ANALISIS SENTIMEN OPINI PUBLIK DI MEDIA SOSIAL X TERHADAP PASANGAN GANJAR PRANOWO DAN MOHAMMAD MAHFUD MD JELANG PILPRES 2024 MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER

SKRIPSI

Diajukan untuk Memenuhi Tugas Akhir dan Melengkapi Syarat Guna Memperoleh Gelar Sarjana Strata Satu (S-1) dalam Teknologi Informasi

Diajukan Oleh:
AINUN FATIMAH
NIM : 2008096016

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
2023/2024

PERNYATAAN KEASLIAN

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Ainun Fatimah
NIM : 2008096016
Jurusan : Teknologi Informasi

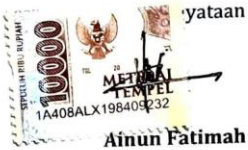
Menyatakan bahwa skripsi yang berjudul:

Analisis Sentimen Opini Publik Di Media Sosial X Terhadap Pasangan Ganjar Pranowo Dan Mohammad Mahfud Md Jelang Pilpres 2024 Menggunakan Metode Naïve Bayes Classifier

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

Semarang, 26 Juni 2024

Pernyataan



Ainun Fatimah

NIM : 2008096016

PENGESAHAN

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI WALISONGO
FAKULTAS SAINS DAN TEKNOLOGI
Alamat: Jl. Prof. Dr. Hamka Km 1, Ngaliyan Semarang Telp.024-76433366 Senarang 50185
Email: fst@walisongo.ac.id, Web: http://fst.walisongo.c.id

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Analisis Sentimen Opini Publik Di Media Sosial X Terhadap Pasangan Ganjar Pranowo dan Mohammad Mahfud MD Jelang Pilpres 2024 Menggunakan Naïve Bayes Classifier

Nama : **Ainun Fatimah**

NIM : 2008096016

Jurusan : Teknologi Informasi

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

Semarang, 26 Juni 2024

DEWAN PENGUJI

Penguji I

Nur Cahyo Hendro Wibowo, S.T., M.Kom
NIP. 197312222006041001

Penguji II

Adzhal Arwani Mahfudh, M.Kom
NIP. 199107032019031006

Penguji III

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

Penguji IV

Siti Nur'aini, M.Kom
NIP. 198401312018012001

Pembimbing I

Dr. Masy Ari Ulinuha, M.T
NIP. 198108122011011007

Pembimbing II

Adzhal Arwani Mahfudh, M.Kom
NIP. 199107032019031006

iii

NOTA DINAS

NOTA DINAS

Semarang, 26 Juni 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	:	Analisis Sentimen Opini Publik Di Media Sosial X Terhadap Pasangan Ganjar Pranowo dan Mohammad Mahfud MD Jelang Pilpres 2024 Menggunakan Naïve Bayes Classifier
Nama	:	**Ainun Fatimah**
NIM	:	2008096016
Jurusan	:	Teknologi Informasi

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

Wassalamu'alaikum. Wr. Wb.

Pembimbing I

Dr. Masy Ari Ulinuha, M.T.

NIP. 198108122011011007

iv

NOTA DINAS

NOTA DINAS

Semarang, 26 Juni 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	:	Analisis Sentimen Opini Publik Di Media Sosial X Terhadap Pasangan Ganjar Pranowo dan Mohammad Mahfud MD Jelang Pilpres 2024 Menggunakan Naïve Bayes Classifier
Nama	:	**Ainun Fatimah**
NIM	:	2008096016
Jurusan	:	Teknologi Informasi

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

Wassalamu'alaikum. Wr. Wb.

Pembimbing II

Adzhal Arwani Mahfudh, M.Kom.

NIP. 199107032019031006

v

LEMBAR PERSEMBAHAN

Dengan mengucapkan Puji dan Syukur Alhamdulillah penulis ucapkan kepada Allah Swt, penulis dapat menyelesaikan karya tulis sebagai laporan tugas akhir ini dengan baik. Karya tulis ini penulis persembahkan untuk:

1. Bapak Tita Pribadi dan Ibu Eny Setiawati selaku orang tua dari penulis.
2. Imam Wahid selaku kakak penulis.
3. Annisa Nur Aini, Maulana Ibrahim, Muhammad Yusuf, Salahuddin al-ayyubi dan Siti Afifah Az-zahra selaku adik penulis.
4. Seluruh dosen Jurusan Teknologi Informasi.
5. Sahabat dan teman-teman seperjuangan khususnya Jurusan Teknologi Informasi Angkatan 2020.
6. Almamater Universitas Islam Negeri Walisongo Semarang.

MOTTO

"Direndahkan dimata manusia, ditinggikan dimata Tuhan, *Prove Them Wrong*"

"Gonna fight and don't stop, until you are proud"

"Selalu ada harga dalam proses. Nikmati saja lelah-lelah itu. Lebarkan lagi rasa sabar itu. Semua yang kau investasikan untuk menjadikan dirimu serupa yang kau impikan. Mungkin tidak akan selalu berjalan lancar. Tapi, gelombang-gelombang itu yang nanti bisa kau ceritakan."
(Boy Chandra)

ABSTRAK

Ganjar Pranowo dan Mohammad Mahfudh MD telah dinyatakan sebagai bakal calon presiden 2024. Opini masyarakat menjadi sumber informasi berharga dalam menganalisis sentimen terhadap kedua tokoh ini. Namun, analisis sentimen manual menghadapi kendala seperti keterbatasan tenaga manusia, ketidakstabilan emosi, dan waktu yang diperlukan. Oleh karena itu, penelitian ini memanfaatkan *machine learning* untuk mengklasifikasikan sentimen terhadap Ganjar Pranowo dan Mohammad Mahfudh MD. Tujuan penelitian ini adalah mengevaluasi kinerja metode *Naïve Bayes Classifier* dalam klasifikasi sentimen tersebut. Tahapan pemrosesan data meliputi *cleansing*, *case folding*, tokenisasi, normalisasi, *stopword removal,* dan *stemming*. Dataset yang digunakan terdiri dari 1000 tweet yang telah dilabelkan menggunakan lexicon InSet dengan bantuan mahasiswa Universitas Mulawarman jurusan Sastra Bahasa, menghasilkan tweet positif dan tweet negatif. Data kemudian dibagi menjadi 80% untuk data latih dan 20% untuk data uji. Pembobotan fitur dilakukan menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency)* sebelum digunakan untuk melatih model Naïve Bayes. Model ini dievaluasi menggunakan *confusion matrix*. Hasil penelitian menunjukkan bahwa metode *Naïve Bayes Classifier* dengan Lexicon Inset memberikan performa dengan *accuracy* sebesar 89%, *precission* 89%, *recall* 89%, dan *f1-score* 84%. Sementara itu, *Naïve Bayes Classifier dengan* ahli bahasa memberikan performa dengan *accuracy* sebesar 99%, *precission* 99%, *recall* 99%, dan *f1-score* 99%. Penelitian ini memberikan kontribusi penting dalam analisis sentimen.

Kata Kunci: Analisis Sentimen, Naïve Bayes Classifier, TF-IDF, Pemrosesan Teks.

KATA PENGANTAR

Puji syukur kehadirat Allah Swt, karena atas berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "**Analisis Sentimen Opini Publik Di Media Sosial X Terhadap Pasangan Ganjar Pranowo Dan Mohammad Mahfud Md Jelang Pilpres 2024 Menggunakan Metode Naïve Bayes Classifier"** yang merupakan salah satu syarat guna memperoleh gelar sarjana strata satu (S-1) dalam teknologi informasi.

Selama proses penyelesaian skripsi tidak terlepas dari bantuan berbagai pihak yang memberikan bantuan dan dukungan dalam bentuk materi, moral, motivasi juga inspirasi. Oleh karena itu dalam kesempatan ini penulis ingin menyampaikan rasa terima kasih yang setulus-tulusnya kepada:

1. Bapak Prof. Dr. Nizar, M.Ag, selaku Plt. Rektor Universitas Islam Negeri Walisongo Semarang.
2. Bapak Prof. 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 Masy Ari Ulinuha, M.T, dan Bapak Adzhal Arwani Mahfudh, M.Kom, selaku Dosen Pembimbing skripsi

saya yang selalu memberikan dukungan, arahan, bimbingan serta motivasi dalam pelaksanaan skripsi hingga pembuatan skripsi ini.

5. Seluruh dosen program studi Teknologi Informasi khususnya, serta dosen dan pegawai di lingkungan Universitas Islam Negeri Walisongo Semarang.
6. Bapak tercinta, Tita Pribadi. Beliau memang tidak sempat merasakan pendidikan sampai bangku perkulihan, namun beliau dapat mendidik, mendoakan, memberi semangat dan motivasi tiada henti kepada penulis sehingga penulis dapat menyelesaikan pendidikannya sampai sarjana.
7. Ibu tersayang, Eny Setiawati. Terima kasih sebesar-besarnya penulis sampaikan kepada beliau atas segala bentuk bantuan, dukungan, semangat dan doa yang diberikan selama ini. Terima kasih atas nasihat yang diberikan meski pikiran kita tidak sejalan. Ibu menjadi pengingat dan penguat yang paling hebat. Terima kasih, ibu.
8. Kakak kesayangan penulis Imam Wahid, S.Pd yang selalu memberikan dukungan moril dan material, memotivasi dan mendoakan penulis.
9. Tidak lupa ucapan kasih sayang kepada adik-adik penulis Annisa Nur Aini, Maulana Ibrahim, Muhammad

Yusuf, Salahuddin Al-Ayyubi dan Siti Afifah Az-zahra yang selalu menghibur penulis selama proses penulisan skripsi yang cukup lelah ini, terima kasih adik-adik yang sudah menemani proses demi proses sampai di titik sekarang.

10. Kepada pemilik NPM 11201089 yang telah menjadi sosok rumah tempat melepaskan segala keluh kesah, terima kasih atas segala usahanya dalam memberikan hal yang baik untuk penulis, serta memberikan semangat, doa, motivasi, dan menemani setiap proses penyusunan skripsi. Terima kasih telah menjadi bagian penting dalam perjalanan penulis hingga saat ini.
11. Semua pihak yang tidak dapat disebutkan satu per satu yang terlibat dalam penyusunan skripsi ini.

Akhir kata, semoga segala kebaikan dan ketulusan diberikan mendapatkan balasan yang setimpal oleh Allah SWT. Semoga skripsi ini bisa bermanfaat bagi para pembaca dan bisa dijadikan bahan rujukan untuk melakukan penelitian selanjutnya.

Semarang, 26 Juni 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Pada zaman yang modern ini, media sosial mempunyai peranan penting sebagai media kampanye pemilu. Hal ini dikarenakan media sosial memberikan kemudahan kepada penggunanya untuk mengakses berita politik terbaru dan opini masyarakat yang tidak dimuat dalam media cetak maupun media penyiaran (Lestari, A. R., et al., 2017). Selain itu, media sosial juga memudahkan penggunanya untuk menyampaikan opini mereka tentang berita politik terbaru. Salah satu media sosial yang paling sering digunakan untuk penyampaian opini ataupun media kampanye adalah X. Hal ini dikarenakan kelebihan media sosial X yang mudah diakses, jumlah pengikut tidak terbatas, dan jumlah karakter yang digunakan hanya 280 karakter sehingga mampu menyampaikan maksud dengan singkat, padat, dan jelas (Lestari, A. R., et al., 2017).

Dalam menyampaikan opini dalam media sosial, tentu akan lebih baik jika dilakukan dengan memperhatikan aturan yang ada dalam syariat Islam. Dalam Islam, ada beberapa aturan yang harus diperhatikan ketika ingin menyampaikan pendapat kepada orang lain. Salah satu aturan tersebut adalah harus menginformasikan kebenaran berdasarkan fakta dan

juga tidak merekayasa atau memanipulasi fakta. Hal ini dijelaskan dalam Qur'an Surat Al-Hajj ayat 30 berikut:

ذٰلِكَ وَمَنْ يُّعَظِّمْ حُرُمٰتِ اللّٰهِ فَهُوَ خَيْرٌ لَّهٗ عِنْدَ رَبِّهٖۗ وَاُحِلَّتْ لَكُمُ الْاَنْعَامُ اِلَّا مَا يُتْلٰى عَلَيْكُمْ فَاجْتَنِبُوا الرِّجْسَ مِنَ الْاَوْثَانِ وَاجْتَنِبُوْا قَوْلَ الزُّوْرِ

Artinya : *Demikianlah (perintah Allah). Dan barangsiapa mengagungkan apa-apa yang terhormat di sisi Allah maka itu adalah lebih baik baginya di sisi Tuhannya. Dan telah dihalalkan bagi kamu semua binatang ternak, terkecuali yang diterangkan kepadamu keharamannya, maka jauhilah olehmu berhala-berhala yang najis itu dan jauhilah perkataan-perkataan dusta.*(Q.S. Al-Hajj(22) ayat 33)

Selain itu, dalam menyampaikan pendapat juga harus dilakukan secara bijaksana dan tidak menyakiti hati orang lain. Hal ini seperti dijelaskan dalam Qur'an Surat An-Nisa' ayat 9 berikut:

وَلْيَخْشَ الَّذِيْنَ لَوْ تَرَكُوْا مِنْ خَلْفِهِمْ ذُرِّيَّةً ضِعٰفًا خَافُوْا عَلَيْهِمْ فَلْيَتَّقُوا اللّٰهَ وَلْيَقُوْلُوْا قَوْلًا سَدِيْدًا

Artinya : *Dan hendaklah takut kepada Allah orang-orang yang seandainya meninggalkan dibelakang mereka anak-anak yang lemah, yang mereka khawatir terhadap (kesejahteraan) mereka. Oleh sebab itu hendaklah mereka bertakwa kepada Allah dan hendaklah mereka mengucapkan perkataan yang benar qaulan sadida.*

Pasangan Pilpres Ganjar Pranowo dan Mohammad Mahfud MD dinyatakan oleh Koalisi PDIP sebagai bakal calon presiden 2024 pada tanggal 18 Oktober 2023 (Bbc News Indonesia, 2023). Pengumuman ini memicu animo masyarakat, yang membagikan pandangan mereka mengenai

pasangan ini di platform media sosial X. Pada bulan November 2022, Ganjar Pranowo menjadi tokoh politik yang paling banyak diperbincangkan di media sosial dengan 389.787 mentions (Rahman, A., 2022). Pendapat masyarakat ini menjadi sumber data berharga yang dapat digunakan untuk menganalisis sentimen terhadap pasangan ini sebagai bakal calon presiden 2024. Analisis sentimen adalah proses untuk menentukan apakah pendapat atau tanggapan terhadap suatu topik mencerminkan sentimen yang positif atau negatif. Analisis sentimen dilakukan untuk menentukan apakah pendapat atau tanggapan pada suatu topik mengarah pada sentimen positif atau negatif.

