)$

$L_D = 2a - 1{,}51155a^2$

$\frac{d(LD)}{da} = \frac{d}{da}(2a - 1{,}51155a^2)$

$\frac{d(LD)}{da} = 2 - 3{,}0231a$

$2 - 3{,}0231a = 0$

$2 = 3{,}0231a$

$a = 0{,}6615$

Dengan demikian, nilai $\alpha_1 = \alpha_2 = 0{,}6615$

5. Menghitung nilai w_i dan b

$w = \sum_{i=1}^{N} \alpha_i y_i x_i$

$w = \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2$

$w = 0{,}6615 . 1 . x_1 + 0{,}6615 . (-1) . x_2$

$w = 0{,}6615 x_1 - 0{,}6615 x_2$

$w =$

$0{,}6615 . [0\ 0{,}22\ 0{,}17\ 0{,}47\ 0\ 0{,}47\ 0{,}47\ 0{,}47\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0] -$

0,6615 .[0 0,11 0,17 0 0 0 0 0 0,47 0,47 0,47 0,35 0,47 0,47 0,47 0,17 0 0 0]

$w = [0\ 0{,}072\ 0\ 0{,}319\ 0\ 0{,}319\ 0{,}319\ 0{,}319 - 0{,}319 - 0{,}319 - 0{,}319 - 0{,}231 - 0{,}319 - 0{,}319 - 0{,}319 - 0{,}112\ 0\ 0\ 0]$

Menghitung nilai bias(b)

$$b = -\frac{w.\ x_i + w\ .\ x_j}{2}$$

$$b = -\frac{w.\ x_1 + w\ .\ x_2}{2}$$

$$b = -\frac{0{,}61536 + (-0{,}77245)}{2}$$

$$b = -\frac{(-0{,}15709)}{2} = 0{,}078545$$

6. Perhitungan Prediksi Data Uji

T1 = [0 0,113 0,176 0 0 0 0 0 0 0 0 0 0 0 0 0,176 0,477 0,477 0,477]

y = (w . T1) + b

y = [0 0,072 0 0,319 0 0,319 0,319 0,319 -0,319 -0,319 -0,319 -0,231 -0,319 -0,319 -0,319 -0,112 0 0 0] . [0 0,113 0,176 0 0 0 0 0 0 0 0 0 0 0 0 0 0,176 0,477 0,477 0,477] + 0,078545

y = (- 0,455936) + 0,078545

y = (- 0,377391)

7. Prediksi

Model prediksi dilakukan menggunakan persamaan 2.5 dan 2.6, maka dari itu berdasarkan data testing yang diperoleh sebesar y = - *0,377391* diklasifikasikan sebagai negatif. Data tweet “Di Jakarta pohon bnyk yg ditebang gedung dimana

mana jd polusi udara sgt mengancam" ini terprediksi secara benar dikarenakan data diberi label negatif dan hasil prediksi juga negatif.

H. Confusion Matrix

Pada penelitian ini peneliti selain menghitung nilai akurasi model juga menggunakan hasil uji *Confusion Matrix* untuk mengevaluasi penerapan model pada perhitungan klasifikasi. Pengujian *confusion matrix* digunakan untuk menggambarkan performa model klasifikasi pada set data uji. Pada tabel *confusion matrix* setiap baris mewakili kelas aktual dan setiap kolom mewakili kelas yang diprediksi oleh model. Pada diagonal utama dari matrix menunjukkan jumlah prediksi yang benar untuk setiap kelas, sementara sel lainnya menunjukkan jumlah kesalahan klasifikasi. Peneliti akan memvisualisasikan tabel confusion matrix dengan menggunakan fungsi *'plot_confusion_matrix'*. Langkah awal yang diperlukan dalam perhitungan confusion matrix yaitu mengimport fungsi evaluasi kinerja model klasifikasi tersebut dari modul *'sklearn.metrics'* yang telah dilakukan sebelumnya pada proses klasifikasi model. *Library* yang diperlukan untuk mendukung fungsi di atas dengan mengimport *library seaborn* untuk membuat visualisasi data statistik yang menarik dan informatif. Selain itu, mengimport *library matplotlib.pyplot* yang berfungsi untuk membuat

visualisasi data secara umum, termasuk plot, grafik, dan diagram. Berikut tampilan *source code* untuk menambahkan *library* yang dibutuhkan ditunjukkan pada gambar 4.48 berikut.

```
import seaborn as sns
import matplotlib.pyplot as plt
```

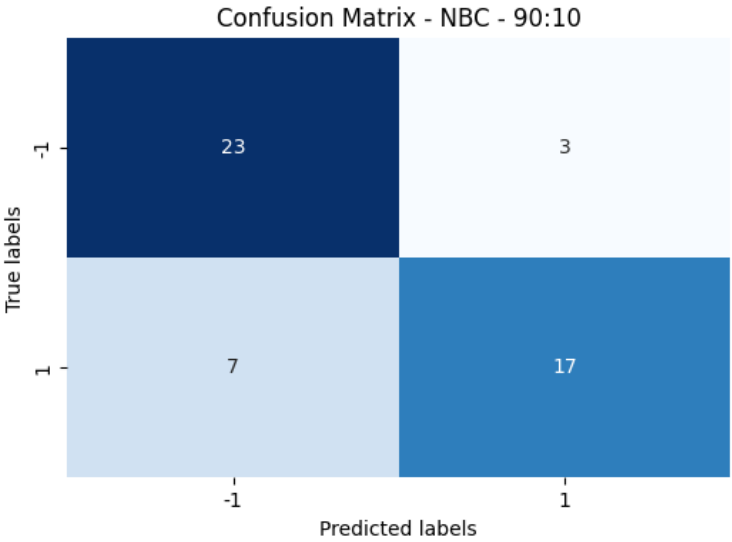
Gambar 4. 48 Library confusion matrix

Pada fungsi *'plot_confusion_matrix'* peneliti membuat *plot heatmap* dari *confusion matrix* dengan memanfaatkan *library seaborn* dan *matplotlib* di atas. Setiap sel di *heatmap* menunjukkan jumlah prediksi yang benar terdapat pada diagonal utama dan jumlah kesalahan lainnya yang terdapat pada sel lainnya. Anotasi dalam sel memberikan informasi nilai yang sesuai dalam *confusion matrix.* Peneliti membuat variabel *'labels'* yang berisi label unik dari kelas target yang digunakan untuk menandai sumbu x dan y pada *heatmap.* Pada hal ini nilai -1 menandakan bahwa itu data negatif. Sedangkan nilai 1 yang berarti bahwa itu data positif. Berikut tampilan *source code* untuk menampilkan *confusion matrix* yang diterapkan pada setiap perbandingan data yang ditentukan, ditunjukkan pada gambar 4.49 berikut dan tampilan *output* dari *script* berikut ditunjukkan pada gambar 4.50.

```
def plot_confusion_matrix(cm, labels):
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Reds', xticklabels=labels, yticklabels=labels, cbar=False)
    plt.xlabel('Predicted labels')
    plt.ylabel('True labels')
    plt.title('Confusion Matrix - SVM - 90:10')
    plt.show()
```

```
labels = df['Label'].unique()

nb_cm = confusion_matrix(y_test, y_pred)
plot_confusion_matrix(nb_cm, labels)
```

Gambar 4. 49 Source code visualisasi confusion matrix



Gambar 4. 50 Hasil visualisasi confusion matrix

Berdasarkan gambar 4.50 merupakan hasil visualisasi confusion matrix untuk metode Naïve Bayes Classifier pada perbandingan 90:10 dengan penjelasan sebagai berikut.

1. Nilai 23 yaitu jumlah nilai aktual yang bernilai negatif dan terprediksi benar negatif (*True Negative*)
2. Nilai 7 yaitu jumlah nilai aktual yang bernilai positif tetapi terprediksi negatif (*False Negative*)
3. Nilai 3 yaitu jumlah nilai aktual yang bernilai negatif namun terprediksi sebagai nilai positif (*False Positive)*

4. Nilai 17 yaitu jumlah nilai aktual yang bernilai positif dan terprediksi benar positif (*True Positive*)

I. Evaluasi

Setelah dilakukan pengujian *confusion matrix* menggunakan *script* sebelumnya. Peneliti melakukan perhitungan *performa* yang terdiri dari nilai accuraccy, *precison, recall,* dan *f1-score* dengan menggunakan persamaan 2.5, 2.6, 2.7, 2.8 yang dihitung berdasarkan hasil dari *confusion matrix* yang didapatkan.