Analisis sentimen melibatkan penentuan apakah opini atau tanggapan terhadap suatu topik cenderung bersifat positif atau negatif. Meskipun analisis sentimen dapat dilakukan secara manual, tugas ini menjadi lebih sulit ketika jumlah opini yang harus dievaluasi sangat banyak. Oleh karena itu, pendekatan yang lebih efisien adalah menggunakan ilmu *Natural Language Processing* (NLP) dan *machine learning*. NLP adalah sebuah disiplin ilmu yang terkait dengan pemahaman dan pemrosesan bahasa manusia oleh komputer. Salah satu tugas penting dalam NLP adalah klasifikasi sentimen, di mana mesin diajarkan untuk mengenali dan mengekstrak opini dalam teks (Lisangan, et al., 2022). Salah satu metode *machine*

learning yang dapat digunakan dalam klasifikasi sentimen adalah *Naïve Bayes Classifier*. *Naïve Bayes Classifier* adalah metode klasifikasi yang sederhana dan efisien. Metode ini bekerja dengan mengasumsikan bahwa probabilitas dari suatu kelas (misalnya, sentimen positif atau negatif) tidak bergantung pada probabilitas kata-kata yang ada dalam teks. Kelebihan dari *Naïve Bayes Classifier* adalah kecepatan dan kesederhanaannya, yang dapat menghasilkan tingkat akurasi yang cukup tinggi dalam analisis sentimen (Astari, N. M. A. J., et al., 2020). Dengan *Naïve Bayes Classifier*, peneliti dapat mengklasifikasikan teks ke dalam kategori sentimen yang relevan, seperti positif atau negatif, berdasarkan karakteristik kata-kata dalam teks tersebut. Metode ini telah terbukti efektif dalam analisis sentimen dan dapat membantu dalam memahami pendapat masyarakat tentang suatu topik secara lebih efisien (Harpizon, H., et al., 2022).

Penelitian terkait analisis sentimen terhadap tokoh politik telah menjadi subjek yang menarik dalam beberapa tahun terakhir. Salah satu metode yang umum digunakan untuk melakukan analisis sentimen adalah *Naïve Bayes Classifier*. Contoh penelitian yang mencerminkan penggunaan metode ini adalah studi yang dilakukan oleh Soer dan Sutrisno pada tahun 2022. Mereka memanfaatkan *Naïve Bayes Classifier* untuk menganalisis opini dan sentimen terhadap Ridwan Kamil, yang

menjabat sebagai Gubernur Jawa Barat. Hasil penelitian mereka menunjukkan tingkat akurasi sekitar 84,38% (Soer, D. & Sutrisno, S., 2022). Penelitian lain yang relevan dilakukan oleh Fatchan dan Sugeng pada tahun 2021, yang berfokus pada analisis sentimen terhadap Tri Rismaharini, seorang Menteri Sosial. Dalam penelitian ini, mereka menggunakan Algoritma *Naïve Bayes* dan mencapai tingkat akurasi sekitar 90,33%, dengan presisi sekitar 77,7%, dan recall sekitar 99,9% (Fatchan, M., & Sugeng, H., 2021).

Berdasarkan pemaparan di atas, penelitian ini menerapkan metode *Naïve Bayes Classifier* pada analisis sentimen terhadap Ganjar Pranowo dan Mohammad Mahfud MD sebagai pasangan pemilihan presiden 2024. Perbedaan penelitian ini dengan penelitian terdahulu adalah penggunaan data penelitian yang terdiri dari tweet yang secara khusus terkait dengan Ganjar Pranowo dan Mohammad Mahfud MD sebagai pasangan pemilihan presiden 2024. Penelitian ini bertujuan untuk mengevaluasi kinerja metode *Naïve Bayes Classifier* dalam mengklasifikasikan sentimen terhadap Ganjar Pranowo dan Mohammad Mahfud MD sebagai pasangan pemilihan presiden 2024. Dataset yang digunakan terdiri dari 375 data tweet yang telah dikategorikan menjadi positif atau negatif. Dalam penelitian ini, dibandingkan dengan

perhitungan manual oleh peneliti menggunakan rumus *naïve bayes classifier.*

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang disampaikan, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana penerapan Naïve Bayes Classifier untuk mengklasifikasikan opini publik pada media sosial X dalam kasus pasangan Ganjar Pranowo dan Mohammad Mahfud MD jelang pilpres 2024 ?
2. Bagaimana hasil pengujian tingkat akurasi, *Precision, Recall, dan F1-Score* dalam mengklasifikasi opini publik terhadap pasangan Ganjar Pranowo dan Mohammad Mahfud MD jelang Pemilihan Presiden 2024 menggunakan metode Naïve Bayes Classifier ?

C. Tujuan Penelitian

Adapun tujuan penelitian berdasarkan rumusan masalah adalah sebagai berikut:

1. Mengklasifikasikan komentar publik positif atau negatif masyarakat terhadap pasangan Ganjar Pranowo dan Mohammad Mahfud MD jelang Pemilihan Presiden 2024 menggunakan metode Naïve Bayes Classifier pada media sosial X.
2. Menguji ketepatan hasil klasifikasi tweet.

D. Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan masalah yaitu sebagai berikut:

1. Model klasifikasi yang digunakan adalah Naïve Bayes Classifier.
2. Data yang digunakan dalam penelitian ini adalah opini masyarakat dalam Bahasa Indonesia yang didapatkan dari media sosial X dengan teknik *Crawling* API Twitter menggunakan modul Tweet-Harvest.
3. Kata kunci yang digunakan dalam teknik *Crawling* API Twitter adalah "*Ganjar Pranowo Mahfud until:2024-01-31 since:2023-10-18 lang:id*".
4. Pengumpulan dan analisis data terbatas pada periode tertentu yang mencakup masa jelang Pemilihan Presiden 2024. Informasi yang terkandung dalam tweet diambil dalam rentang waktu 18 oktober 2023 hingga 31 januari 2024 dan Bahasa indonesia untuk memastikan relevansi dengan konteks politik yang sedang berlangsung.
5. *Tools* yang digunakan dalam penelitian ini adalah Google Colab 2024 dan Modul Tweet-Harvest.
6. Hasil dari penelitian ini berupa data sentimen yang dikategorikan dalam dua kelas sentimen yaitu kelas positif dan kelas negatif, serta analisis interpretasi data yang diperoleh dari data tersebut.

7. Evaluasi terhadap metode *Naïve Bayes Classifier* akan Dibandingkan dengan perhitungan manual oleh peneliti menggunakan rumus *naïve bayes classifier.*

E. Manfaat Penelitian

Adapun setelah mengetahui tujuan dari penelitian ini, manfaat yang akan didapat dalam penelitian ini adalah:

1. Memahami tanggapan masyarakat terhadap pasangan Ganjar Pranowo dan Mohammad Mahfud MD jelang Pemilihan Presiden 2024 di Media Sosial X. Penelitian ini dapat membantu mengidentifikasi apakah tanggapan masyarakat terhadap pasangan Ganjar Pranowo dan Mohammad Mahfud MD jelang Pemilihan Presiden 2024 secara umum positif atau negatif. Hal ini dapat memberikan informasi yang berguna bagi Masyarakat.
2. Menyediakan informasi kepada peneliti selanjutnya mengenai kinerja metode Naïve Bayes Classifier dalam menganalisis opini publik di media sosial X.

BAB II

LANDASAN PUSTAKA

A. Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah cabang *Artificial Intelligence* (AI) yang berfokus pada pemrosesan bahasa alami. Bahasa alami merupakan bahasa yang banyak digunakan oleh manusia untuk berkomunikasi satu sama lain (Cucus et al., 2019). NLP digunakan untuk proses penerjemahan bahasa alami ke bahasa komputer untuk dapat diolah data tersebut menghasilkan suatu analisis tertentu. Pada proses NLP pengolah kata, data yang akan diolah dapat disaring dengan cepat dan akurat sehingga menghasilkan satuan kata yang baik (Furqan, M., & Shidqi, M. N., 2023).

Penggunaan *Natural Language Processing* (NLP) dalam penelitian ini tentu dirancang untuk menganalisis opini publik di media sosial X mengenai pasangan Ganjar Pranowo dan Mohammad Mahfud MD jelang Pilpres 2024. Dengan memanfaatkan teknologi NLP, data *tweet* yang berkaitan dengan kedua tokoh politik tersebut akan diubah menjadi format yang dapat dipahami oleh mesin. Proses ini bertujuan untuk mendapatkan wawasan mendalam tentang persepsi, dukungan, kritik, atau berbagai pandangan lain yang mungkin

muncul dari masyarakat melalui platform media sosial ini. Selanjutnya, analisis NLP akan memfasilitasi pemahaman tentang dinamika, tren, dan potensi isu-isu yang muncul terkait elektabilitas dan popularitas pasangan Ganjar pranowo dan Mohammad Mahfud MD dalam konteks Pilpres 2024.

B. Text Mining

Text mining adalah penambangan data yang memecahkan masalah yang berkaitan dengan kebutuhan informasi melalui penerapan teknik penambangan data, pembelajaran mesin, pemrosesan bahasa alami, penyajian informasi dan manajemen informasi penambangan teks mencakup pemrosesan awal dokumen seperti klasifikasi teks, ekstraksi informasi, dan ekstraksi kata. Metode ini digunakan untuk mengekstraksi informasi dari sumber data dengan mengidentifikasi dan memeriksa pola yang menarik (Engelhart, M. D., & Moughamian, H., 1968).

Text mining adalah bidang baru yang berkembang yang berupaya mengekstraksi informasi yang bermakna dari teks bahasa alami. Ini dapat secara kasar dicirikan sebagai proses menganalisis teks untuk mengekstraksi informasi yang berguna untuk tujuan tertentu. Teks memiliki fungsi untuk menyampaikan informasi atau opini faktual, dan insentif untuk

secara otomatis mengekstraksi informasi dari teks (Witten, I. H., 2004).

Text mining secara umum merupakan teknik atau konsep yang digunakan untuk menganalisis teks. Dalam *text mining* terdapat beberapa hal yang mendukung seperti pemrosesan bahasa alami (NLP), tokenisasi, *text cleaning*, *stemming* and *lemmatization*, representasi *vector* hingga menjadikannya sebagai analisis sentimen.

C. Analisis Sentimen

Analisis sentimen atau penambangan pendapat adalah bidang luas pemrosesan bahasa alami, linguistik komputasi, dan penambangan teks, yang tujuannya adalah untuk menemukan pendapat, perasaan, evaluasi, sikap, dan perasaan pembicara atau penulis tentang suatu topik, produk, layanan atau organisasi, individu, atau aktivitas lainnya (Dale, 2010). Analisis sentimen juga dapat mengungkapkan kesedihan emosional, kegembiraan atau kemarahan. Peneliti dapat mencari pendapat tentang produk, merek, atau orang dan melihat apakah mereka memiliki ulasan positif atau negatif yang diambil datanya secara online (Sarjana et al., 2017).

Analisis sentimen merupakan salah satu NLP, yakni suatu proses agar komputer bisa berinteraksi dengan bahasa

manusia dengan beberapa proses di dalamnya. Dalam analisis sentimen masih sangat memungkinkan untuk komputer tidak bisa membedakan antara ulasan positif atau negatif. Jika peneliti mengambil data dari sosial media, karena biasanya bahasa yang digunakan dalam sosial media sangat ambigu untuk dimengerti komputer. Sehingga memungkinkan untuk beberapa ratus data yang akan terambil mungkin hanya puluhan data. Namun, cara lain bisa dapat meningkatkan akurasi analisis seperti pelabelan ulasan secara manual dan dihitung analisisnya dengan menggunakan machine learning (Furqan, M., & Shidqi, M. N., 2023).

D. Media Sosial X

Twitter yang berganti nama menjadi X sejak Juli 2023 (David, E., & Wes, T., 2023) adalah media sosial daring dan layanan jejaringan sosial yang dioperasikan oleh perusahaan Amerika Serikat, X Corp., penerus Twitter, Inc. Di X, pengguna terdaftar dapat memposting teks, gambar, dan video (Conger, K., & Kate, C., 2023). Suka (like), posting ulang (retweet), memberi komentar dan mengutip posting (quote posts), hingga mengirim pesan langsung (DM) ke pengguna terdaftar lainnya. Pengguna berinteraksi dengan X melalui peramban

atau perangkat lunak Frontend seluler, atau secara terprogram melalui antarmuka pemrograman aplikasi (API).

Media sosial X diciptakan oleh Jack Dorsey, Noah Glass, Biz Stone dan Evan Williams pada bulan maret 2006 dan diluncurkan pada bulan juli tahun itu. Perusahan induk sebelumnya, Twitter, Inc., berbasis di San Francisco, California dan memiliki lebih dari 25 kantor di seluruh dunia. Pada tahun 2012, lebih dari 100 juta pengguna telah men-tweet 340 juta tweet per hari, dan layanan ini menangani rata-rata 1,6 miliar permintaan pencarian per hari. Pada tahun 2013, media sosial X merupakan salah satu dari sepuluh situs web yang paling sering dikunjungi dan digambarkan sebagai "SMS-nya Internet" (Monte, D., & Leslie, 2009). Pada awal tahun 2019, media sosial X memiliki lebih dari 330 juta pengguna aktif bulanan (Molina M., & Brett, (2017). Dalam praktiknya, sebagian besar tweet di-tweet oleh sebagai kecil pengguna (Wojcik, et al., 2019). Pada tahun 2020, diperkirakan sekitar 48 juta akun (15% dari semua akun) adalah akun palsu (*fake accounts*) (Rodriguez-Ruiz, et al., 2020).

Pada tanggal 27 Oktober 2022, tokoh bisnis Elon Musk mengakuisisi Twitter senilai US$44 miliar dan mendapatkan kendali atas platform tersebut (Isaac, et al., 2022). Sejak

akuisisi tersebut, platform ini dikritik karena memfasilitasi pertumbuhan konten yang mengandung ujaran kebencian (Sato, & Mia, 2022). Linda Yaccarino, mantan kepala bagian penjualan iklan untuk NBCUniversal, menggantikan posisi Elon Musk sebagai Ceo pada tanggal 5 juni 2023 (Fier, & Sarah, 2023). Pada bulan juli 2023, Musk mengumumkan bahwa Twitter akan diganti namanya menjadi X dan logo burung Twitter akan dihapus secara bertahap (Valinsky, & Jordan, 2023).

E. Python

Python merupakan salah satu bahasa pemrograman yang dapat digunakan untuk segala jenis analisis data. Didalam python terdapat banyak *library* yang membantu *text preprocessing* seperti *text folding, tokenizing, convert emoticon, stemming* dan lain-lain sebagaimana dalam analisis sentimen diperlukan adanya pemrosesan teks (Syah, H., & Witanti, A., 2022).

Python juga memiliki kapabilitas untuk mendukung klasifikasi menggunakan berbagai algoritma, termasuk Naïve Bayes, yang menjadi fokus utama dalam penelitian ini. Algoritma Naïve Bayes telah dipilih untuk analisis data dalam penelitian ini. Untuk mendukung pelaksanaan penelitiaan,

berbagai *library* khusus Python telah digunakan. Seperti *Pandas*, *NumPy*, *Matplotlib* dan *Seaborns* (sns), serta beberapa library tambahan yang harus di *install* seperti *Google API Client Library for Python*, *Scikit-Learn*(sklearn), *Natural Language Toolkit* (NLTK*), Journey to the Center Of Python Machine Learning* (jcopml), *Sastrawi*.

Google API Client Library for Python merupakan *library* yang digunakan dalam *python* agar dapat mengakses data yang ada di google. *Scikit-learn* merupakan *library* yang mencakup banyak algoritma untuk pembelajaran klasifikasi ataupun *regresi*, serta memiliki banyak algoritma yang menunjang dalam pemrosesan data (Gridin, I., 2022). NLTK merupakan *library* yang digunakan dalam *komputasi linguistik*. NLTK mencakup pemrosesan bahasa alami berupa statistik, simbolik dan hubungan bertautan dengan copra (Rifano, E. J., et al., 2020). *Library jcompl* digunakan untuk pendekatan *workflow* yang terstruktur. *Library sastrawi* mencakup pemrosesan bahasa alami yang berhubungan dengan bahasa yang digunakan untuk *stemmer* yang digunakan untuk mengatasi permasalahan perubahan kata berimbuhan menjadi kata dasar (Rosid, M. A., et al., 2020).

F. Pelabelan

Pelabelan di sini bermaksud melabeli sentimen pada setiap komentar apakah data termasuk ke dalam sentimen positif atau negatif. Dalam proses pelabelan sentimen komentar ini, peneliti akan menentukan nilai sentimen positif apabila di dalam komentar terdapat kata-kata yang bermakna afirmasi ataupun kata-kata positif yang lebih dominan dari kata negatifnya.

Dalam proses pelabelan seharusnya bisa dilakukan dengan memanfaatkan polarity dengan library textblob yang ada pada python. Namun, dikarenakan pada kasus penelitian ini menggunakan bahasa Indonesia yang kompleks dan banyak data yang tidak menggunakan bahasa baku sehingga hasil percobaan program tidak memunculkan hasil yang valid pada nilai sentimen. Dalam proses pelabelan manual seharusnya dilakukan oleh pakar bahasa ataupun seseorang yang ahli di bidangnya, dan setidaknya membutuhkan dua orang atau lebih dalam pendiskusian mengenai pelafalannya untuk menghindari nilai subjektif dalam melakukan sentimen terhadap data (Imron, 2019).

Pada pelabelan sentimen penelitian ini, peneliti didampingi oleh salah satu mahasiswa jurusan sastra bahasa untuk membantu proses pelabelan manual ini. Namun, dalam

proses pelabelan manual ini membutuhkan waktu yang cukup lama karena jumlah datanya yang banyak.

G. Split Data

Split Data adalah membagi data menjadi data *training* atau data latih dan data *testing* atau data uji dengan menggunakan *split validation*. *Split validation* dilakukan dengan jumlah data testing diambil 20% dari data *training*. Pengambilan data dilakukan secara random dengan batuan *library python* (Turmudi Zy, A., et al., 2021). *Split Data* dilakukan dengan membagi data *training* dan data *testing* dengan perbandingan 80;20. Pembagian data dengan perbandingan tersebut merupakan strategi yang paling sederhana dan paling umum digunakan (Joseph V. R., & Vakayil, A., 2022).

H. Crawling

Web crawling atau *web Scraping* adalah ekstraksi otomatis data dari situs web menggunakan perangkat lunak. Ini adalah proses yang sangat penting dalam bidang-bidang seperti intelijen bisnis saat ini. Pengikisan web adalah teknik yang memungkinkan peneliti mengekstrak data terstruktur dari teks seperti *Hypertext Markup Language* (HTML). Pengikisan web sangat berguna dalam situasi di mana data tidak disediakan dalam format yang dapat dibaca mesin seperti

Javascript Object Notation (JSON) atau *Extensible Markup Language* (XML). Dengan mengumpulkan data melalui pengikisan web, harga dapat diperoleh dari situs web ritel hampir secara waktu nyata dan informasi tambahan dapat diberikan (Khder, M. A., 2021).

Crawling data adalah proses pengumpulan data secara otomatis dari berbagai sumber di web menggunakan perangkat lunak yang disebut *web crawler* atau *spider*. *Web crawler* bertugas untuk mengunjungi berbagai halaman web, mengidentifikasi pautan dan konten, dan mengambil informasi yang diinginkan untuk diolah lebih lanjut (Khder, M. A., 2021).

I. Text Preprocessing

Text *Preprocessing* merupakan level pertama dari *text processing*, yang berfungsi untuk mengubah format dokumen sesuai kebutuhannya menjadi data terstruktur agar dapat diproses lebih lanjut dalam proses *text mining* (Sarjana et al., 2017). Tahap *text preprocessing* klasifikasi data. *Preprocessing* dalam *text mining* cukup rumit, karena aturan penulisan kalimat dan pembentukan afiks dalam bahasa indonesia berbeda. Adapun beberapa kata imbuhan yang dapat merubah makna adalah sebagai berikut:

1. Prefiks: Prefiks adalah morfem yang ditambahkan di awal sebuah kata untuk mengubah makna atau jenis kata tersebut. Prefiks biasanya terdiri dari satu atau lebih huruf atau fonem. Contoh prefiks dalam bahasa indonesia adalah “di-“ dalam kata “diambil” atau “ber-“ dalam kata “berlari”. Prefiks dapat mengubah verba menjadi nomina, adjektiva menjadi verba, dan sebagainya (Chaer, A., & Agustina, L., 2010).
2. Sufiks: Sufiks adalah morfem yang ditambahkan di akhir sebuah kata untuk mengubah makna atau jenis kata tersebut. Sufiks juga terdiri dari satu atau lebih huruf atau fonem. Contoh sufiks dalam bahasa Indonesia adalah “-kan” dalam kata “membaca” atau “-an” dalam kata “penulisan”. Sufiks dapat mengubah verba menjadi verba kausatif, nomina menjadi adjective, dan sebagainya (Aronoff, M., & Fudeman, K., 2011).
3. Infiks: Infiks adalah morfem yang dimasukkan di tengah-tengah sebuah kata untuk mengubah makna atau jenis kata tersebut. Infiks umumnya terdiri dari satu atau lebih huruf atau fonem. Infiks biasanya digunakan dalam bahasa-bahasa tertentu, seperti bahasa Tagalog atau bahasa Jawa. Contoh infiks

dalam bahasa Indonesia adalah “-el-“ dalam kata “kemilau” atau “-me-“ dalam kata “gemetar”. Infiks dapat mengubah nomina menjadi verba, verba menjadi verba kausatif, dan sebagainya (Aronoff, M., & Fudeman, K., 2011).

4. Konfiks: Konfiks adalah kombinasi dari prefiks dan sufiks yang digunakan bersama-sama untuk membentuk sebuah kata. Prefiks dan sufiks dalam konfiks tidak dapat digunakan secara terpisah. Contoh konfiks dalam bahasa Indonesia adalah “peng-“ dan “-an” dalam kata “pengarahan”. Konfiks dapat mengubah nomina menjadi verba atau adjectiva, atau mengubah verba menjadi nomina (Aronoff, M., & Fudeman, K., 2011).

Text preprocessing adalah proses menghilangkan *noise* dan *unwanted attribute*. Langkah awalnya adalah menghapus teks yang tidak relevan, kemudian mengisi bagian yang kosong (*missing value*) dan menghapus kolom yang tidak berhubungan (Ratniasih, N. L., et al., 2023). Dalam text preprocessing terdapat beberapa tahapan seperti:

1. *Cleansing*

   Cleansing adalah tahapan pembersihan teks, seperti menghapus tanda baca, menghapus enter, mengubah emoji menjadi *string*, menghapus URL dan sebagainya.

2. *Case Folding*

   Case folding, adalah proses mengubah semua karakter dalam teks menjadi huruf kecil atau huruf besar, tergantung pada aturan yang ditentukan (Akbar et al., 2021). *Case folding* digunakan untuk menghapus perbedaan kasus (besar kecil huruf) dalam teks agar dapat dilakukan pemrosesan teks yang konsisten dan seragam.

3. *Tokenizing*

   Tokenizing adalah proses merubah teks menjadi token-token atau unit terkecil dari teks. Token merupakan unit dasar dalam pemrosesan teks, seperti kata, frasa, simbol, atau karakter tertentu.

4. *Normalization*

   Normalization adalah proses penormalan kata, dimana tidak setiap kata dapat terdeteksi oleh program maka akan dilakukan normalisasi. Seperti

kata yang tidak baku, kata asing, *typo* dan proses penormalan lainnya.

5. *Stopword Removal*

   Stopword removal adalah kata-kata umum yang sering muncul dalam teks namun cenderung tidak memiliki nilai informasi yang tinggi dalam analisis teks atau pemrosesan bahasa alami (purwanti, 2023). Contohnya adalah kata-kata seperti "di", "dan", "akan", "untuk" dan lain sebagainya. Daftar stopword removal umum biasanya disediakan oleh *library* atau *framework* pemrosesan teks dalam berbagai bahasa pemrograman, seperti python *NLTK (Natural Language Toolkit)* atau library *scikit-learn*. Namun, daftar stopword removal dapat disesuaikan atau diperluas sesuai dengan kebutuhan dan konteks analisis teks yang spesifik.

6. *Stemming*

   Stemming adalah salah satu proses dalam pemrosesan teks yang dilakukan untuk mengubah kata-kata menjadi bentuk dasar atau kata dasar (*root word*) dengan cara menghilangkan akhiran atau awalan tertentu (Amin & Alfa Razaq, 2018). Proses stemming tidak mempertimbangkan konteks atau

makna kata, tetapi hanya berfokus pada pola struktural kata. Algoritma stemming yang umum digunakan adalah algoritma *Porter stemming* atau algoritma *Snowball stemming*. Algoritma-algoritma ini menggunakan aturan-aturan berbasis aturan morfologi bahasa untuk melakukan pemangkasan akhiran atau awalan kata.

Contoh penerapan *stemming* adalah mengubah kata-kata seperti "makanan", "makananmu", "makanannya" menjadi bentuk dasar "makan". Dengan demikian, semua kata-kata tersebut dapat dianggap sama dan diperlakukan sebagai satu entitas dalam analisis teks atau pemrosesan bahasa alami.

Tujuan utama dari tokenisasi adalah untuk mempersiapkan teks mentah agar dapat diolah lebih lanjut dalam analisis teks atau pemrosesan bahasa alami. Dengan memecah teks menjadi token-token yang lebih kecil, peneliti dapat mengidentifikasi dan mengisolasi unit-unit penting dalam teks, seperti kata-kata atau frasa, yang akan menjadi dasar dalam analisis lebih lanjut.

J. TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF merupakan gabungan dari 2 proses yaitu Term Frequency (TF) dan Inverse Document Frequency (IDF). TF-

IDF digunakan untuk merubah teks menjadi vector namun tetap memperhatikan apakah kata tersebut cukup informatif atau tidak. Kata yang sering muncul memiliki nilai besar dan yang jarang muncul memiliki nilai yang kecil. Kata yang sering muncul disebut juga stopword removal, yang artinya kata tersebut dianggap kurang penting oleh model (Fatmawati, M., 2017).

TF adalah menghitung banyaknya kemunculan kata (t) pada sebuah dokumen (d). disebabkan panjang kata setiap dokumen berbeda-beda, maka nilai TF dibagi dengan panjang dokumen (jumlah seluruh kata pada dokumen).

$$tf_{t,d} = \frac{n_{t,d}}{total\ number\ of\ terms\ in\ document} \quad (2.1)$$

Dimana:

tf = kata/dokumen

n = frekuensi kemunculan pada kata/doc

total = jumlah seluruh kata di setiap dokumen

Jika TF yang menghitung banyaknya kata yang sering muncul, IDF akan menganggap kata yang sering muncul adalah kata yang kurang penting dan kata jarang muncul adalah kata yang penting. Rumusnya:

$$idf_d = \frac{Number\ of\ document}{Number\ of\ document\ with\ term\ t'} \quad (2.2)$$

Kemudian hitung masing-masing TF-IDF di setiap korpus.

$$(tf_{idf})_{t,d} = tf_{t,d}\ .\ idf_t \quad (2.3)$$

K. Naïve Bayes Classifier

Naïve Bayes Classifier (NBC) merupakan salah satu metode klasifikasi yang sangat popular. Naïve bayes merupakan metode klasifikasi yang sederhana namun memiliki nilai akurasi yang tinggi. Algoritma NBC merupakan algoritma yang digunakan untuk mencari nilai probabilitas tertinggi untuk mengklasifikasi data uji pada kategori yang paling tepat (Engelhart, M. D., & Moughamian, 1968).

Naïve bayes adalah pengklasifikasi probabilitas sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari kumpulan data yang diberikan. Algoritma menggunakan teorema Bayes dan mengasumsikan bahwa semua atribut independen atau tidak bergantung satu sama lain berdasarkan nilai variabel kelas (Yasar, A., & Saritas, M. M., 2019). Definisi lain *naïve bayes* adalah klasifikasi yang menggunakan probabilitas dan metode

statistik yang diusulkan oleh ilmuwan Inggris Thomas Bayes, yang memprediksi kemungkinan masa depan berdasarkan pengalaman masa lalu (Bustami, 2014). Adapun persamaan *naïve bayes* adalah:

$$P(H|X) = \frac{P(X|H) * P(H)}{P(X)} \tag{2.4}$$

Dimana:

X : Data dengan class yang belum diketahui

H : Hipotesis data X merupakan suatu class spesifik

P(H|X) : Probabilitas hipotesis H berdasar kondisi X (*posteriori probability*)

P(H) : Probabilitas hipótesis H (*prior probability*)

P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis H (*likelihood probability*)

P(X) : Probabilitas X (*evidence probability*)

Ide landasan dari aturan Bayes adalah hasil dari hipotesis (H) dapat diperkirakan berdasarkan beberapa *evidence* (X) yang diamati. Hal yang ada perlu diperhatikan juga dalam Bayes adalah sebuah probabilitas awal atau P(H) adalah probabilitas dari hipotesis yang belum ada bukti yang diamati. Sebuah probabilitas *posterior* H atau P(H|X) adalah

probabilitas dari hipotesis setelah ada bukti yang diamati (Yurniarti et al., 2020).

L. Evaluasi

Pengukuran evaluasi penting untuk dilakukan untuk melihat apakah metode klasifikasi yang digunakan sudah tepat. Cara untuk mengetahui evaluasi adalah dengan menghitung nilai akurasi, *recall, precision*, dan *f1* score (Grandini et al., 2020). Nilai-nilai tersebut nantinya akan dituangkan ke dalam *confusion matrix*.

Confusion matrix atau matriks kebingungan adalah alat visualisasi yang biasa digunakan untuk *supervised learning.* Setiap kolom pada matriks berisi kls prediksi dan setiap baris berisi kelas kejadian sebenarnya atau *actual* (Mahfudh A. A., & Mustofa, H., 2019). Tabel *confusion matrix* ada pada tabel 2.1.

Tabel 2. 1 Tabel confusion matrix 2x2

Kelas Aktual	Kelas Prediksi	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Dimana:

1. TP (true positif), adalah data yang diprediksi positif dan data sesungguhnya juga positif.
2. TN (true negatif), adalah data yang diprediksi negatif dan data sesungguhnya juga negatif.
3. FP (false Positif), adalah data yang diprediksi positif dan data sesungguhnya negatif.
4. FN (false negatif), adalah data yang diprediksi negatif dan data sesungguhnya positif.