Berikut pehitungan yang dilakukan pada setiap perbandingan dari kedua model klasifikasi.

1. Naïve Bayes Classifier (NBC)

a) Perhitungan dengan data latih 90 % dan data uji 10 %

Pada percobaan pertama digunakan perbandingan 90:10 yang akan diklasifikasikan menggunakan model Naive Bayes maka didapatkan hasil *confusion matrix* pada gambar 4.50 yang ditunjukkan pada tabel 4.11 berikut.

Tabel 4. 11 Confusion matrix NBC perbandingan data 90:10

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	23	3
	Positif	7	17

Berikut penerapan perhitungan performa tiap-tiap kelas manual dalam perbandingan 90:10 model klasifikasi NBC yang dapat dijabarkan sebagai berikut.

1) Kelas Positif

- Precision

$$Precision = \frac{TP}{TP + FP} = \frac{17}{17 + 3} = 0.85$$

Precision sebesar 0,85 menunjukkan bahwa dari sampel yang diprediksi sebagai positif oleh model, 85 % diantaranya benar-benar positif. Sisanya sebesar 15% diprediksi positif namun sebenarnya negatif.

- Recall

$$Recall = \frac{TP}{TP + FN} = \frac{17}{17 + 7} = 0{,}708 \approx 0{,}71$$

Recall sebesar 0,71 menunjukkan bahwa dari semua sampel yang benar-benar positif, model berhasil mengidentifikasi 71% diantaranya dengan benar, ini berarti terdapat 29% sampel positif yang tidak teridentifikasi.

- F1 Score

$$f1\ score = 2\ x \frac{Precision\ x\ Recall}{Precision + Recall}$$

$$= 2\ x \frac{0.85\ x\ 0{,}71}{0{,}85 + 0{,}71} = 0{,}7737 \approx 0{,}77$$

F1 Score adalah ukuran keseimbangan atau harmonisasi dari precision dan recall. Pada kelas positif memiliki nilai precision tinggi namun nilai recall lebih rendah yang menyebabkan f1 score menjadi lebih rendah. Nilai 0,77 menunjukkan bahwa model kurang optimal dalam mengidentifikasi semua sampel positif dan menghindari kesalahan dalam prediksi positif.

2) Kelas Negatif

- Precision

$$Precision = \frac{TN}{TN + FN} = \frac{23}{23 + 7} = 0{,}766 \approx 0{,}77$$

Precision sebesar 0,77 menunjukkan bahwa dari semua sampel yang diprediksi negatif oleh model, 77% diantaranya benar-benar negatif. Sedangkan 23% yang diprediksi negatif sebenarnya positif .

- Recall

$$Recall = \frac{TN}{TN + FP} = \frac{23}{23 + 3} = 0{,}884 \approx 0{,}88$$

Recall sebesar 0,88 menunjukkan bahwa dari semua sampel yang benar-benar

negatif, sebanyak 88% model berhasil mengidentifikasinya dengan benar. Sampel negatif yang tidak teridentifikasi sebesar 12%.

- F1 Score

$$f1\ score = 2\ x \frac{Precision\ x\ Recall}{Precision + Recall}$$

$$= 2\ x \frac{0.77\ x\ 0{,}88}{0{,}77 + 0{,}88} = 0{,}8213\ \approx 0{,}82$$

F1 Score sebesar 0,82 menunjukkan keseimbangan yang baik antara precision dan recall pada kelas negatif, memiliki nilai lebih tinggi dibandingkan f1 score pada kelas positif.

Perhitungan accuracy secara manual pada model klasifikasi Naive Bayes Classifier dengan pembagian data 90 : 10 dijelaskan pada perhitungan di bawah ini.

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN}\ x\ 100\%$$

$$= \frac{17+23}{17+23+3+7}\ x\ 100\% = 80\ \%$$

Setelah dilakukan perhitungan di atas didapatkan nilai *precision, recall, dan f1 score* dari tiap-tiap kelas yang dapat disimpulkan seperti pada tabel 4.12 dengan mengubahnya dalam bentuk presentase sebagai berikut.

Kelas	Precision	Recall	F1-score
Positif	85 %	71 %	77 %
Negatif	77 %	88 %	82 %
Accuracy	80 %		

Tabel 4. 12 Hasil perhitungan manual confusion matrix NBC

Hasil pengukuran performa tiap-tiap kelas diperoleh rata-rata nilai *precision, recall, dan f1 score* klasifikasi Naive Bayes Classifier dengan perbandingan 90 : 10 dapat di proses menggunakan kode program pada gambar 4.51 berikut.

```
# Inisialisasi model klasifikasi Naive Bayes
nb_classifier = MultinomialNB()

# Latih model menggunakan data latih
nb_classifier.fit(X_train, y_train)

# Lakukan prediksi menggunakan data uji
y_pred = nb_classifier.predict(X_test)

# Hitung akurasi prediksi
accuracy = accuracy_score(y_test, y_pred)
print("Akurasi Naive Bayes:", accuracy*100)

# Tampilkan laporan klasifikasi
print("Laporan Klasifikasi Naive Bayes - 90:10 :")
print(classification_report(y_test, y_pred))
```

Gambar 4. 51 Kode program perhitungan confusion matrix NBC

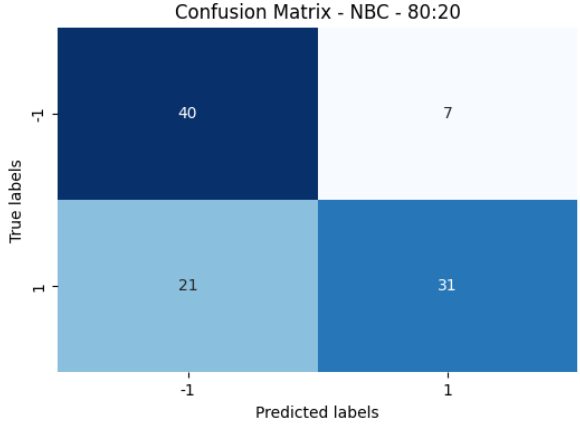
Berikut didapatkan hasil dari kode program pada gambar 4.51 di atas. Hasil pengukuran performa rata-rata nilai precision, recall, dan f1 score ditampilkan pada baris '*weighted avg'.* Sehingga dapat

ditunjukkan pada gambar 4.52 berikut merupakan hasil dari keseluruhan proses evaluasi model NBC pada pembagian data 90 : 10.

```
Akurasi Naive Bayes: 80.0
Laporan Klasifikasi Naive Bayes - 90:10 :
              precision    recall  f1-score   support

          -1       0.77      0.88      0.82        26
           1       0.85      0.71      0.77        24

    accuracy                           0.80        50
   macro avg       0.81      0.80      0.80        50
weighted avg       0.81      0.80      0.80        50
```

Gambar 4. 52 Hasil performa NBC perbandingan 90:10

b) Perhitungan dengan data latih 80 % dan data uji 20 %



Gambar 4. 53 Visualisasi confusion matrix nbc 80:20

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	40	7
	Positif	21	31

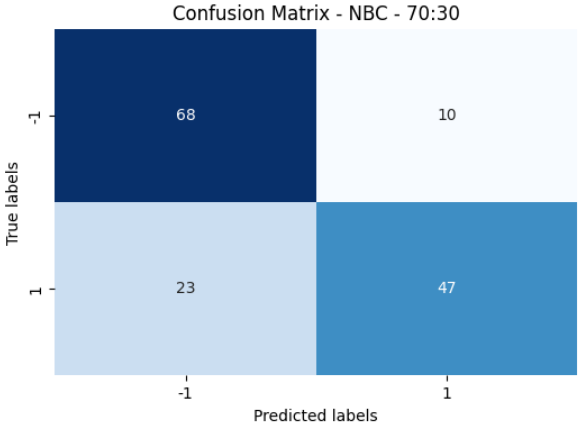
Tabel 4. 13 Confusion matrix NBC perbandingan data 80:20