Adapun rumus untuk perhitungannya adalah sebagai berikut:

$$Akurasi = \frac{TN + TP}{TN + TP + FN + FP} \quad (2.5)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.6)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.7)$$

$$f1\ score = 2x \frac{Recall\ .\ Precision}{Recall +\ Precision} x\ 100\% \quad (2.8)$$

confusion matrix adalah metode yang umum digunakan untuk menghitung akurasi, *recall, precision*, dan tingkat kesalahan (*error rate*). Yang mana, akurasi berarti mengevaluasi kemampuan sistem untuk menemukan kecocokan terbaik dan ditentukan oleh persentase dokumen

yang diambil dan benar-benar relevan dengan kueri. *Recall* berarti mengevaluasi kemampuan sistem untuk menemukan semua item yang relevan dalam sekumpulan dokumen dan didefinisikan sebagai persentase dokumen yang relevan dengan kueri. Akurasi adalah perbandingan kasus yang teridentifikasi dengan benar dengan jumlah total kasus, dan tingkat kesalahan adalah perbandingan kasus yang teridentifikasi secara salah dengan jumlah total kasus. Presisi adalah ukuran seberapa besar tingkat kebenaran antara informasi yang diminta dengan respon yang diberikan system (arini et al., 2020). Adapun *f1 score* adalah parameter tunggal yang digunakan untuk mengukur keberhasilan gabungan dari *recall* dan *precision* (Zidan, M., 2022).

M. Kajian Penelitian Terkait

Tabel 2. 2 Penelitian Terdahulu

No	Pustaka	Metode	Hasil
1	(Soer, D., & Sutrisno, S., 2022) "Analisis Sentimen terhadap Pemerintahan Ridwan Kamil Sebagai Gubernur Jawa Barat Menggunakan Naïve Bayes"	Naïve Bayes Classifier	Tingkat akurasi analisis sentimen sekitar 84,38%.
2	(Fatchan, M., & Sugeng H., 2021)	Naïve Bayes Classifier	Tingkat akurasi analisis

	"Analisa Terpilihnya Tri Rismaharini sebagai Menteri Sosial dengan Pendekatan Algoritma Naïve Bayes"		terpilihnya Tri Rismaharini sekitar 90,33%, dengan presisi 77,7% dan recall 99,9%.
3	(Widowati, T., & Sadikin M., 2020) "Analisis Sentimen Twitter terhadap Tokoh Publik dengan Algoritma Naïve Bayes dan Support Vector Machine"	Naïve Bayes Classifier dan Support Vector Machine	Analisis sentimen terhadap tokoh publik dengan tingkat akurasi sekitar 92%, dengan presisi 85% dan recall 88%.

Tiga penelitian terkait mengenai analisis sentimen menggunakan algoritma Naïve Bayes memberikan pemahaman yang mendalam tentang dinamika opini dan reaksi masyarakat terhadap isu-isu politik dan sosial yang relevan di Indonesia. Dalam penelitian pertama yang dilakukan oleh Soer dan Sutrisno, mereka mengarahkan fokus pada analisis sentimen terhadap kinerja Ridwan Kamil sebagai Gubernur Jawa Barat. Dengan tingkat akurasi sekitar 84,38%, mereka menggunakan algoritma Naïve Bayes untuk memeriksa opini masyarakat dari berbagai sumber data, termasuk mungkin media sosial, survei publik, dan ulasan

online. Penelitian ini memberikan gambaran yang kuat tentang bagaimana masyarakat menilai kinerja pemerintahan regional di Jawa Barat (Soer, D., & Sutrisno, S., 2022).

Di sisi lain, penelitian kedua yang dilakukan oleh Fatchan dan Sugeng mengeksplorasi respons masyarakat terhadap penunjukan Tri Rismaharini sebagai Menteri Sosial. Dengan tingkat akurasi sekitar 90,33%, analisis sentimen ini memberikan wawasan yang berharga tentang bagaimana masyarakat merespons keputusan politik tertentu. Mereka juga menggunakan algoritma Naïve Bayes untuk menganalisis respons publik terhadap keputusan politik tersebut, memberikan pemahaman mendalam tentang persepsi masyarakat terhadap aparat pemerintah (Fatchan, M., & Sugeng H., 2021).

Selain itu, penelitian ketiga oleh Widowati dan Sadikin menggali sentimen Twitter terhadap tokoh publik menggunakan algoritma Naïve Bayes dan Support Vector Machine. Dengan tingkat akurasi sekitar 92%, penelitian ini menyoroti peran penting media sosial dalam membentuk opini dan pandangan masyarakat terhadap tokoh politik dan publik. Hasil penelitian ini memberikan wawasan yang berharga tentang dinamika komunikasi publik dalam lingkungan digital.

Secara keseluruhan, ketiga penelitian tersebut memberikan kontribusi yang signifikan dalam memahami opini publik dan sentimen masyarakat dalam konteks politik dan sosial di Indonesia (Widowati, T., & Sadikin M., 2020).

BAB III

METODOLOGI PENELITIAN

A. Sumber Data

Penelitian ini merupakan penelitian kuantitatif yang menggunakan metode *Naïve Bayes Classifier* untuk mengukur tingkat akurasi dan hasil sentimen pengguna media sosial X terhadap tokoh politik Ganjar Pranowo dan Mohammad Mahfud MD dalam konteks pemilihan presiden 2024. Data yang digunakan terdiri dari data primer dan data sekunder. Data primer diperoleh dari cuitan di aplikasi media sosial X dengan rentang waktu antara 18 Oktober 2023 hingga 31 Januari 2024. Kata kunci yang digunakan adalah "ganjar pranowo mahfud until:2024-01-31 since:2023-10-18 lang:id". Sebanyak 5303 cuitan sebagai sampel yang dihasilkan dan 1000 cuitan diambil sebagai sampel dalam penelitian ini. Sementara itu, data sekunder diperoleh dari literatur jurnal yang relevan dan digunakan sebagai pendukung dalam penelitian ini. Dibandingkan dengan perhitungan manual oleh peneliti menggunakan rumus *naïve bayes classifier*.

B. Kebutuhan Perangkat Penelitian

Dalam penelitian ini diperlukan perangkat yang mendukung untuk kebutuhan penelitian, diantaranya perangkat keras dan perangkat lunak yang digunakan peneliti dengan spesifikasi seperti berikut:

1. Kebutuhan Perangkat Keras

Tabel 3. 1 Tabel Kebutuhan Perangkat keras

No	Perangkat Keras	Spesifikasi
1	Device	Asus A516JA-HD3121
2	Processor	Intel i3 gen 10th?
3	Memori (RAM)	1TB 256 SII
4	Monitor	15”
5	Keyboard dan Mouse	Normal

2. Kebutuhan Perangkat Lunak

Tabel 3. 2 Tabel Kebutuhan Perangkat Lunak

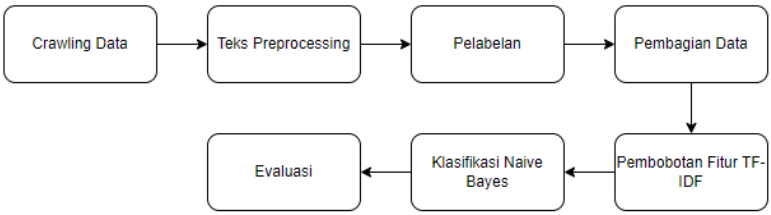
No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 11 64-bit
2	Bahasa Pemrograman	Python
3	Ms. Office	Ms. Word, Ms Excel 2019
4	Google Colab	Colab 2024-01-08
5	Browser	Chrome

C. Metode Penelitian

Penelitian ini merupakan jenis dari penelitian kuantitatif

yaitu adalah jenis penelitian yang menghasilkan penemuan-penemuan yang dapat dicapai (diperoleh) dengan menggunakan prosedur-prosedur statistik atau cara lain dari kuantifikasi(pengukuran).

Adapun langkah-langkah pada penelitian ini direpresentasikan melalui diagram flowchart pada Gambar 3.1. Flowchart berfungsi sebagai diagram untuk menggambarkan suatu proses urutan operasi secara visual dengan menampilkan sejumlah simbol (Yurniarti, 2019).



Gambar 3. 1 Desain Penelitian

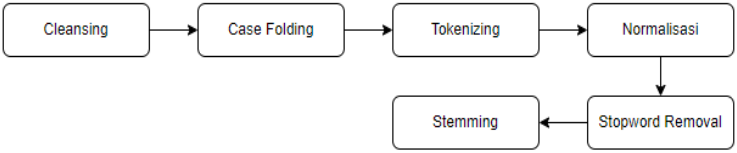
1. Pengambilan Data (*Crawling Data*)

   Pengumpulan data adalah Langkah pertama untuk melakukan analisis sentimen. Data dikumpulkan menggunakan teknik crawling dengan menggunakan google collab dan bantuan modul tweet-harvest. Dataset yang dikumpulkan merupakan tweet berbahasa Indonesia yang mengandung kata kunci yang berkaitan dengan

Ganjar Pranowo dan Mahfud MD sebagai pasangan pemilihan presiden 2024.

2. Text Preprocessing

Text preprocessing adalah tahap untuk mempersiapkan dataset dari teks mentah supaya siap untuk dianalisis teks sesuai dengan metode yang digunakan. Teks yang berasal dari media sosial X dapat mengandung noise, tanda baca, karakter khusus, kata-kata yang tidak relevan, dan berbagai variasi kata yang memberikan arti yang serupa.



Gambar 3. 2 Tahapan Text Preprocessing

Gambar menyajikan tahapan pada *text preprocessing*. Dalam penelitian ini, *text preprocessing* terdiri dari enam proses, yaitu *cleansing, case folding, tokenizing, normalisasi, stopword removal,* dan *stemming.* Penjelasan setiap proses pada tahapan *text preprocessing* sebagai berikut:

a) Cleansing

Cleansing adalah tahap membersihkan data dari

noise, *missing value*, *error*, data yang tidak penting dan data yang tidak konsisten. Pada penelitian ini mengevaluasi *column* data yang akan digunakan.

b) *Case Folding*

Case folding adalah tahapan yang dilakukan untuk menyamaratakan penggunaan huruf kapital pada *text*, dengan mengubah semua huruf menjadi kecil (*lowercase*). Contoh hasil dari proses Case Folding ada pada tabel 3.3.

Tabel 3. 3 Tabel Contoh Case Folding

Input Process	Output Process
Hari Sumpah Pemuda, Dico Ganinduto Ajak Generasi Muda Kuasai Teknologi #airlanggahartarto #kuningkeren #PrabowoGibran #partaigolkar #golkarpedia https://t.co/CXZRHXCQ7W	hari sumpah pemuda, dico ganinduto ajak generasi muda kuasai teknologi #airlanggahartarto #kuningkeren #prabowogibran #partaigolkar #golkarpedia https://t.co/cxzrhxcq7w
@Uki23 Betul….. itu hasil survei yg transparant diambil dari partai pendukung ganjar dan Mahfud	@uki23 betul….. itu hasil survei yg transparant diambil dari partai pendukung ganjar dan mahfud

c) *Tokenizing*

Tokenizing adalah tahap yang dilakukan untuk memecah kalimat-kalimat *fulltext* menjadi kata atau biasa disebut dengan token. Proses ini juga membersihkan teks dari *mentions* (@), *Uniform Resource Locator* (URL), *hashtag* (#), *special character* (=+_-()*&^%#@!$?.,<>;':"|\{}[]), emoji, angka, dan spasi (*white space)*. Contoh hasil proses Tokenizing pada tabel 3.5.

Tabel 3. 4 Tabel Contoh Tokenizing

Input Process	Output Process
hari sumpah pemuda, dico ganinduto ajak generasi muda kuasai teknologi #airlanggahartarto #kuningkeren #prabowogibran #partaigolkar #golkarpedia https://t.co/cxzrhxcq7w	[hari, sumpah, pemuda, dico, ganinduto, ajak, generasi, muda, kuasai, teknologi]
@uki23 betul….. itu hasil survei yg transparant diambil dari partai pendukung ganjar dan Mahfud	[betul, itu, hasil, survei, yg, transparant, diambil, dari, partai, pendukung, ganjar, dan, mahfud]

d) *Normalisasi*

Normalisasi adalah tahapan untuk menormalisasikan

data dan agresi data. Normalisasi data mengubah kata dalam teks menjadi bentuk baku bahasa Indonesia dengan memeriksa setiap token atau kata. Contoh hasil normalisasi pada Tabel 3.5.

Tabel 3. 5 Tabel Contoh Hasil Normalisasi

Input Process	Output Process
[keraguan, pdip, akan, kemenangan, capresnya, **jd**, diputaran, ke, ada, alasan, tepat, utk, dukung, ganjarmahfud]	[keraguan, pdip, akan, kemenangan, capresnya, **jadi**, diputaran, ke, ada, alasan, tepat, untuk, dukung, ganjarmahfud]
[selamat, pagi, hidup, dan, politik, yang, tanpa, caci, maki, bukan, dengan, **mrndhkn**, seseorang, **utk**, mencapai, suatu, kehormatan, sudaryono, mahfud]	[selamat, pagi, hidup, dan, politik, yang, tanpa, caci, maki, bukan, dengan, **merendahkan**, seseorang, **untuk**, mencapai, suatu, kehormatan, sudaryono, mahfud]

e) Stopword Removal

Stopword removal adalah tahap yang dilakukan untuk menghapus kata-kata yang tidak memiliki makna atau tidak penting pada data *text*. Proses stopword removal dilakukan dengan memeriksa setiap kata dalam teks, jika ada kata yang merupakan *stoplist*, maka kata tersebut akan dihapus. Contoh hasil stopword removal pada Tabel 3.6.

Tabel 3. 6 Tabel Contoh Hasil Stopword Removal

Input Process	Output Process
[keraguan, pdip, **akan**, kemenangan, capresnya, **jadi**, diputaran, **ke**, **ada**, alasan, **tepat**, **untuk**, dukung, ganjarmahfud]	[keraguan, pdip, kemenangan, capresnya, diputaran, alasan, dukung, ganjarmahfud]
[andai, ganjar, wafat, **saat**, presiden, **dan**, digantikan, mahfud, mencuat, gerindra, bereaksi]	[andai, ganjar, wafat, presiden, digantikan, mahfud, mencuat, gerindra, bereaksi]

f) Stemming

Stemming adalah tahap untuk mengubah kata-kata yang memiliki akar kata yang sama menjadi dasarnya. Contoh hasil stemming pada Tabel 3.7.

Tabel 3. 7 Tabel Contoh Hasil Stemming

Input Process	Output Process
[hasil, survei, transparant, **di**ambil, partai, **pen**dukung, ganjar, mahfud]	[hasil, survei, transparant, **ambil**, partai, **dukung**, ganjar, mahfud]
[**per**juangan, qodari, **pen**dukung, ganjar, mahfud, harga, mati, amplope, akeh, **meng**harap, nikmati, kue, **ke**kuasa**an**, **peng**amat, abal, abal]	[**juang**, qodari, **dukung**, ganjar, mahfud, harga, mati, amplope, akeh, **harap**, nikmat, kue, **kuasa**, **amat**, abal, abal]

3. Pelabelan

Setelah dilakukan *crawling,* langkah selanjutnya adalah pelabelan. Data yang telah disimpan dalam bentuk *csv*, akan dilakukan pelabelan. Contoh proses pelabelan ada pada tabel 3.8.

Tabel 3. 8 Tabel Contoh Hasil Labeling

Komentar	Sentimen
[hasil, survei, transparant, ambil, partai, dukung, ganjar, mahfud]	Positive
[tuju, narasi, khianat, hilang, anies, tuduh, khianat, dukung, ganjarmahfud, tuduh, khianat, dukung, pdip, maju, tuduh, khianat, smuamasak, maju, kasih, sempat, muda]	Negative

4. Pembagian data

Pembagian data bertujuan untuk memisahkan dataset menjadi data latih dan data uji. Data latih digunakan untuk melatih model, sedangkan data uji digunakan untuk mengukur kinerja model yang telah dilatih. Dalam penelitian ini, digunakan teknik k-fold cross-validation untuk membagi data. Teknik ini melibatkan pembagian

data menjadi k partisi yang sama besar, di mana satu partisi digunakan sebagai data uji dan partisi lainnya digunakan sebagai data latih. Proses pengujian dilakukan dalam k iterasi secara bergantian, di mana setiap partisi bergantian menjadi data latih dan data uji (Erfina, A. & Lestari, R. A., 2023).

Pembagian data adalah membagi data menjadi data *training* atau data latih dan data *testing* atau data uji dengan menggunakan *split validation*. *Split validation* dilakukan dengan jumlah data testing diambil 20% dari data *training*. Pengambilan data dilakukan secara random dengan batuan *library python* (Turmudi Zy, A., et al., 2021). *Split Data* dilakukan dengan membagi data *training* dan data *testing* dengan perbandingan 80;20. Pembagian data dengan perbandingan tersebut merupakan strategi yang paling sederhana dan paling umum digunakan (Joseph V. R., & Vakayil, A., 2022).

5. Pembobotan Fitur TF-IDF

   Pembobotan fitur dilakukan untuk mengubah fitur berupa kata menjadi data numerik. Pada langkah ini, setiap fitur yang diperoleh dari langkah text preprocessing diberi bobot dengan *Term Frequency Inverse Document*

Frequency (TF-IDF), yang menghasilkan bobot berdasarkan keseringan kemunculan kata tertentu dalam dokumen. Proses pembobotan fitur TF-IDF meliputi perhitungan nilai *Term Frequency* (TF) dan nilai *Inverse Document Frequency* (IDF) untuk setiap fitur di setiap dokumen (Marga, N., et al., 2021).

6. Klasifikasi Naïve Bayes Classifier

   Setelah proses pembobotan fitur TF-IDF, langkah berikutnya adalah melakukan klasifikasi menggunakan metode naïve bayes. Peneliti akan menggunakan metode naïve bayes untuk menilai keakuratan klasifikasi dengan membaginya ke dalam dua output, yaitu positif dan negatif. Setelah proses *preprocessing* dan ekstraksi fitur dengan TF-IDF, data akan diuji menggunakan data *training* dan data *testing* untuk mengevaluasi keakuratan klasifikasi dengan metode naïve bayes. Kelas yang memiliki skor terbanyak akan dianggap sebagai kelas yang tepat (Fitri, E., 2020).

7. Evaluasi

   Langkah berikutnya adalah melakukan evaluasi model untuk setiap model klasifikasi. Evaluasi model dilakukan dengan menguji kinerja metode melalui matriks konfusi

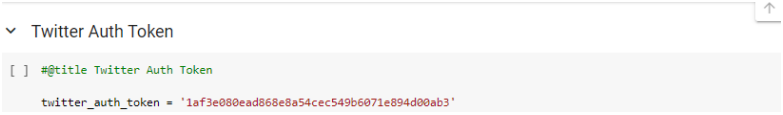
multi-layer. Matriks konfusi berisi informasi yang membandingkan hasil klasifikasi sistem dengan hasil klasifikasi yang seharusnya (Turmudi Zy et al., 2021). Dengan kata lain, data uji akan menghasilkan prediksi kelas, yang kemudian dibandingkan dengan kelas sebenarnya dari data pengujian yang sebelumnya disembunyikan (Zidan, M., 2022). Dengan demikian, performa model naïve bayes dalam mengklasifikasi sentimen akan dievaluasi berdasarkan tingkat akurasi, *presisi, recall, dan f1 score*. Hasil evaluasi performa klasifikasi model naïve bayes classifier dalam mengklasifikasi sentimen dan perhitungan manual oleh peneliti menggunakan rumus dari klasifikasi naïve bayes classifier untuk mengetahui apakah model naïve bayes classifier memiliki performa yang baik dalam penelitian ini

BAB IV

HASIL DAN PEMBAHASAN

A. Pengambilan Data Media Sosial X

Pengambilan data atau proses *crawling data* twitter ini dilakukan menggunakan modul Tweet-Harvest yang dapat digunakan pada *python*. Modul *Tweet-Harvest* digunakan untuk mempermudah melakukan pengambilan data melalui media sosial X. Tujuannya untuk mengambil data-data twitter yang ingin dicari. Modul *Tweet-Harvest* ini sangat efektif karena hanya diperlukan *Auth Token* akun media sosial X dan kata kunci pencarian yang ingin dicari pada akun media sosial X. Maka untuk menggunakan modul *Tweet-Harvest* peneliti menginisiasi dahulu *Auth Token* akun media sosial X pada google colab. Berikut cara inisiasi Auth Token akun media sosial X seperti gambar 4.1 di bawah.



Gambar 4. 1 Auth Token

Selain membutuhkan inisiasi *Auth Token* akun media sosial X membutuhkan beberapa python *packages* untuk melakukan proses *crawling* data maka harus meng-*import* data

yang diperlukan seperti gambar 4.2.

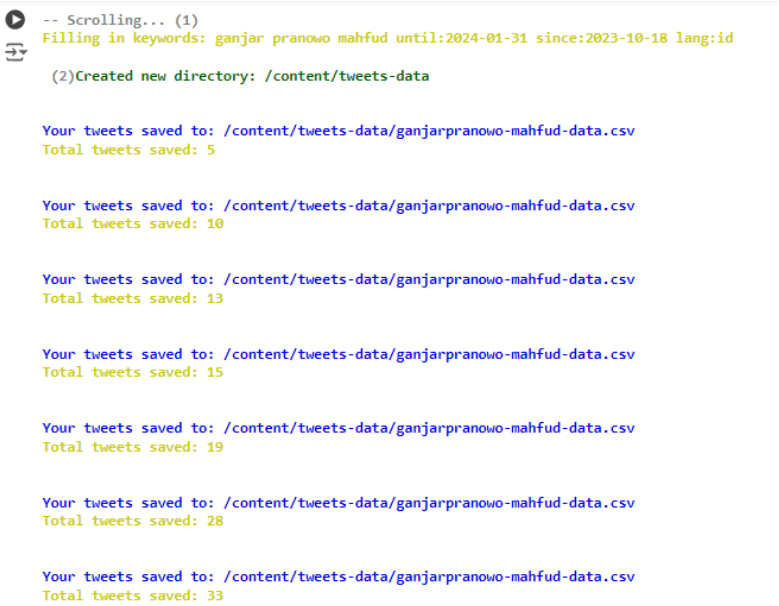
Gambar 4. 2 Import Reqired Python Package

Pada Gambar 4.2 terdapat proses instalasi pandas yang merupakan salah satu paket analisis data yang paling populer di lingkungan python. Selain itu terdapat proses instalasi Node JS versi 20 yang dibutuhkan untuk mendukung modul *Tweet-Harvest* yang digunakan pada crawling data media sosial X. Setelah proses instalasi packages yang diperlukan, selanjutnya memasukkan kode proses *crawling* data. Berikut pada gambar 4.3 *source code* penerapan proses *crawling* data.

Gambar 4. 3 Source Code Crawl Data

Pada Gambar 4.3 terdapat proses *crawling* data media sosial X menggunakan modul *Tweet-Harvest*. Pertama, sebuah

variabel `*filename*` diinisialisasi dengan nilai 'ganjarpranowo-mahfud-data.csv'. Ini akan digunakan untuk memberi nama file yang akan menyimpan data yang di-*crawl*. Kemudian, sebuah *string* `*search_keyword*` didefinisikan untuk menentukan kata kunci pencarian yang akan digunakan dalam proses *crawling*. Pencarian ini dibatasi oleh tanggal, yaitu mulai dari '2023-10-18' hingga '2024-01-31', dan hanya mencakup bahasa Indonesia ('lang:id'). Ini memastikan bahwa data yang di-*crawl* sesuai dengan kriteria tertentu. Selanjutnya, variabel `limit` diinisialisasi dengan nilai 5303, yang menentukan batas jumlah tweet yang akan di-*crawl*. Terakhir, perintah `npx` digunakan untuk menjalankan Tweet-Harvest dengan parameter yang sesuai, seperti `-o` untuk menentukan nama file output, `-s` untuk kata kunci pencarian, `-l` untuk batas jumlah tweet, dan `--token` untuk token otorisasi media sosial X yang tidak tersedia dalam potongan kode yang diberikan.



Gambar 4. 4 Hasil Proses Crawl Data

Pada Gambar 4.4 hasil proses crawl data ini berhasil mengumpulkan tweet sesuai dengan kata kunci dan rentang waktu yang ditentukan, menghasilkan total 5303 tweet yang disimpan dalam file CSV untuk analisis.

B. Teks Preprocessing

Tahapan ini terdiri dari beberapa proses karena data cuitan memiliki karakteristik yang tidak terstruktur yang sangat memuat *noise*. Maka, pada tahapan ini bertujuan untuk mengubah suatu data yang masih mentah/kotor lalu diolah

menjadi data bersih sehingga dapat dilakukan pengklasifikasian. Penerapan tahap preprocessing data pada penelitian ini dilakukan dengan melakukan 6 proses secara urut, di antaranya:

1. *Cleansing*

Cleansing dengan Lexicon Inset proses cleansing bertujuan untuk membersihkan data teks dari karakter dan elemen yang tidak relevan, sehingga teks yang dihasilkan lebih konsisten dan siap untuk analisis lebih lanjut. Langkah-langkah cleansing yang diterapkan pada data **'df'** mencakup:

1. Menghapus Kolom Tidak Relevan: Menghapus beberapa kolom yang tidak diperlukan untuk analisis teks.
2. Menghapus Kata "RT": Menghapus kata "RT" yang sering muncul dalam retweet.
3. Menghapus Karakter Khusus: Menghapus tab, baris baru, backslash, dan karakter non-ASCII, serta mention, tautan, dan hashtag.
4. Menghapus Angka: Menghapus semua angka dari teks.
5. Menghapus Tanda Baca: Menghapus semua tanda baca.

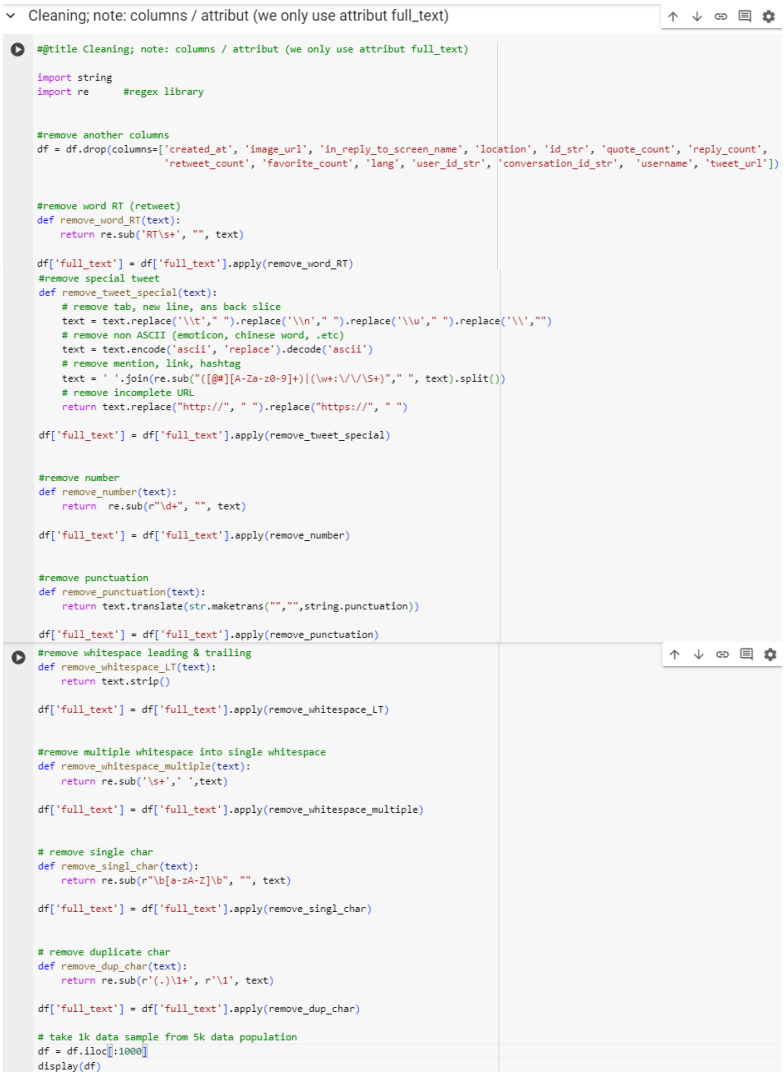
6. Menghapus Spasi Awal dan Akhir: Menghapus spasi di awal dan akhir teks.

7. Menghapus Spasi Ganda: Mengganti spasi ganda dengan spasi tunggal.

8. Menghapus Karakter Tunggal: Menghapus karakter tunggal yang terpisah.

9. Menghapus Karakter Duplikat: Mengganti karakter berulang dengan satu karakter.

Setelah proses cleansing selesai, data yang telah dibersihkan ditampilkan untuk memastikan proses berjalan dengan benar. Proses ini memastikan bahwa teks lebih bersih dan seragam untuk analisis lebih lanjut. Berikut *source code* yang dari tahap cleansing menggunakan lexicon inset ditunjukkan pada gambar 4.5.



Gambar 4. 5 Cleansing Menggunakan Lexicon Inset

Cleansing dengan Ahli Bahasa proses cleansing ini serupa

dengan proses cleansing lexicon inset, namun diterapkan pada DataFrame **'df_hasil_labeling'.** Data yang telah di-label manual dibersihkan dengan langkah-langkah yang sama seperti di atas, memastikan teks yang dihasilkan lebih konsisten dan siap untuk analisis lebih lanjut. Berikut *source code* yang dari tahap cleansing menggunakan ahli bahasa ditunjukkan pada gambar 4.6.

Gambar 4. 6 Cleasning Menggunakan Ahli Bahasa

Hasil dari penerapan hasil dari *cleansing* yang ditunjukkan pada tabel 4.1.

Tabel 4. 1 Hasil Penerapan Cleansing

Cuitan Tweet	Output Process
Ganjar Mahfud menyoroti ketergantungan teknologi baterai membuktikan pemahaman mendalam akan kebutuhan masa depan . Ganjar Pranowo - Hanya dia yang betul merakyat dan akan berbuat lebih demi rakyat @biabiwbiw #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/9Xza2Da5X5	Ganjar Mahfud menyoroti ketergantungan teknologi baterai membuktikan pemahaman mendalam akan kebutuhan masa depan Ganjar Pranowo hanya dia yang betul merakyat dan akan berbuat lebih demi rakyat

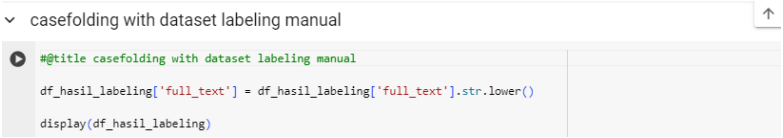
2. Case Folding

Case Folding dengan Lexicon Inset proses case folding bertujuan untuk mengubah semua huruf dalam teks menjadi huruf kecil untuk konsistensi. Dengan menggunakan pustaka Pandas, metode **'str.lower()'** diterapkan pada kolom 'full_text' dalam DataFrame **'df'.** Hal ini mengubah semua teks menjadi huruf kecil, membantu dalam penanganan teks yang lebih seragam dan mengurangi variasi kata yang sama namun berbeda dalam huruf besar/kecil. Hasil case folding

ditampilkan untuk memastikan proses berjalan dengan benar. Berikut *source code* yang dari tahap case folding menggunakan lexicon inset ditunjukkan pada gambar 4.7.