Hasil dari *Confusion Matrix* untuk metode NBC dengan perbandingan 80:20 ditunjukkan padad gambar 4.53 untuk visualisasi confusion matrix dan pada tabel 4.13 penggambaran dalam bentuk tabel. Berdasarkan perhitungan nilai *precision, recall, f1 score* yang dilakukan pada perbandingan 90 : 10 dengan menggunakan langkah sama didapatkan hasil performa perhitungan klasifikasi dengan Naive Bayes Classifier pada pembagian data 80:20 dapat ditunjukkan pada gambar 4.54 berikut.

```
Akurasi Naive Bayes: 71.71717171717171
Laporan Klasifikasi Naive Bayes - 80:20 :
              precision    recall  f1-score   support

          -1       0.66      0.85      0.74        47
           1       0.82      0.60      0.69        52

    accuracy                           0.72        99
   macro avg       0.74      0.72      0.71        99
weighted avg       0.74      0.72      0.71        99
```

Gambar 4. 54 Hasil performa NBC perbandingan data 80:20

c) Perhitungan dengan data latih 70 % dan data uji 30 %



Gambar 4. 55 Visualisasi confusion matrix nbc 70:30

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	68	10
	Positif	23	47

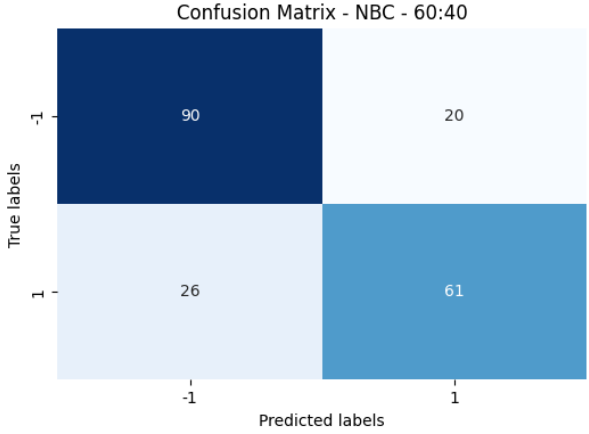
Tabel 4. 14 Confusion matrix NBC perbandingan data 70:30

Berdasarkan confusion matrix yang ditunjukkan pada gambar 4.55 dan tabel 4.14, dilakukan implementasi perhitungan yang sama seperti sebelumnya. Pada perhitungan performa dengan pembagian data 70 : 30 didapatkan hasil yang ditunjukkan pada gambar 4.56 di bawah ini.

```
Akurasi Naive Bayes: 77.7027027027027
Laporan Klasifikasi Naive Bayes - 70:30 :
              precision    recall  f1-score   support

          -1       0.75      0.87      0.80        78
           1       0.82      0.67      0.74        70

    accuracy                           0.78       148
   macro avg       0.79      0.77      0.77       148
weighted avg       0.78      0.78      0.77       148
```

Gambar 4. 56 Hasil performa NBC perbandingan data 70:30

d) Perhitungan dengan data latih 60 % dan data uji 40 %



Gambar 4. 57 Visualisasi confusion matrix nbc 60:40

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	90	20
	Positif	26	61

Tabel 4. 15 Confusion matrix NBC perbandingan data 60:40

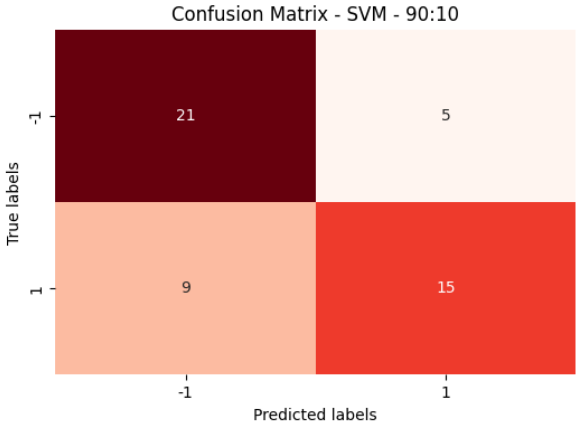
Berdasarkan confusion matrix yang ditunjukkan pada gambar 4.57 dan tabel 4.15, dilakukan implementasi perhitungan yang sama seperti sebelumnya. Hasil dari nilai *accuracy*, *precision, recall,* dan *f1-score* tiap-tiap kelas serta nilai rata-rata dari perhitungan performa tiap kelas ditunjukkan pada gambar 4.58 berikut.

```
Akurasi Naive Bayes: 76.6497461928934
Laporan Klasifikasi Naive Bayes - 60:40 :
              precision    recall  f1-score   support

          -1       0.78      0.82      0.80       110
           1       0.75      0.70      0.73        87

    accuracy                           0.77       197
   macro avg       0.76      0.76      0.76       197
weighted avg       0.77      0.77      0.77       197
```

Gambar 4. 58 Hasil performa NBC perbandingan data 60:40

2. Support Vector Machine (SVM)

a) Perhitungan dengan data latih 90% dan data uji 10 %



Gambar 4. 59 Visualisasi confusion matrix svm 90:10

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	21	5
	Positif	9	15

Tabel 4. 16 Confusion matrix SVM perbandingan 90:10

Pada tabel 4.16 dan gambar 4.59 di atas menunjukkan hasil *confusion matrix* yang akan dihitung nilai performa perhitungan klasifikasi Support Vector Machine (SVM) berupa *accuracy, precision, recall, dan f1-score* tiap-tiap kelas dengan perhitungan manualisasi dijabarkan sebagai berikut.

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \; x \; 100\%$$

$$= \frac{15 + 21}{15 + 21 + 5 + 9} \; x \; 100\% = 72 \; \%$$

Berikut penerapan perhitungan performa tiap-tiap kelas secara manual dalam perbandingan 90:10 model klasifikasi SVM yang dapat dijabarkan sebagai berikut.

1) Kelas Positif

- Precision

$$Precision = \frac{TP}{TP + FP} = \frac{15}{15 + 5} = 0.75$$

Precision sebesar 0,75 menunjukkan bahwa dari sampel yang diprediksi sebagai positif oleh model, 75 % diantaranya benar-benar positif. Sisanya sebesar 25% diprediksi positif namun sebenarnya negatif.

- Recall

$$Recall = \frac{TP}{TP + FN} = \frac{15}{15 + 9} = 0{,}62$$

Recall sebesar 0,62 menunjukkan bahwa dari semua sampel yang benar-benar positif, model berhasil mengidentifikasi 62% diantaranya dengan benar, ini berarti terdapat 38% sampel positif yang tidak teridentifikasi.

- F1 Score

$$f1\ score = 2\ x \frac{Precision\ x\ Recall}{Precision + Recall}$$

$$= 2\ x \frac{0.75\ x\ 0{,}62}{0{,}75 + 0{,}62} = 0{,}6788 \approx 0{,}68$$

F1 Score adalah ukuran keseimbangan atau harmonisasi dari precision dan recall. Pada kelas positif memiliki nilai precision tinggi namun nilai recall lebih rendah yang menyebabkan f1 score menjadi lebih rendah. Nilai 0,68 menunjukkan bahwa model kurang optimal dalam mengidentifikasi semua sampel positif dan menghindari kesalahan dalam prediksi positif.

2) Kelas Negatif

- Precision

$$Precision = \frac{TN}{TN + FN} = \frac{21}{21 + 9} = 0{,}70$$

Precision sebesar 0,70 menunjukkan bahwa dari semua sampel yang diprediksi negatif oleh model, 70% diantaranya benar-benar negatif. Sedangkan 30% yang diprediksi negatif sebenarnya positif .

- Recall

$$Recall = \frac{TN}{TN + FP} = \frac{21}{21 + 5} = 0{,}807 \approx 0{,}81$$

Recall sebesar 0,81 menunjukkan bahwa dari semua sampel yang benar-benar negatif, sebanyak 81% model berhasil mengidentifikasinya dengan benar. Sampel negatif yang tidak teridentifikasi sebesar 19%.

- F1 Score

$$f1\ score = 2\ x \frac{Precision\ x\ Recall}{Precision + Recall}$$

$$= 2\ x \frac{0.70\ x\ 0{,}81}{0{,}70 + 0{,}81} = 0{,}7466 \approx 0{,}75$$

F1 Score sebesar 0,75 menunjukkan keseimbangan yang baik antara precision dan recall pada kelas negatif, memiliki nilai lebih tinggi dibandingkan f1 score pada kelas positif.

Selanjutnya, perhitungan rata-rata nilai performa pada perhitungan dengan data latih 90 % dn data uji 10% dapat dilakukan menggunakan kode program pada gambar 4.60 berikut.

```
# Inisialisasi model SVM
svm_model = SVC(kernel='linear')

# Melatih model SVM
svm_model.fit(X_train, y_train)

# Membuat prediksi menggunakan model yang telah dilatih
y_pred = svm_model.predict(X_test)

# Menghitung akurasi prediksi
accuracy = accuracy_score(y_test, y_pred)
print("Akurasi:", accuracy*100)


# Tampilkan laporan klasifikasi
print("Laporan Klasifikasi SVM - 90:10 :")
print(classification_report(y_test, y_pred))
```

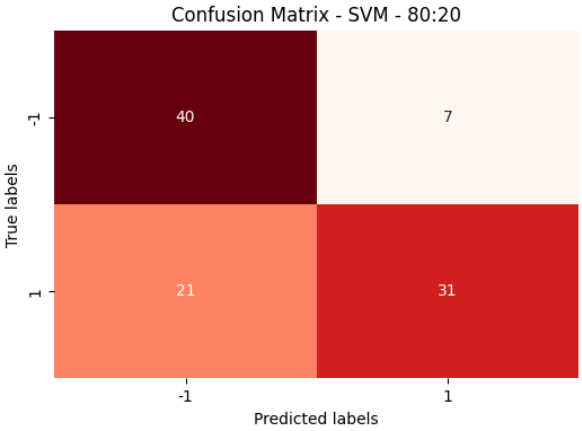
Gambar 4. 60 Kode program perhitungan confusion matrix SVM

Hasil dari perhitungan performa secara keseluruhan akan ditampilkan pada gambar 4.61 di bawah ini.

```
Akurasi: 72.0
Laporan Klasifikasi SVM - 90:10 :
              precision    recall  f1-score   support

          -1       0.70      0.81      0.75        26
           1       0.75      0.62      0.68        24

    accuracy                           0.72        50
   macro avg       0.72      0.72      0.72        50
weighted avg       0.72      0.72      0.72        50
```

Gambar 4. 61 Hasil performa SVM perbandingan 90:10

b) Perhitungan dengan data latih 80% dan data uji 20%



Gambar 4. 62 Visualisasi confusion matrix svm 80:20

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	40	7
	Positif	21	31

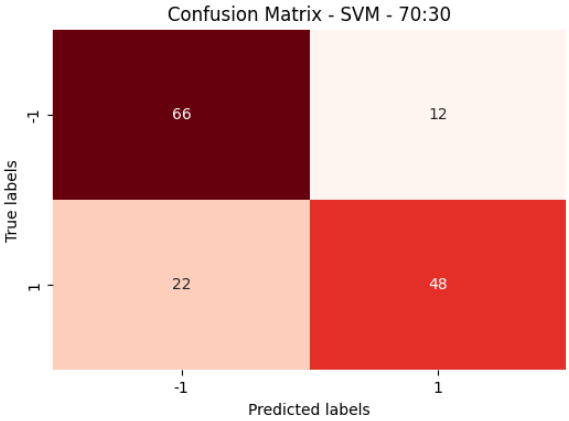
Tabel 4. 17 Confusion matrix SVM perbandingan 80:20

Berdasarkan gambar 4.62 dan tabel 4.17 di atas yang menunjukkan tabel *confusion matrix* yang akan dilakukan proses perhitungan performa berupa accuracy, precision, recall, dan f1 score tiap-tiap kelas dan keseluruhan yang akan di proses menggunakan kode program seperti sebelumnya dan akan menampilkan hasil yang akan ditunjukkan pada gambar 4.63 berikut.

```
Akurasi: 71.71717171717171
Laporan Klasifikasi SVM - 80:20 :
              precision    recall  f1-score   support

          -1       0.66      0.85      0.74        47
           1       0.82      0.60      0.69        52

    accuracy                           0.72        99
   macro avg       0.74      0.72      0.71        99
weighted avg       0.74      0.72      0.71        99
```

Gambar 4. 63 Hasil performa SVM perbandingan 80:20

c) Perhitungan dengan data latih 70% dan data uji 30%



Gambar 4. 64 Visualisasi confusion matrix svm 70:30

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	66	12
	Positif	22	48

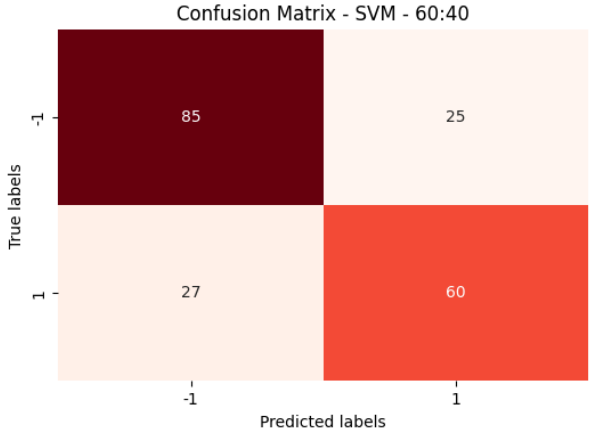
Tabel 4. 18 Confusion matrix SVM perbandingan 70:30

Berdasarkan tabel confusion matrix perhitungan klasifikasi dengan perbandingan 70:30 yang ditampilkan pada tabel 4.18 dan gambar 4.64 di atas. Pada perhitungan performa dengan pembagian data 70 : 30 didapatkan hasil yang ditunjukkan pada gambar 4.65 di bawah ini.

```
Akurasi: 77.02702702702703
Laporan Klasifikasi SVM - 70:30 :
              precision    recall  f1-score   support

          -1       0.75      0.85      0.80        78
           1       0.80      0.69      0.74        70

    accuracy                           0.77       148
   macro avg       0.78      0.77      0.77       148
weighted avg       0.77      0.77      0.77       148
```

Gambar 4. 65 Hasil performa SVM perbandingan 70:30

d) Perhitungan data latih dan 60% dan data uji 40%



Gambar 4. 66 Visualisasi confusion matrix svm 60:40

		Predicted Class	
		Negatif	**Positif**
True Class	**Negatif**	85	25
	Positif	27	60

Tabel 4. 19 Confusion matrix SVM perbandingan data 60:40

Dari gambar 4.66 dan tabel 4.19 di atas yang menunjukkan tabel *confusion matrix* didapatkan asil dari nilai *accuracy, precision, recall,* dan *f1-score* tiap-tiap kelas serta nilai rata-rata dari perhitungan performa tiap kelas yang ditunjukkan pada gambar 4.67 berikut.

```
Akurasi: 73.60406091370558
Laporan Klasifikasi SVM - 60:40 :
              precision    recall  f1-score   support

          -1       0.76      0.77      0.77       110
           1       0.71      0.69      0.70        87

    accuracy                           0.74       197
   macro avg       0.73      0.73      0.73       197
weighted avg       0.74      0.74      0.74       197
```

Gambar 4. 67 Hasil performa SVM perbandingan 60:40

Hasil dari perhitungan nilai rata-rata performa dari kedua model klasifikasi secara keseluruhan disimpulkan bahwa nilai tertinggi diperoleh model klasifikasi Naive Bayes Classifier dengan perbandingan 90 % data latih dan 10% data uji seperti dijelaskan dalam tabel 4.20 berikut ini.