Gambar 4. 7 Case Folding Menggunakan Lexicon Inset

Case Folding dengan Ahli Bahasa proses ini serupa dengan proses lexicon inset, di mana metode **'str.lower()'** diterapkan pada kolom 'full_text' dalam DataFrame **'df_hasil_labeling'.** Semua teks diubah menjadi huruf kecil, memastikan konsistensi dalam analisis teks selanjutnya. DataFrame hasil case folding ditampilkan untuk memastikan proses berjalan dengan benar. Berikut *source code* yang dari tahap case folding menggunakan lexicon inset ditunjukkan pada gambar 4.8.



Gambar 4. 8 Tokenizing Menggunakan Ahli Bahasa

Hasil dari penerapan hasil dari *case folding* yang ditunjukkan pada tabel 4.2.

Tabel 4. 2 Hasil Penerapan Case Folding

Input Process	Output Process
Ganjar Mahfud menyoroti ketergantungan teknologi baterai membuktikan pemahaman mendalam akan kebutuhan masa depan **Ganjar Pranowo** hanya dia yang betul merakyat dan akan berbuat lebih demi rakyat	ganjar mahfud menyoroti ketergantungan teknologi baterai membuktikan pemahaman mendalam akan kebutuhan masa depan ganjar pranowo hanya dia yang betul merakyat dan akan berbuat lebih demi rakyat

3. Tokenizing

Tokenizing dengan Lexicon Inset proses tokenizing menggunakan lexicon inset melibatkan penggunaan pustaka NLTK untuk memecah teks menjadi token atau kata-kata individual. Pertama, pustaka NLTK diimpor dan modul **'punkt'** diunduh untuk mendukung tokenisasi. Fungsi **'word_tokenize_wrapper'** dibuat untuk memanggil **'word_tokenize'** dari NLTK, yang digunakan untuk memecah teks menjadi token. Fungsi ini kemudian diterapkan pada kolom 'full_text' dalam DataFrame **'df',** menghasilkan kolom baru 'tweet_tokens' yang berisi daftar kata-kata dari setiap teks. Hasil tokenisasi ditampilkan untuk memastikan proses

berjalan dengan benar. Berikut *source code* yang dari tahap tokenizing menggunakan lexicon inset ditunjukkan pada gambar 4.9.

Gambar 4. 9 Tokenizing Menggunakan Lexicon Inset

Tokenizing dengan Ahli Bahasa proses ini menggunakan fungsi **'word_tokenize_wrapper'** yang sama untuk memecah teks menjadi token. Fungsi ini diterapkan pada kolom 'full_text' dalam DataFrame **'df_hasil_labeling',** menghasilkan kolom baru 'tweet_tokens' yang berisi daftar kata-kata dari setiap teks. DataFrame hasil tokenisasi ditampilkan untuk memastikan proses berjalan dengan benar. Berikut *source code* yang dari tahap tokenizing menggunakan ahli bahasa ditunjukkan pada gambar 4.10.

Gambar 4. 10 Tokenizing Menggunakan Ahli Bahasa

Sehingga, hasil dari tahapan tokenization dapat ditunjukkan pada tabel 4.3.

Tabel 4. 3 Hasil Penerapan Tokenizing

Input Process	Output Process
ganjar mahfud menyoroti ketergantungan teknologi baterai membuktikan pemahaman mendalam akan kebutuhan masa depan ganjar pranowo hanya dia yang betul merakyat dan akan berbuat lebih demi rakyat	['ganjar', 'mahfud', 'menyoroti', 'ketergantungan', 'teknologi', 'baterai', 'membuktikan', 'pemahaman', 'mendalam', 'akan', 'kebutuhan', 'masa', 'depan', 'ganjar', 'pranowo', 'hanya', 'dia', 'yang', 'betul', 'merakyat', 'dan', 'akan', 'berbuat', 'lebih', 'demi', 'rakyat']

4. Normalisasi

Normalisasi dengan Lexicon Inset proses normalisasi menggunakan lexicon inset dimulai dengan membaca file CSV yang berisi kata-kata dan padanannya yang sudah dinormalisasi. File ini dimuat ke dalam sebuah DataFrame

'normalized_word' menggunakan **'pandas'.** Sebuah kamus **'normalized_word_dict'** kemudian dibangun dari DataFrame ini, di mana setiap kata yang perlu dinormalisasi menjadi kunci dan padanannya menjadi nilai.

Fungsi **'normalized_term'** dibuat untuk menggantikan setiap kata dalam dokumen dengan padanannya dari kamus jika ada, atau membiarkan kata tersebut tidak berubah jika tidak ada dalam kamus. Fungsi ini diterapkan pada kolom 'tweet_tokens' dalam DataFrame **'df',** menghasilkan kolom baru 'normalisasi_tweet'. Hasil normalisasi ditampilkan untuk memastikan proses berjalan dengan benar. Berikut *source code* yang dari tahap normalisasi menggunakan lexicon inset ditunjukkan pada gambar 4.11.

Normalisasi attribut tweet_tokens

```
#@title Normalisasi attribut tweet_tokens

# Specify the path to your CSV file
file_path_norm = f"normalisasi.csv"

# Read the CSV file into a pandas DataFrame
normalized_word = pd.read_csv(file_path_norm, encoding='latin1')
pd.set_option('display.max_colwidth', None)

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_tweet'] = df['tweet_tokens'].apply(normalized_term)

print('Normalize Result : \n')
display(df)
print('\n\n\n')
```

Gambar 4. 11 Normalisasi Menggunakan Lexicon Inset

Normalisasi dengan Ahli Bahasa proses ini menggunakan fungsi **'normalized_term'** yang sama untuk menggantikan setiap kata dalam dokumen dengan padanannya dari kamus jika ada. Fungsi ini diterapkan pada kolom 'tweet_tokens' dalam DataFrame **'df_hasil_labeling',** menghasilkan kolom baru 'normalisasi_tweet'. DataFrame hasil normalisasi ditampilkan untuk memastikan proses berjalan dengan benar. Berikut *source code* yang dari tahap normalisasi menggunakan ahli bahasa ditunjukkan pada gambar 4.12.

Gambar 4. 12 Normalisasi Menggunakan Ahli Bahasa

Sehingga, hasil dari tahapan normalisasi dapat ditunjukkan pada tabel 4.4.

Tabel 4. 4 Hasil Proses Normalisasi

Input Process	Output Process
['ganjar', 'mahfud', 'menyoroti', 'ketergantungan', 'teknologi', 'baterai', 'membuktikan', 'pemahaman', 'mendalam', 'akan', 'kebutuhan', 'masa', 'depan', 'ganjar', 'pranowo', 'hanya', 'dia', 'yang', 'betul', 'merakyat', 'dan', 'akan', 'berbuat', 'lebih', 'demi', 'rakyat']	['ganjar', 'mahfud', 'menyoroti', 'ketergantungan', 'teknologi', 'baterai', 'membuktikan', 'pemahaman', 'mendalam', 'akan', 'kebutuhan', 'masa', 'depan', 'ganjar', 'pranowo', 'hanya', 'dia', 'yang', 'betul', 'merakyat', 'dan', 'akan', 'berbuat', 'lebih', 'demi', 'rakyat']

5. Stopword Removal

Stopword Removal dengan Lexicon Inset proses ini mengimpor stopwords dari NLTK dan menambahkan beberapa stopwords tambahan dari file eksternal. Daftar stopwords digabungkan dan diubah menjadi set untuk efisiensi pencarian kata. Fungsi stopwords_removal dibuat untuk menghapus kata-kata yang ada dalam daftar stopwords dari dokumen. Fungsi ini diterapkan pada kolom 'normalisasi_tweet' dalam dataframe df, menghasilkan kolom baru 'stopword_tweet'. Hasil penghapusan stopwords ditampilkan untuk memastikan proses berjalan dengan benar. Berikut *source code* yang dari tahap stopword removall menggunakan lexicon inset ditunjukkan pada gambar 4.13.

Gambar 4. 13 Stopword Removall Menggunakan Lexicon Inset

Stopword Removal dengan Ahli Bahasa proses ini menggunakan fungsi **'stopwords_removal'** untuk menghapus stopwords dari teks. Fungsi ini diterapkan pada kolom 'normalisasi_tweet' dalam dataframe **'df_hasil_labeling',** menghasilkan kolom baru 'stopword_tweet'. Dataframe hasil penghapusan stopwords ditampilkan untuk memastikan proses berjalan dengan benar. Berikut *source code* yang dari tahap stopword removall menggunakan ahli bahasa ditunjukkan pada gambar 4.14.

Gambar 4. 14 Stopword Removall Menggunakan Ahli Bahasa

Sehingga, hasil dari tahapan stopword removal dapat ditunjukkan pada tabel 4.5.

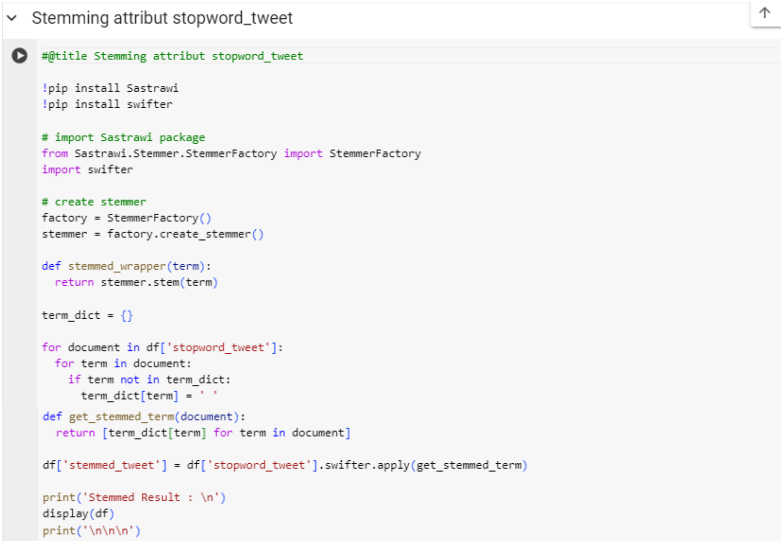
Tabel 4. 5 Hasil Proses Stopword Removal

Input Process	Output Process
['ganjar', 'mahfud', 'menyoroti', 'ketergantungan', 'teknologi', 'baterai', 'membuktikan', 'pemahaman', 'mendalam', '**akan**', 'kebutuhan', '**masa**', '**depan**', 'ganjar', 'pranowo', '**hanya**', '**dia**', '**yang**', '**betul**', 'merakyat', '**dan**', '**akan**', 'berbuat', '**lebih**', '**demi**', 'rakyat']	['ganjar', 'mahfud', 'menyoroti', 'ketergantungan', 'teknologi', 'baterai', 'membuktikan', 'pemahaman', 'mendalam', 'kebutuhan', 'ganjar', 'pranowo', 'merakyat', 'berbuat', 'rakyat']

6. Stemming

Proses stemming menggunakan lexicon inset bertujuan untuk melakukan stemming pada teks dalam atribut 'stopword_tweet' menggunakan paket Sastrawi. Proses ini dimulai dengan menginstal paket **'Sastrawi'** dan **'swifter',** kemudian membuat stemmer menggunakan **'StemmerFactory'** dari Sastrawi. Fungsi **'stemmed_wrapper'** dibuat untuk membungkus proses stemming dari Sastrawi.

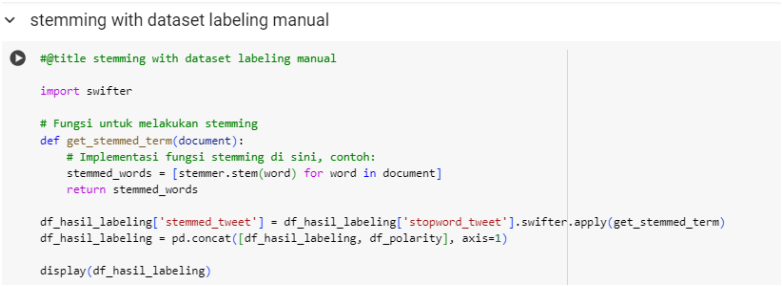
Sebuah kamus **'term_dict'** dibangun untuk menyimpan hasil stemming dari setiap kata unik dalam kolom 'stopword_tweet'. Setiap kata dalam kamus diproses melalui fungsi **'stemmed_wrapper'** dan hasilnya disimpan kembali dalam kamus. Selanjutnya, fungsi **'get_stemmed_term'** digunakan untuk mengganti setiap kata dalam dokumen dengan hasil stemming dari kamus, dan diterapkan pada kolom 'stopword_tweet' menggunakan **'swifter.apply'** untuk mempercepat proses. Hasil stemming kemudian ditampilkan untuk memastikan proses telah berjalan dengan benar. Seperti pada tabel 4.6 dapat dilihat kata sebelumnya perjuangan dan setelah proses stemming kata berjuang menjadi juang, kata imbuhannya dibuang menjadi kata dasar. Berikut *source code* yang dari tahap stemming menggunakan lexicon inset ditunjukkan pada gambar 4.15.



Gambar 4. 15 Stemming Menggunakan Lexicon Inset

Source code menggunakan ahli bahasa juga melakukan stemming pada teks, tetapi dengan konvensi penamaan berbeda dan implementasi yang lebih ringkas. Fungsi **'get_stemmed_term'** digunakan untuk melakukan stemming pada setiap kata dalam dokumen menggunakan stemmer yang dibuat sebelumnya. Fungsi ini diterapkan pada kolom 'stopword_tweet' dalam dataframe **'df_hasil_labeling'** menggunakan **'swifter.apply'.** Hasil stemming kemudian digabungkan dengan dataframe **'df_polarity'** menggunakan **'pd.concat'** dan hasil akhir ditampilkan. Berikut *source code* yang dari tahap stemming menggunakan ahli bahasa

ditunjukkan pada gambar 4.16.



Gambar 4. 16 Stemming Menggunakan Ahli Bahasa

Kedua *source code* di atas menunjukkan proses stemming yang efektif menggunakan paket Sastrawi, dengan penerapan yang dioptimalkan melalui penggunaan 'swifter'. *Source code* menggunakan lexicon inset lebih detail dengan penggunaan kamus untuk menyimpan hasil stemming sementara *source code* menggunakan ahli bahasa lebih ringkas dengan langsung menerapkan fungsi stemming pada dataframe. Meskipun pendekatannya berbeda, kedua *source code* berhasil melakukan stemming untuk persiapan analisis teks lebih lanjut. Sehingga, hasil dari tahapan stemming dapat ditunjukkan pada tabel 4.6.