	NBC				SVM			
	Accuracy	Precission	Recall	F1 Score	Accuracy	Precission	Recall	F1 Score
90:10	80 %	0,72	0,72	0,72	72%	0,81	0,80	0,80
80:20	71,71%	0,74	0,72	0,71	71,71%	0,74	0,72	0,71
70:30	77,70%	0,77	0,77	0,77	77,02%	0,78	0,78	0,77
60:40	76,64%	0,74	0,74	0,74	73,60%	0,77	0,77	0,77

Tabel 4. 20 Hasil rata-rata nilai performa NBC dan SVM

Berdasarkan evaluasi model yang telah dilakukan pada dua model klasifikasi yaitu Naive Bayes Classifier dan Support Vector Machine dapat dilihat nilai accuracy, *precision, recall* dan *f1 score* dari kedua model tersebut dengan berbagai pembagian data yang dilakukan. Hasil *accuracy* yang didapatkan dari perhitungan klasifikasi oleh dua model tersebut yang memiliki hasil *accuracy* tertinggi diperoleh oleh algoritma Naive Bayes Classifier dengan perbandingan data latih 90 % dan data uji 10% yang mendapatkan nilai *accuracy* sebesar 80 %.

Hasil dari nilai *precission, recall, dan f1 score* berupa angka desimal dengan rentang berkisar 0 – 1. Maka dari itu, semakin tinggi nilai yang diperoleh, semakin baik kinerja modelnya. Dari penelitian yang telah dilakukan didapatkan

untuk nilai *precision, recall,* dan *f1 score* pada tiap-tiap kelas yang memiliki nilai tertinggi diperoleh algoritma Naive Bayes Classifier dengan perbandingan data 90 : 10. Sehingga dari perolehan *precission* tersebut diartikan tingkat keberhasilan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna dengan proporsi label yang diprediksi dengan positif lebih tinggi daripada kelas negatif sebesar 0,85 untuk kelas positif. Sedangkan perhitungan *recall* untuk mengukur tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi yang bernilai negatif sebesar 0,88 lebih tinggi dibandingkan dengan menemukan informasi kembali yang bernilai positif sebesar 0,71. Pada perhitungan *f1 score* yang telah dilakukan digunakan untuk mengukur performa suatu model klasifikasi. Apabila semakin tinggi nilai *f1 score* maka semakin baik performa modelnya dalam mengklasifikasikan data. Dalam hal ini didapatkan nilai *f1 score* tertinggi diperoleh kelas positif sebesar 0,82 lebih tinggi dibandingkan kelas negatif sebesar 0,77.

BAB V

PENUTUP

A. Kesimpulan

Beberapa hasil penelitian yang telah dilakukan maka diperoleh beberapa kesimpulan sebagai berikut:

1. Dari penelitian ini dilakukan perhitungan klasifikasi sentimen menggunakan metode Naive Bayes Classifier dan Support Vector Machine dengan pengujian dilakukan sebanyak 4 kali setiap model klasifikasi. Tahapan yang dilakukan dimulai dengan melakukan pembersihan teks melalui tahapan *text preprocessing.* Selanjutnya dilakukan pembobotan kata dengan proses TFIDF. Pengujian dilakukan dengan melakukan perhitungan klasifikasi teks dengan metode NBC dan SVM serta menghitung evaluasi pada model berdasarkan 4 pembagian data yaitu 90:10, 80:20, 70:30, dan 60:40.
2. Sentimen yang diolah pada penelitian ini sebanyak 510 data *tweets.* Setelah dilakukan proses *text preprocessing,* data *tweet* menjadi 492 data dengan sentimen positif sebanyak 208 data *tweets,* sedangkan sentimen negatif sebanyak 284 data *tweets.*

3. Hasil performa pada metode Naive Bayes Classifier menghasilkan nilai akurasi lebih tinggi dibandingkan metode Support Vector Machine. Pada penelitian ini dihasilkan nilai akurasi tertinggi pertama diperoleh oleh metode NBC pada perbandingan 90:10 dengan nilai sebesar 80%. Perolehan nilai akurasi tertinggi selanjutnya oleh metode NBC pada perbandingan 70:30 sebesar 77,702%. Selanjutnya nilai akurasi pada metode SVM dengan perbandingan 70:30 sebesar 77,027%.
4. Hasil evaluasi pada metode Naive Bayes Classifier dan Support Vector Machine dihasilkan nilai rata-rata performa berupa *precission, recall,* dan *f1 score* secara keseluruhan dengan nilai tertinggi didapatkan oleh metode metode Naive Bayes Classifier pada pembagian data latih 90% dan data uji 10% sebesar 0,81 untuk nilai *precission,* nilai *recall* sebesar 0,80 dan nilai *f1 score* sebesar 0,80.

B. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran sebagai berikut:

1. Dalam penelitian ini hanya mengklasifikasikan data ke dalam sentimen positif dan negatif. Penelitian

selanjutnya diharapkan dapat dilakukan penelitian menggunakan sentimen positif, negatif, dan netral.

2. Dalam penelitian ini hanya menggunakan data sebanyak 500 data dari satu media sosial Twitter, pada penelitian selanjutnya diharapkan dapat menambahkan jumlah data dan dilakukan pada media sosial lainnya.

DAFTAR PUSTAKA

Aliyah Salsabila, N., Ardhito Winatmoko, Y., Akbar Septiandri, A., & Jamal, A. (2018). Colloquial Indonesian Lexicon. *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, 226–229. https://doi.org/10.1109/IALP.2018.8629151

Andrivet, M.(2023). *Twitter's Extreme Rebrand to X: A Calculated Risk or Puse Chaos?.* Diunduh di https://www.thebrandingjournal.com/ tanggal 22 Oktober 21 Oktober 2023

Ansori, Y., & Holle, K. F. H. (2022). Perbandingan Metode Machine Learning dalam Analisis Sentimen Twitter. *Jurnal Sistem Dan Teknologi Informasi (JustIN)*, *10*(4), 429. https://doi.org/10.26418/justin.v10i4.51784

Amalia, D. H., & Yustanti, W. (2021). Klasifikasi Buku Menggunakan Metode *Support Vector Machine* pada Digital Library. Journal of Informatics and Computer Science, 3(1), 55–61. https://opac.unesa.ac.id

Astriyani, M., Laela, I. N., Lestari, D. P., Anggraeni, L., & Astuti, T. (2023). Analisis Klasifikasi Data Kualitas Udara Dki Jakarta Menggunakan Algoritma C.45. *JuSiTik : Jurnal Sistem Dan Teknologi Informasi Komunikasi*, *6*(1), 36–41. https://doi.org/10.32524/jusitik.v6i1.790

Arifin, O. & Sasongko, T. B., 2018. Analisa Perbandingan Tingkat Performansi Metode *Support Vector Machine* dan *Naive Bayes Classifier* Uuntuk Klasifiaksi Jalur Minat SMA. Seminar Nasional Teknologi Informasi dan Multimedia, 2(1), pp. 67-72.

Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data *Twitter* BMKG Nasional. Jurnal TEKNO KOMPAK, 15(1), 131–145.

Faradhillah, N. Y. A. (2016). Analisis Sentimen Terhadap Kinerja Pelayanan Publik Di Kota Surabaya Berdasarkan Klasifikasi Komentar Di Media Sosial Dengan

Menggunakan Algoritma *Naive Bayes*. *Doctoral Dissertation, Institut Teknologi Sepuluh Nopember*, 1–198.

Fikri, Mujaddid Izzul, Trifebi Shina Sabrila, And Yufis Azhar. 2020. "Perbandingan Metode *Naive Bayes* Dan *Support Vector Machine* Pada Analisis Sentimen *Twitter*." Smatika Jurnal 10(02): 71–76.

Fitriyah, Nur, Budi Warsito, And Di Asih I Maruddani. 2020. "Analisis Sentimen Gojek Pada Media Sosial *Twitter* Dengan Klasifikasi *Support Vector Machine*(SVM)." Jurnal Gaussian 9 (3): 376-90

Fridayati, R. (2023). Analisis Sentimen Review Pengguna Aplikasi Photomath Dengan Menggunakan Metode *Support Vector Machine* (SVM). Universitas Lampung. Bandar Lampung

Hakim, N. S.(2021) Analisis Sentimen Persepsi Pengguna Myindihome Menggunakan Metode *Support Vector Machine* (Svm) Dan *Naive Bayes Classifier* (Nbc). Universitas Islam Indonesia. Yogyakarta

Herianto, "PENERAPAN TEXT-MINING UNTUK MENGIDENTIFIKASI PENGGUNA *TWITTER* TERHADAP FENOMENA PERAN DPR RI," Darma Persada, vol. VIII, no. 2, pp. 2088-060X, 2018.

Indraswari, L. D. 2023. *Pasang Surut Penanganan Polusi Udara Jakarta.* Diunduh di https://www.kompas.id/ tanggal 18 Oktober 2023

I. Taufik dan S.A.Pamungkas. (2018). Analisis Sentimen Terhadap Tokoh Publik Menggunakan Algoritma *Support Vector Machine* (Svm). Jurnal "LOG!K@," 8(1), 71–79.

Luthfanida, L. (2022). Analisis Sentimen Data *Twitter* Menggunakan Metode Naive Bayes Dan *Support Vector Machine* (Svm) Tentang Presiden Jokowi 3 Periode. Djtechno: Jurnal Teknologi Informasi, 3(1), 5–11. https://doi.org/10.46576/djtechno.v3i1.2143

Mas Pintoko, B., & Muslim, K. (2018). Analisis Sentimen Jasa Transportasi Online pada *Twitter* Menggunakan

Metode *Naive Bayes Classifier*. E-Proceeding of Engineering, 5(3), 8121–8130.

Meilany, S. A. (2022) Analisis Sentimen Opini Masyarakat Pengguna *Twitter* Terhadap Pariwisata Lampung Menggunakan *Support Vector Machine* Dan Naive Bayes. Universitas Lampung. Bandar Lampung

Mustofa, H., & Mahfudh, A. A. (2019). Klasifikasi Berita Hoax Dengan Menggunakan Metode Naive Bayes. *Walisongo Journal of Information Technology*, *1*(1), 1. https://doi.org/10.21580/wjit.2019.1.1.3915

Nuari, A.S. (2018) *Analysis sentimen pada media sosial menggunakan metode Naive Bayes Classifier tesis*. Informatics & Business Institute Darmajaya

Nurmalasari, D., Hermanto, T. I., & Nugroho, I. M. (2023). *Perbandingan Algoritma SVM , KNN dan NBC Terhadap Analisis Sentimen Aplikasi Loan Service*. *7*, 1521–1530. https://doi.org/10.30865/mib.v7i3.6427

Nomleni, P. (2015). Sentiment Analysis Menggunakan *Support Vector Machine* (Svm). *Seminar Nasional Teknologi Dan Komunikasi 2015*, *2015*(Sentika), 1–8.

Pradana, H. Y., Slamet, I., & Zukhronah, E. (2023). Analisis Sentimen Kinerja Pemerintahan Menggunakan Algoritma Nbc, Knn, Dan Svm. *Prosiding Simposium Nasional Multidisiplin (SinaMu)*, *4*, 114. https://doi.org/10.31000/sinamu.v4i1.7869

QORITA, A. K. (2022). *Analisis Sentimen Berbasis Aspek Pada Ulasan Tempat Wisata Diy*. https://dspace.uii.ac.id/handle/123456789/39202%0A https://dspace.uii.ac.id/bitstream/handle/123456789/39202/18523214.pdf?sequence=1

Raharjo, R. A., Sunarya, I. M. G., & Divayana, D. G. H. (2022). Perbandingan Metode *Naive Bayes Classifier* Dan *Support Vector Machine* Pada Kasus Analisis Sentimen Terhadap Data Vaksin Covid-19 Di *Twitter*. Elkom : Jurnal Elektronika Dan Komputer, 15(2), 456–464. https://doi.org/10.51903/elkom.v15i2.918

Rio Chandra. (2018). analisis-sentimen. https://github.com/ramaprakoso/analisissentimen/blob/master/kamus/kbba.txt. https://github.com/ramaprakoso/analisissentimen/blob/master/kamus/stopword.txt

Sains, F. (2019). ANALISIS ALGORITHMS *SUPPORT VECTOR MACHINE* DENGAN NAIVE BAYES KERNEL PADA KLASIFIKASI DATA. 6.

Salsabila, A. N. (2022). *Analisis Sentimen Pada Media Sosial Twitter Terhadap Tokoh Gus Dur Menggunakan Metode Naive Bayes Dan Support Vector Machine (SVM). Universitas Islam Negeri Syarif Hidayatullah. Jakarta*

Santoso, G. T. (2021). Analisis sentimen pada tweet dengan tagar #bpjsrasarentenir menggunakan metode support vectore machine (svm) skripsi.

Sholekha, I., Faqih, A., & Bahtiar, A. (2022). Sentiment Analysis of Public Opinion Covid-19 Vaccine Using *Naive Bayes* and Random Forest Methods. JURNAL TEKNIK INFORMATIKA, 15(1), 34–43. https://doi.org/10.15408/jti.v15i1.24847

Sodik, F., & Kharisudin, I. (2021). Analisis Sentimen dengan SVM , NAIVE BAYES dan KNN untuk Studi Tanggapan Masyarakat Indonesia Terhadap Pandemi Covid-19 pada Media Sosial *Twitter*. Prisma, 4, 628–634.

Sonawanse, S. & V. A. K. (2016). Teknik Analisis Sentimen Data *Twitter*: Survei. http://ai.stanford.

S. Samsir, A. Ambiyar, U. Verawardina, F. Edi, and R. Watrianthos, "Analisis Sentimen Pembelajaran Daring pada *Twitter* di Masa Pandemi COVID-19 Menggunakan Metode *Naive Bayes*," J. Media Inform. Budidarma, Vol. 5, No. 1, pp. 157–163, 2021, doi: http://dx.doi.org/10.30865/mib.v5i1.2580.

Suyanto. (2018). Machine Learning Tingkat Dasar dan Lanjut. Bandung: Informatika Bandung.

Syakuro, A. (2017). Pada Media Sosial Menggunakan Metode *Naive Bayes Classifier* (NBC) Dengan Seleksi Fitur

Information Gain (IG) Halaman Judul Skripsi Oleh : Abdan Syakuro. *Analisis Sentimen Masyarakat Terhadap E-Commerce Pada Media Sosial Menggunakan Metode Naive Bayes Classifier (NBC) Dengan Seleksi Fitur Information Gain (IG)*, 1–89. http://etheses.uin-malang.ac.id/11706/

Turmudi Zy, A., Adji Ardiansyah, L., & Maulana, D. (2021). Implementasi Algoritma *Naive Bayes* Dalam 98 Mendiagnosa Penyakit Angin Duduk. Jurnal Pelita Teknologi, 16(1), 52–65.

V. Ambasador Flores, Lie Jasa, dan Linawati. "Analisis Sentimen untuk Mengetahui Kelemahan dan Kelebihan Pesaing Bisnis Rumah Makan Berdasarkan Komentar Positif dan Negatif di Instagram". Majalah Ilmiah Teknologi Elektro. Vol 19. No 1 Jan-Jun 2020. DOI: https://doi.org/10.24843/MITE.2020.v19i01.P07

Wandani, A. (2021). Sentimen Analisis Pengguna *Twitter* pada Event Flash Sale Menggunakan Algoritma K-NN , Random Forest , dan Naive Bayes. 5(September), 651–665.

Wenty Dwi Yuniarti (2019). *Dasar-Dasar Pemrograman dengan Python*. Deepublish

Wibisono, A. B., & Fahrurozi, A. (2019). Perbandingan Algoritma Klasifikasi Dalam Pengklasifikasian Data Penyakit Jantung Koroner. *Jurnal Ilmiah Teknologi Dan Rekayasa*, *24*(3), 161–170. https://doi.org/10.35760/tr.2019.v24i3.2393

Widowati, Tanthy Tawaqalia, And Mujiono Sadikin. 2021. "Analisis Sentimen *Twitter* Terhadap Tokoh Publik Dengan Algoritma Naive Bayes Dan *Support Vector Machine*." Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer 11(2): 626–36.