Tabel 4. 6 Hasil Proses Stemming

Input Process	Output Process
['ganjar', 'mahfud', '**menyoroti**', '**ketergantungan**', 'teknologi', 'baterai', '**membuktikan**', '**pemahaman**', '**mendalam**', '**kebutuhan**', 'ganjar', 'pranowo', '**merakyat**', '**berbuat**', 'rakyat']	['ganjar', 'mahfud', '**sorot**', '**gantung**', 'teknologi', 'baterai', '**bukti**', '**paham**', '**dalam**', '**butuh**', 'ganjar', 'pranowo', 'rakyat', '**buat**', 'rakyat']

C. Pelabelan

Pelabelan dengan Lexicon Inset proses labelling menggunakan lexicon inset melibatkan analisis sentimen teks berdasarkan dua kamus: **'positive.tsv'** dan **'negative.tsv'.** Kamus ini memuat daftar kata dengan skor bobot yang menunjukkan polaritasnya.

1. Membaca Kamus: Kamus kata positif dan negatif dibaca dari file TSV dan dimuat ke dalam kamus Python (**'lexicon_positive'** dan **'lexicon_negative'),** di mana setiap kata dipetakan ke bobotnya.

2. Fungsi Analisis Sentimen: Fungsi **'sentiment_analysis'** menghitung skor sentimen untuk setiap teks dengan menambahkan bobot kata

positif dan mengurangi bobot kata negatif. Skor ini kemudian digunakan untuk menentukan polaritas sentimen sebagai 'positive', 'negative', atau 'neutral'.

3. Penerapan Analisis: Fungsi **'sentiment_analysis'** diterapkan pada kolom **'stemmed_tweet'** dari DataFrame **'df',** menghasilkan kolom baru polarity_score dan polarity_inset.

4. Penghapusan Polaritas Netral: Baris dengan polaritas 'neutral' dihapus, dan hasil labelling disimpan dalam file CSV.

Berikut merupakan *source code* untuk melakukan proses pelabelan menggunakan Lexicon Inset yang ditunjukkan pada gambar 4.17.

Labelling

```
#@title Labelling
lexicon_positive = {}
# Membaca file positive.tsv
kamus_positive = f'positive.tsv'  # Ganti dengan path file positive.tsv di komputer Anda
positive = pd.read_csv(kamus_positive, delimiter='\t')
for index, row in positive.iterrows():
    word = row['word']
    weight = row['weight']
    lexicon_positive[word] = int(weight)

lexicon_negative = {}
# Membaca file negative.tsv
kamus_negative = f'negative.tsv'  # Ganti dengan path file negative.tsv di komputer Anda
negative = pd.read_csv(kamus_negative, delimiter='\t')
for index, row in negative.iterrows():
    word = row['word']
    weight = row['weight']
    lexicon_negative[word] = int(weight)
```

```
def sentiment_analysis(text):
    score = 0
    for word in text:
        if word in lexicon_positive:
            score += lexicon_positive[word]
        if word in lexicon_negative:
            score += lexicon_negative[word]
    polarity = 'neutral'
    if score > 0:
        polarity = 'positive'
    elif score < 0:
        polarity = 'negative'
    return score, polarity

results = df['stemmed_tweet'].apply(sentiment_analysis)
results = list(zip(*results))
df['polarity_score'] = results[0]
df['polarity_inset'] = results[1]
print(df['polarity_inset'].value_counts())
df = df[df['polarity_inset'] != 'neutral']

df.to_csv(r'hasil_labelling_InSet.csv', index=False, header=True)
display(df)
```

Gambar 4. 17 Pelabelan Menggunakan Lexicon Inset

Pelabelan dengan Ahli Bahasa Labelling dengan dataset labeling manual dilakukan dengan cara yang serupa, tetapi data berasal dari DataFrame **'df_hasil_labeling'** yang sudah memiliki label. Proses ini memastikan bahwa data yang sudah diberi label oleh ahli bahasa tetap digunakan dan diperiksa untuk polaritas yang sesuai.

1. Pengecekan Polaritas: Proses pengecekan polaritas diulang untuk memastikan bahwa data dalam

'df_hasil_labeling 'sesuai dengan polaritas yang diharapkan.

2. Penghapusan Polaritas Netral: Baris dengan polaritas 'neutral' dihapus untuk analisis lebih lanjut.

Berikut merupakan *source code* untuk melakukan proses pelabelan menggunakan ahli bahasay ang ditunjukkan pada gambar 4.18.

Gambar 4. 18 Pelabelan Menggunakan Ahli Bahasa

D. Pembagian Data

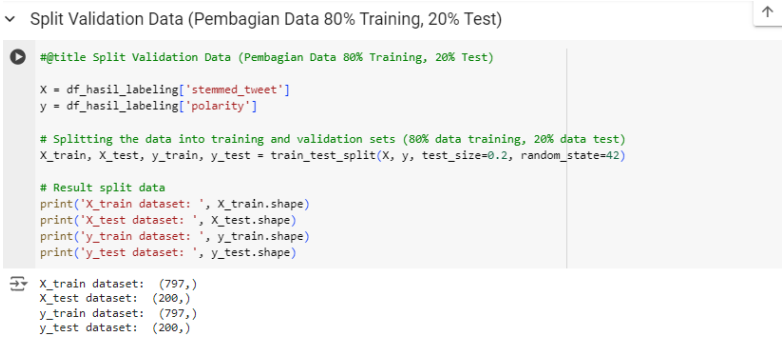
Proses pembagian data menjadi data latih dan uji menggunakan metode yang sama, tetapi dengan konvensi penamaan yang berbeda. Pembagian dilakukan dengan memisahkan 80% data untuk pelatihan dan 20% untuk pengujian menggunakan fungsi **'train_test_split'** dari **'sklearn.model_selection'.**

Pada *source code* menggunakan lexicon inset, data yang digunakan berasal dari kolom 'stemmed_tweet' dan 'polarity_inset' dari dataframe df. Data ini kemudian

dipisahkan menjadi **'X_train_inset', 'X_test_inset', 'y_train_inset',** dan **'y_test_inset'.** Hasil pembagian data dicetak untuk memastikan ukuran dataset yang dihasilkan. Berikut merupakan *source code* untuk melakukan proses pembagian data menggunakan Lexicon Inset yang ditunjukkan pada gambar 4.19.

Gambar 4. 19 Pembagian Data Menggunakan Lexicon Inset

Pada *source code* menggunakan ahli bahasa, data yang digunakan berasal dari kolom ‘stemmed_tweet’ dan ‘polarity’ dari dataframe ‘df_hasil_labeling’. Data ini kemudian dipisahkan menjadi ‘X_train’, ‘X_test’, ‘y_train’, dan ‘y_test’. Hasil pembagian data juga dicetak untuk memastikan ukuran datasaet yang dihasilkan. Berikut merupakan *source code* untuk melakukan proses pembagian data menggunakan ahli bahasa yang ditunjukkan pada gambar 4.20.



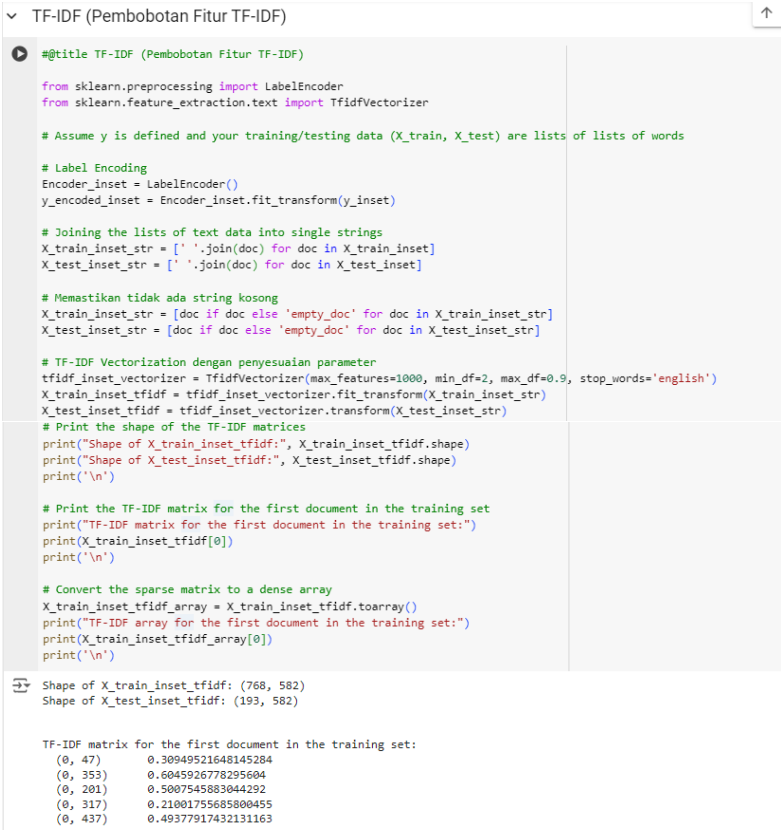
Gambar 4. 20 Pembagian Data Menggunakan Ahli Bahasa

Meskipun kedua *source code* memiliki fungsionalitas yang sama, yaitu membagi data menjadi data latih dan uji dengan proporsi yang sama, perbedaan utama terletak pada penamaan variabel dan dataframe yang digunakan. Hal ini menunjukkan bahwa metode pembagian data dapat diterapkan secara fleksibel dengan berbagai konvensi penamaan tanpa mengubah hasil akhir.

E. Pembobotan Fitur TF-IDF

Pembobotan fitur menggunakan TF-IDF (Term Frequency-Inverse Document Frequency) untuk mempersiapkan data teks sebelum digunakan dalam model pembelajaran mesin. Proses dimulai dengan pengkodean label

teks menjadi angka menggunakan **'LabelEncoder',** yang pada *source code* menggunakan lexicon inset variabelnya adalah **'Encoder_inset'** dan **'y_encoded_inset'**, sedangkan pada *source code* dengan ahli bahasa menggunakan **'Encoder'** dan **'y_encoded**'. Setelah itu, teks dalam setiap dokumen digabungkan menjadi satu string tunggal, memastikan tidak ada string kosong dengan menggantinya menjadi 'empty_doc'. Selanjutnya, vektorisasi TF-IDF dilakukan dengan parameter yang sama menggunakan **'TfidfVectorizer',** dan hasilnya disimpan dalam **'X_train_inset_tfidf'** dan **'X_test_inset_tfidf'** untuk *source code* dengan lexicon inset, serta **'X_train_tfidf'** dan **'X_test_tfidf'** untuk *source code* dengan ahli bahasa. Meskipun fungsi dan logika dari kedua *source code* identik, perbedaannya terletak pada penamaan variabel yang digunakan. Berikut merupakan *source code* untuk melakukan proses pembobotan fitur tf-idf menggunakan Lexicon Inset yang ditunjukkan pada gambar 4.21.



Gambar 4. 21 Pembobotan tf-idf Menggunakan Lexicon Inset

Berikut merupakan *source code* untuk melakukan proses pembobotan tf-idf menggunakan Ahli Bahasa yang ditunjukkan pada gambar 4.22.

TF-IDF (Pembobotan Fitur TF-IDF)

```
#@title TF-IDF (Pembobotan Fitur TF-IDF)

from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import TfidfVectorizer

# Assume y is defined and your training/testing data (X_train, X_test) are lists of lists of words

# Label Encoding
Encoder = LabelEncoder()
y_encoded = Encoder.fit_transform(y)

# Joining the lists of text data into single strings
X_train_str = [' '.join(doc) for doc in X_train]
X_test_str = [' '.join(doc) for doc in X_test]

# Memastikan tidak ada string kosong
X_train_str = [doc if doc else 'empty_doc' for doc in X_train_str]
X_test_str = [doc if doc else 'empty_doc' for doc in X_test_str]

# TF-IDF Vectorization dengan penyesuaian parameter
tfidf_vectorizer = TfidfVectorizer(max_features=1000, min_df=2, max_df=0.9, stop_words='english')
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train_str)
X_test_tfidf = tfidf_vectorizer.transform(X_test_str)
# Print the shape of the TF-IDF matrices
print("Shape of X_train_tfidf:", X_train_tfidf.shape)
print("Shape of X_test_tfidf:", X_test_tfidf.shape)
print('\n')

# Print the TF-IDF matrix for the first document in the training set
print("TF-IDF matrix for the first document in the training set:")
print(X_train_tfidf[0])
print('\n')

# Convert the sparse matrix to a dense array
X_train_tfidf_array = X_train_tfidf.toarray()
print("TF-IDF array for the first document in the training set:")
print(X_train_tfidf_array[0])
print('\n')
```

```
Shape of X_train_tfidf: (797, 583)
Shape of X_test_tfidf: (200, 583)


TF-IDF matrix for the first document in the training set:
  (0, 467)      0.2246106118610412 3
  (0, 82)       0.16490567838427347
  (0, 450)      0.19545019616603185
  (0, 420)      0.3267668838919183
  (0, 274)      0.44563491683499473
  (0, 502)      0.34028280306310904
  (0, 3)        0.34457095461813936
```

Gambar 4. 22 Pembobotan fitur tf-idf menggunakan Ahli Bahasa

Dalam penelitian ini, untuk contoh cara kerja perhitungan TF-IDF peneliti mengambil 3 contoh data yakni sebagai berikut:

(Doc 1) = “Semangat positif ibu atikoh menyebar ke seluruh lapisan masyarakat membangkitkan harapan akan masa depan

yang lebih cerah bersama ganjarmahfud ganjar pranowo terbukti lebih baik”

(Doc 2) = “Rakyat optimis bahwa ganjar mahfud akan membawa indonesia menjadi negara yang lebih inklusif ganjar pranowo terbukti lebih baik”

(Doc 3) = “Program keluarga miskin sarjana mencerminkan kepedulian ganjar pranowo terhadap nasib keluarga kurang mampu ganjar mahfud pasangan yang dapat membawa kepedulian ini ke tingkat nasional bersama ganjar kita”

Setelah dilakukan text preprocessing maka data tersebut menjadi seperti:

(Doc 1) = ['semangat', 'positif', 'atikoh', 'sebar', 'lapis', 'masyarakat', 'bangkit', 'harap', 'cerah', 'ganjarmahfud', 'ganjar', 'pranowo', 'bukti']

(Doc 2) = ['rakyat', 'optimis', 'ganjar', 'mahfud', 'bawa', 'indonesia', 'negara', 'inklusif', 'ganjar', 'pranowo', 'bukti']

(Doc 3) = ['program', 'keluarga', 'miskin', 'sarjana', 'cermin', 'peduli', 'ganjar', 'pranowo', 'nasib', 'keluarga', 'ganjar', 'mahfud', 'pasang', 'bawa', 'peduli', 'tingkat', 'nasional', 'ganjar']

Tahap selanjutnya yakni perhitungan TFIDF menggunakan word vector. Perhitungan jumlah kata pada setiap dokumen dinamakan proses TF, sedangkan IDF adalah mengurangi bobot suatu kata apabila kemunculan kata

tersebut tersebar banyak pada setiap dokumen. Adapun perhitungan TF ada pada tabel 4.7.