Zidan, M. (2022). Analisis Sentimen Kenaikan Harga Bahan Bakar Minyak (BBM) Berdasarkan Respon Pengguna Media Sosial *Twitter* Di Indonesia Menggunakan Metode Naive Bayes. In Skripsi. Semarang: UIN Walisongo.

DAFTAR LAMPIRAN

LAMPIRAN 1 : Hasil Crawling Data Twitter

No	username	full_text
1	irfanwahidi60	Akhirnya terang benderang, gembar gembor polusi udara Jakarta hanyalah untuk akselerasi penjualan mobil motor listrik dan menaikan harga pertalite dgn menggati pertalite dgn katanya pertamax green ?" yg tentunya dgn harga yg lebih tinggi dari harga pertalite sekarang,, https://t.co/ypzl1vpOBN"
2	piotrj	ðŸš©ITU ðŸš©BUKAN ðŸš©KABUT Selama seminggu ada WFH, semprot air, hujan buatan, tapi hari ini polusi udara tinggi sampai gak kelihatan gedung2 Jakarta. Ada yang 30x di atas guideline WHO ðŸ˜±ðŸ˜± Tarumajaya - 171 ug/m3 Puri Indah - 149 ug/m3 Pakai masker hari ini!! Data dariâ€¦ https://t.co/77Pxt1klPs
3	cgtrisanjaya	@geloraco Betul. Sebaiknya pemerintah segera memberikan KTP buat si udara Polusi dan segera memerintahkan udara Polusi menghindari wilayah Jakarta.
4	limitfungsii	gila parah, jakarta krisis polusi udara, gue dri flyover bpkp liatnya asep tebel doang
5	Nofendra19	Polusi udara di Jakarta, lalu yang disalahkan ukm dan industri. Padahal yang paling besar penyebabnya adalah kendaraan bermotor. Hadeh na.....
6	tempodotco	Polusi Udara Jakarta Masih Tinggi Meski ASN WFH, Begini Respons Heru Budi #TempoMetro https://t.co/wVptZ6l3VH

7	oewlc	Disinyalir ada Konflik kepentingan antara @kemenkomarves LBP &amp @KementerianLHK @SitiNurbayaLHK yg justru menghambat proses perizinan AMDAL LNG Bali. Multiplier effect dalam hal kebijakan EBT yg ga jelas ini ya #PolusiMeningkatEBTdihambat &amp polusi udara di Jakarta jd menggila. https://t.co/UpipNfjtzL
8	DoankWarto	Memangnya selama 5 tahun ANIES menjabat GABENER DKI kualitas udara di DKI sudah bagus? Kalau Anies sudah punya jurus jitu mengatasi masalah polusi udara di Jakarta, mengapa tidak dia lakukan saat menjabat GABENER DKI Jakarta,kok baru sekarang koar koar? https://t.co/eLfqN6tZ6p
9	ISalutem	SANGSI PEMERINTAH SERIUS TANGANI POLUSI Putusan pengadilan dalam gugatan warga negara atas polusi udara Jakarta tak terlaksana. Pemerintah ogah dianggap bersalah. https://t.co/JADThkigZ9
10	Cerry0709	Meteri BUMN @erickthohir Yakin dengan LRT ini dapat menjadi salah satu solusi untuk mengurangi kemacetan ibu kota Jakarta.Selain itu juga akan mengurangi penggunaan jumlah kendaraan berbasis BBM sehingga akan mengurangi polusi udara.#GanjarErickDuetTerbaik https://t.co/5h9wZzCSYe
11	ZulFan22389096	@tvOneNews Warga Jakarta harus beralih dari mobil pribadi ke moda transportasi masal yang ramah lingkungan , kalau tidak sampai kpn pun polusi udara di jakarta tidak akan pernah selesai - selesai
12	Polusi_udara01	Polusi udara pagi hari ini pasti buruk. 4 jam lagi anak2 masuk sekolah dan akan ketiban musibah dr udara beracun made in jakarta. Kasihan bukan. Blum lagi orang

		dewasa produktif tp bokek krn perdagangan sedang sepi berangkat ke kerjaan mereka, diperjalanan bernafas dgn udara
13	Polusi_udara01	#Asean #Fifa #trending #southeastAsia #BakarSampah #iU17 #Ios17 #Iphone15 #JakartaðŸ‡®ðŸ‡© #AirPollution #ndonesia #PolusiDosaDkiJakarta 2:26am jakarta rank 4 dunia. Ada 3 lokasi yg polusi udara mengkuatirkan krn sangat beracun keadaannya. Apakah disana gak ada manusia yg huni?? https://t.co/1LmodZXmBy

.....

497	Cerry0709	Meteri BUMN @erickthohir menyebut transportasi LRT Jabodebek dapat menjadi solusi untuk mengurangi polusi dan kemacetan. Seperti diketahui, saat ini Jakarta dan sekitarnya menyandang status dengan kualitas udara yang buruk di dunia.#GanjarErickDuetTerbaik
498	sukaraja	Mengatasi Polusi Udara di Jakarta, Peran PLTU dan Teknologi Ramah Lingkungan https://t.co/uZJzWYck47 #BeritaJakarta #infojakarta #Jakarta
499	anies_ulani	Pemprov DKI akan Bentuk Satgas Tangani Polusi Udara Jakarta https://t.co/EIg1ZoP3xp https://t.co/S7GJbLwJKd
500	Polusi_udara01	Dalam setiap pendataan level polusi udara Jakarta selalu dgn kesungguhan dan kejujuran. Semoga data2 ini bermanfaat bagi seluruh penduduk jakarta, cendikiawan dan juga instansi pemerintah. Gunakanlah data ini agar kalian bs mendapatkan solusi tepat mengatasi maximal polusi udara

LAMPIRAN 2 : Tabel Labelling Data *Twitter*

No	Full_text	Sentimen			Sentimen Akhir	Label
		Mhs 1	Mhs 2	Mhs 3		
1	Akhirnya terang benderang, gembar gembor polusi udara Jakarta hanyalah untuk akselerasi penjualan mobil motor listrik dan menaikan harga pertalite dgn menggati pertalite dgn katanya pertamax green ?" yg tentunya dgn harga yg lebih tinggi dari harga pertalite sekarang,, https://t.co/ypzl1vpOBN"	Negatif	Negatif	Negatif	Negatif	-1
2	▶ ITU ▶ BUKAN ▶ KABUT Selama seminggu ada WFH, semprot air, hujan buatan, tapi hari ini polusi udara tinggi sampai gak kelihatan gedung2 Jakarta. Ada yang 30x di atas guideline WHO 😱😱 Tarumajaya - 171 ug/m3 Puri Indah - 149 ug/m3 Pakai masker hari ini!! Data dari… https://t.co/77Pxt1klPs	Negatif	Negatif	Negatif	Negatif	-1
3	@geloraco Betul. Sebaiknya pemerintah segera memberikan KTP buat si udara Polusi dan segera memerintahkan udara Polusi menghindari wilayah Jakarta.	Negatif	Negatif	Negatif	Negatif	-1
4	gila parah, jakarta krisis polusi udara, gue dri flyover bpkp liatnya asep tebel doang	Negatif	Negatif	Negatif	Negatif	-1
5	Polusi udara di Jakarta, lalu yang disalahkan ukm dan industri. Padahal yang paling besar penyebabnya adalah kendaraan bermotor. Hadeh na.....	Negatif	Negatif	Negatif	Negatif	-1
6	Polusi Udara Jakarta Masih Tinggi Meski ASN WFH, Begini Respons Heru Budi #TempoMetro https://t.co/wVptZ6l3VH	Negatif	Negatif	Negatif	Negatif	-1

7	Disinyalir ada Konflik kepentingan antara @kemenkomarves LBP & @KementerianLHK @SitiNurbayaLHK yg justru menghambat proses perizinan AMDAL LNG Bali . Multiplier effect dalam hal kebijakan EBT yg ga jelas ini ya #PolusiMeningkatEBTdihambat & polusi udara di Jakarta jd menggila. https://t.co/UpipNfjtzL	Negatif	Negatif	Negatif	Negatif	-1
8	Memangnya selama 5 tahun ANIES menjabat GABENER DKI kualitas udara di DKI sudah bagus? Kalau Anies sudah punya jurus jitu mengatasi masalah polusi udara di Jakarta, mengapa tidak dia lakukan saat menjabat GABENER DKI Jakarta, kok baru sekarang koar koar? https://t.co/eLfqN6tZ6p	Negatif	Negatif	Negatif	Negatif	-1
9	SANGSI PEMERINTAH SERIUS TANGANI POLUSI Putusan pengadilan dalam gugatan warga negara atas polusi udara Jakarta tak terlaksana. Pemerintah ogah dianggap bersalah. https://t.co/JADThkigZ9	Negatif	Negatif	Negatif	Negatif	-1
10	Meteri BUMN @erickthohir Yakin dengan LRT ini dapat menjadi salah satu solusi untuk mengurangi kemacetan ibu kota Jakarta. Selain itu juga akan mengurangi penggunaan jumlah kendaraan berbasis BBM sehingga akan mengurangi polusi udara.#GanjarErickDuetTerbaik https://t.co/5h9wZzCSYe	Positif	Positif	Positif	Positif	1

11	@tvOneNews Warga Jakarta harus beralih dari mobil pribadi ke moda transportasi masal yang ramah lingkungan , kalau tidak sampai kpn pun polusi udara di jakarta tidak akan pernah selesai - selesai	Positif	Positif	Positif	Positif	1
12	Polusi udara pagi hari ini pasti buruk. 4 jam lagi anak2 masuk sekolah dan akan ketiban musibah dr udara beracun made in jakarta. Kasihan bukan. Blum lagi orang dewasa produktif tp bokek krn perdagangan sedang sepi berangkat ke kerjaan mereka, diperjalanan bernafas dgn udara	Negatif	Negatif	Negatif	Negatif	-1
13	#Asean #Fifa #trending #southeastAsia #BakarSampah #iU17 #Ios17 #Iphone15 #JakartaðŸ‡®ðŸ‡© #AirPollution #ndonesia #PolusiDosaDkiJakarta 2:26am jakarta rank 4 dunia. Ada 3 lokasi yg polusi udara mengkuatirkan krn sangat beracun keadaannya. Apakah disana gak ada manusia yg huni?? https://t.co/1LmodZXmBy	Negatif	Negatif	Negatif	Negatif	-1
....						
503	Bener si langit jakarta membaik tapi knp harus ada produk yg di jual mencari cuan banget, gembar-gembor polusi udara buruk, tapi ujung-ujungnya jualan alat pemantau polusi. Di mana merah putihmu? https://t.co/xilR6ez4De	Negatif	Negatif	Negatif	Negatif	-1
504	Sumber polusi udara jakarta ini berasal dr lokasi yg sama setiap kali terjadi peningkatan.. bukan perkara yg sulit untuk menangkap sumber polusi udara	Negatif	Negatif	Positif	Negatif	-1

	jika pemerintah mau bertindak yaa ðŸ™ðŸ»					
505	Ambil contoh: polusi di jakarta udah di kondisi mencekam. Udara aja udah bahaya untuk dihirup. Di pemikiran w (gatau di orang2 lain), buat apa bangun masa depan yang belum tentu ada?	Negatif	Positif	Negatif	Negatif	-1
506	#Asean #Fifa #trending #southeastAsia #BakarSampah #iU17 #Ios17 #Iphone15 #JakartaðŸ‡®ðŸ‡© #AirPollution #ndonesia #PolusiDosaDkiJakarta 29/08/2023 Selasa Udara klp gading kembali tercemar oleh asap bakaran sampah plastik.. bener2 mengerikan semua keadaan ini ðŸ¤¢ https://t.co/2GPOyMCyfx	Negatif	Negatif	Negatif	Negatif	-1
507	Nah jokowi mengumpulkan menteri untuk membahas polusi udara di Jakarta, layaknya memanggil 'pelatih' untuk menghadapi pertandingan besar melawan asap tebal. Semoga kali ini strategi yang mereka susun bisa membawa kita meraih 'trofi' udara bersih. ðŸ˜	Positif	Positif	Positif	Positif	1
508	Polusi Udara di Jabodetabek, Menparekraf Sebut Kunjungan Wisatawan ke Jakarta Meningkat https://t.co/L69ShD5uLi	Negatif	Negatif	Negatif	Negatif	-1
509	yang masih nyemprot parfum biar ga bau badan menambah polusi udara jakarta #tanyanetizen	Negatif	Negatif	Negatif	Negatif	-1
510	Upaya pemerintah mengatasi polusi udara di DKI Jakarta bakal dilakukan lewat gedung-gedung pencakar langit. https://t.co/i4SGc2pWRX	Positif	Positif	Positif	Positif	1

LAMPIRAN 3 : Tabel colloquial-indonesian-lexicon

slang	formal
woww	wow
aminn	amin
met	selamat
netaas	menetas
keberpa	keberapa
eeeehhhh	eh
kata2nyaaa	kata-katanya
hallo	halo
kaka	kakak
ka	kak
daah	dah
aaaaahhhh	ah
yaa	ya
smga	semoga
slalu	selalu
amiin	amin
kk	kakak
trus	terus
kk	kakak
sii	sih
nyenengin	menyenangkan
bgt	banget

gemess	gemas
akuuu	aku
jgn	jangan
yaa	ya
udah	sudah
gitu	begitu
aja	saja
gemesiin	menggemaskan
menyenangkn	menyenangkan
rb	ribu
akau	aku
saranin	menyarankan
nemuin	menemukan

.....

dlm	dalam
wktu	waktu
hr	hari
gatau	enggak tau
gataunya	enggak taunya
gtau	enggak tau
gatau	enggak tau
fans2	fan-fan
gaharus	enggak harus

LAMPIRAN 4 : Kbba.txt

7an tujuan
@ di
ababil abg labil
abis habis
acc accord
ad ada
adlah adalah
adlh adalah
adoh aduh
afaik as far as i know
aha tertawa
ahaha haha
aing saya
aj saja
aja saja
ajep-ajep dunia gemerlap
ajj saja
ak saya
aka dikenal juga sebagai
akika aku
akko aku
....
puyeng pusing
songongsombong
taik kotoran
soft halus
setting atur
angis nangis
benarjujur benar
benarjujur jujur
sayan sayang
mgkin mungkin

LAMPIRAN 5 : Slangword.txt

&:dan
+:tambah
/:atau
=:sama dengan
ababil:anak ingusan
abal2:palsu
abal:palsu
ad:ada
akooh:aku
alay:norak
albm:album
ampe:sampai
anjir:waw
anyway:ngomong-ngomong
app:aplikasi
aq:aku
asap:secepatnya
asn: aparatur sipil negara
ato:atau
atw:atau
ava:foto profil
...
wtf:apa-apaan
wfh:kerja dari rumah
x:kali
y:ya
yakali:iya mana mungkin
yg:yang
yth:yang terhormat

LAMPIRAN 6 : Stopword-id.txt

ada
adalah
adanya
adapun
agak
agaknya
agar
akan
akankah
akhir
akhiri
akhirnya
aku
akulah
amat
amatlah
anda
andalah
antar
antara
antaranya
apa
.....

wah
wahai
waktu
walau
walaupun
yaitu
yakin
yakni
yang

RIWAYAT HIDUP

A. Identitas Diri

1. Nama Lengkap : Isti Nur Azizah
2. Tempat & Tgl. Lahir : Banyumas, 18 Juni 2002
3. Alamat Rumah : Purwodadi, RT 04 RW 03 Kecamatan Tambak, Kabupaten Banyumas
4. HP : 085713865257
5. E-mail : istinurazizah1862@gmail.com

B. Riwayat Pendidikan

1. Pendidikan Formal:

a. TK Pertiwi Purwodadi
b. SD Negeri 2 Purwodadi
c. SMP Negeri 1 Sumpiuh
d. SMA Negeri 1 Sumpiuh
e. Universitas Islam Negeri Walisongo Semarang

2. Pendidikan Non Formal

a. –
b. –
c. –

C. Prestasi Akademik

a. –
b. –

D. Karya Ilmiah

a. –
b. –

Semarang, Juni 2024

Isti Nur Azizah
NIM : 2008096041