Tabel 4. 7 Contoh Perhitungan TF (Term-Frequency)

Kata	TF		
	D1	D2	D3
Semangat	1	0	0
Positif	1	0	0
Atikoh	1	0	0
Sebar	1	0	0
Lapis	1	0	0
Masyarakat	1	0	0
Bangkit	1	0	0
Harap	1	0	0
Cerah	1	0	0
Ganjarmahfud	1	0	0
Ganjar	1	2	3
Pranowo	1	1	1
Bukti	1	1	0
Rakyat	0	1	0

Optimis	0	1	0
Mahfud	0	1	1
Bawa	0	1	1
Indonesia	0	1	0
Negara	0	1	0
Inklusif	0	1	0
Program	0	0	1
Keluarga	0	0	2
Miskin	0	0	1
Sarjana	0	0	1
Cermin	0	0	1
Peduli	0	0	2
Nasib	0	0	1
Pasang	0	0	1
Tingkat	0	0	1
Nasional	0	0	1

Berdasarkan tabel diatas maka nilai DF telah didapatkan dimana sebagai contoh pengunaan jumlah dokumen pada tabel

yaitu sejumlah tiga komentar. Dengan demikian diketahui dokumen atau D = 3. Selanjutnya akan dilakukan perhitungan IDF dan TFIDF menggunakan rumus yang dijelaskan pada bab sebelumnya dan selanjutnya akan di implementasikan pada tanel 4.10.

Tabel 4. 8 Contoh Perhitungan TF-IDF

Kata	TF			IDF	IDF+1	TF*IDF		
	D1	D2	D3	(Log(D/DF)		D1	D2	D3
Semangat	1	0	0	0,477	1,477	1,477	0	0
Positif	1	0	0	0,477	1,477	1,477	0	0
Atikoh	1	0	0	0,477	1,477	1,477	0	0
Sebar	1	0	0	0,477	1,477	1,477	0	0
Lapis	1	0	0	0,477	1,477	1,477	0	0
Masyarakat	1	0	0	0,477	1,477	1,477	0	0
Bangkit	1	0	0	0,477	1,477	1,477	0	0
Harap	1	0	0	0,477	1,477	1,477	0	0
Cerah	1	0	0	0,477	1,477	1,477	0	0

Ganjarmahfud	1	0	0	0,477	1,477	1,477	0	0
Ganjar	1	2	3	0	1	1	2	3
Pranowo	1	1	1	0	1	1	1	1
Bukti	1	1	0	0,176	1.176	1,176	1,176	0
Rakyat	0	1	0	0,477	1,477	0	1,477	0
Optimis	0	1	0	0,477	1,477	0	1,477	0
Mahfud	0	1	1	0,176	1.176	0	1,176	1,176
Bawa	0	1	1	0,176	1.176	0	1,176	1,176
Indonesia	0	1	0	0,477	1,477	0	1,477	0
Negara	0	1	0	0,477	1,477	0	1,477	0
Inklusif	0	1	0	0,477	1,477	0	1,477	0
Program	0	0	1	0,477	1,477	0	0	1,477
Keluarga	0	0	2	0,477	1.477	0	0	2,954
Miskin	0	0	1	0,477	1,477	0	0	1,477
Sarjana	0	0	1	0,477	1,477	0	0	1,477
Cermin	0	0	1	0,477	1,477	0	0	1,477
Peduli	0	0	2	0,477	1.477	0	0	2,954
Nasib	0	0	1	0,477	1,477	0	0	1,477
Pasang	0	0	1	0,477	1,477	0	0	1,477

Tingkat	0	0	1	0,477	1,477	0	0	1,477
Nasional	0	0	1	0,477	1,477	0	0	1,477

Hasil dari TF-IDF tersebut jika dijabarkan dalam bentuk array, maka akan seperti berikut:

Array([

[1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 1.477, 1, 1, 1.176, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.176, 1.477, 1.477, 1.176, 1.176, 1.477, 1.477, 1.477, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1, 0, 0, 0, 1.176, 1.176, 0, 0, 0, 1.477, 2.954, 1.477, 1.477, 1.477, 2.954, 1.477, 1.477, 1.477, 1.477])

Pada array tersebut setiap barisnya mengimplementasikan TF-IDF setiap kata pada dokumen. Dari hasil tersebut dapat terbukti bahwa TF menunjukkan banyaknya jumlah data, sedangkan IDF adalah menunjukkan sering tidaknya kata tersebut muncul pada setiap dokumen, semakin banyak muncul kata, nilai IDFnya semakin kecil.

F. Klasifikasi Naive Bayes

Setelah dilakukannya pemrosesan teks pada data komentar, kemudian dibagi datanya menjadi data latih dan data uji lalu diubahnya teks menjadi *vector* pada proses TF-IDF langkah selanjutnya adalah proses klasifikasi. Dalam algoritma klasifikasi *naive bayes*, sentiment dilakukan dengan menghitung nilai probabilitas, mana yang nilainya lebih tinggi antara positif dan negatifnya nantinya akan menjadi label sentimen pada komentar tersebut. Contoh proses perhitungan sentimen dalam algoritma *naïve bayes* secara manual sebagai berikut:

(Doc) = "Ganjar mahfud bikin greget koruptor langsung bawa ke nusakambangan ganjar pranowo terbukti lebih baik".

1. Data.

a. Bikin (positif)
b. Langsung (positif)
c. Bawa (positif)
d. Bukti (positif)
e. Greget (negatif)
f. Koruptor (negatif)

2. Perhitungan Prior

a. Jumlah prior positif (N_{Pos}) = 4

b. Jumlah prior negative (N_{Neg}) = 2

c. Probabilitas prior positif

$$\text{P (Positif)} = \frac{N_{Pos}}{N_{Pos}+N_{Neg}} = \frac{4}{4+2} = \frac{4}{6} = 0{,}67$$

d. Probabilitas prior negatif

$$\text{P (Negatif)} = \frac{N_{neg}}{N_{Pos}+N_{Neg}} = \frac{2}{4+2} = \frac{2}{6} = 0{,}33$$

3. Menghitung probabilitas likelihood

a. P (“Bikin”|pos) = $\frac{N_a}{N_{Pos}} = \frac{1}{4} = 0{,}25$

b. P (“Bikin”|neg) = $\frac{N_a}{N_{Neg}} = \frac{1}{2} = 0{,}5$

c. P (“Langsung”|pos) = $\frac{N_a}{N_{Pos}} = \frac{1}{4} = 0{,}25$

d. P (“Langsung”|neg) = $\frac{N_a}{N_{Neg}} = \frac{1}{2} = 0{,}5$

e. P (“Bawa”|pos) = $\frac{N_a}{N_{Pos}} = \frac{1}{4} = 0{,}25$

f. P (“Bawa”|neg) = $\frac{N_a}{N_{Neg}} = \frac{1}{2} = 0{,}5$

g. P (“Bukti”|pos) = $\frac{N_a}{N_{Pos}} = \frac{1}{4} = 0{,}25$

h. P (“Bukti”|neg) = $\frac{N_a}{N_{Neg}} = \frac{1}{2} = 0{,}5$

i. P (“Greget”|pos) = $\frac{N_a}{N_{Pos}} = \frac{1}{4} = 0{,}25$

j. P (“Greget”|neg) = $\frac{N_a}{N_{Neg}} = \frac{1}{2} = 0{,}5$

k. P (“Koruptor”|pos) = $\frac{N_a}{N_{Pos}} = \frac{1}{4} = 0{,}25$

l. P (“Koruptor”|neg) = $\frac{N_a}{N_{Neg}} = \frac{1}{2} = 0{,}5$

4. Perhitungan probabilitas keseluruhan:

a. P (Positif | “Bikin Langsung Bawa Bukti Greget Koruptor”) ∝ P (Positif) x P (“Bikin”|Positif) x P (“Langsung”|Positif) x P (“Bawa”|Positif) x P (“Bukti”|Positif) x P (“Greget”|Positif) x P (“Koruptor”|Positif) =

P (Positif | “Bikin Langsung Bawa Bukti Greget Koruptor”) ∝ 0,67*0,25*0,25*0,25*0,25*0,25*0,25

b. P (Negatif | “Bikin Langsung Bawa Bukti”) ∝ P(Negatif) x P (“Bikin”|Negatif) x P (“Langsung”|Negatif) x P (“Bawa”|Negatif) x P (“Bukti”|Negatif) x P (“Greget”|Negatif) x P (“Koruptor”|Negatif)

P (Negatif | “Bikin Langsung Bawa Bukti Greget Koruptor”) ∝ 0,33*0,5*0,5*0,5*0,5*0,5*0,5*0,5

c. Nilai P (Positif | “Bikin Langsung Bawa Bukti Greget Koruptor”) lebih tinggi dari nilai negatifnya.

Dari perhitungan menggunakan rumus bayes diatas maka dikatakan bahwa kalimat “Ganjar mahfud bikin greget koruptor langsung bawa ke nusakambangan ganjar pranowo terbukti lebih baik” Memiliki sentimen positif.

Algoritma *Naïve Bayes* yang bertujuan untuk klasifikasi teks menggunakan frekuensi kata yang diatur dalam bentuk matriks yang jarang (sparse matrix). Algoritma ini memanfaatkan smoothing Laplace untuk mengatasi masalah probabilitas nol. Kelas **'NaiveBayes_inset'** dan **'NaiveBayes'** memiliki metode utama yaitu **'fit', 'predict',** dan **'score'.** Metode **'fit'** menghitung probabilitas prior kelas dan probabilitas fitur dengan menambahkan parameter smoothing alpha. Dalam metode **'predict',** input berupa matriks yang jarang diubah menjadi array dan dihitung log probabilitasnya untuk memprediksi kelas yang memiliki probabilitas tertinggi. Hasil prediksi ini kemudian dapat dievaluasi menggunakan metode **'score'** yang membandingkan prediksi dengan label sebenarnya. Perbedaan utama antara kedua *source code* hanya pada penamaan kelas dan variabel, sedangkan logika dan implementasi algoritmanya tetap sama. Berikut merupakan *source code* untuk melakukan proses klasifikasi Naïve Bayes menggunakan Lexicon Inset yang ditunjukkan pada gambar 4.23.

Model Naïve Bayes Classifier dengan Lexicon Inset

```
#@title Model Naïve Bayes Classifier dengan Lexicon Inset

import numpy as np
from scipy.sparse import issparse
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

class NaiveBayes_inset:
    def __init__(self, alpha=1.0):
        self.alpha = alpha  # Laplace smoothing parameter
        self._classes = None
        self._class_prior = None
        self._feature_count = None
        self._feature_prob = None

    def fit(self, X, y):
        n_samples, n_features = X.shape
        self._classes = np.unique(y)
        n_classes = len(self._classes)

        # Initialize arrays for feature counts and class priors
        self._class_prior = np.zeros(n_classes)
        self._feature_count = np.zeros((n_classes, n_features))
        self._feature_prob = np.zeros((n_classes, n_features))
    def score(self, X, y):
        y_pred = self.predict(X)
        return np.mean(y_pred == y)

# Train the classifier
nb_classifier_inset = NaiveBayes_inset()
nb_classifier_inset.fit(X_train_inset_tfidf, y_train_inset)

# Predict on the test set
y_pred_inset = nb_classifier_inset.predict(X_test_inset_tfidf)
        # Calculate class priors and feature counts
        for i, c in enumerate(self._classes):
            X_c = X[y == c]
            self._class_prior[i] = (len(X_c.data) + self.alpha) / (n_samples + n_classes * self.alpha)
            self._feature_count[i] = np.array(X_c.sum(axis=0)).flatten() + self.alpha
            # Convert to 1D array and add Laplace smoothing

        # Calculate smoothed feature probabilities
        smoothed_class_count = self._feature_count.sum(axis=1)
        self._feature_prob = self._feature_count / smoothed_class_count[:, np.newaxis]

    def _predict_single(self, x):
        # Calculate log probabilities
        log_probs = np.log(self._feature_prob) * x
        log_probs_sum = log_probs.sum(axis=1) + np.log(self._class_prior)

        # Return class with the highest log probability
        return self._classes[np.argmax(log_probs_sum)]

    def predict(self, X):
        if issparse(X):
            return np.array([self._predict_single(x.toarray().flatten()) for x in X])
        else:
            return np.array([self._predict_single(x) for x in X])
```

Gambar 4. 23 Klasifikasi Metode NBC Menggunakan Lexicon Inset

Berikut merupakan *source code* untuk melakukan proses klasifikasi Naïve Bayes menggunakan Ahli Bahasa yang ditunjukkan pada gambar 4.24.

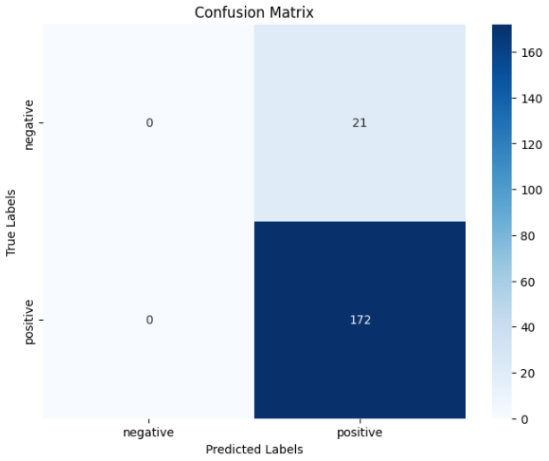
Model Naïve Bayes Classifier

```
#@title Model Naïve Bayes Classifier

import numpy as np
from scipy.sparse import issparse
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, classification_report

class NaiveBayes:
    def __init__(self, alpha=1.0):
        self.alpha = alpha  # Laplace smoothing parameter
        self._classes = None
        self._class_prior = None
        self._feature_count = None
        self._feature_prob = None

    def fit(self, X, y):
        n_samples, n_features = X.shape
        self._classes = np.unique(y)
        n_classes = len(self._classes)

        # Initialize arrays for feature counts and class priors
        self._class_prior = np.zeros(n_classes)
        self._feature_count = np.zeros((n_classes, n_features))
        self._feature_prob = np.zeros((n_classes, n_features))
        # Calculate class priors and feature counts
        for i, c in enumerate(self._classes):
            X_c = X[y == c]
            self._class_prior[i] = (len(X_c.data) + self.alpha) / (n_samples + n_classes * self.alpha)
            self._feature_count[i] = np.array(X_c.sum(axis=0)).flatten() + self.alpha
            # Convert to 1D array and add Laplace smoothing

        # Calculate smoothed feature probabilities
        smoothed_class_count = self._feature_count.sum(axis=1)
        self._feature_prob = self._feature_count / smoothed_class_count[:, np.newaxis]

    def _predict_single(self, x):
        # Calculate log probabilities
        log_probs = np.log(self._feature_prob) * x
        log_probs_sum = log_probs.sum(axis=1) + np.log(self._class_prior)

        # Return class with the highest log probability
        return self._classes[np.argmax(log_probs_sum)]

    def predict(self, X):
        if issparse(X):
            return np.array([self._predict_single(x.toarray().flatten()) for x in X])
        else:
            return np.array([self._predict_single(x) for x in X])
    def score(self, X, y):
        y_pred = self.predict(X)
        return np.mean(y_pred == y)

# Train the classifier
nb_classifier = NaiveBayes()
nb_classifier.fit(X_train_tfidf, y_train)

# Predict on the test set
y_pred = nb_classifier.predict(X_test_tfidf)
```

Gambar 4. 24 Klasifikasi Metode NBC Menggunakan Ahli Bahasa

G. Evaluasi

Selanjutnya dilakukan tahap evaluasi yang digunkan untuk menilai kesesuaian model dalam pengklasifikasian. Pada proses ini akan ditampilkan perbandingan hasil performa model dari data yang telah dilakukan pelabelan dengan menggunakan Lexicon Inset dan Ahli Bahasa. Performa pada model tersebut didapatkan dengan melakukan perhitungan *confusion matrix*. Nilai perhitungan performa diukur dengan menghitung nilai *accuracy, precission, recall,* dan *f1-score.* Berikut *confusion matrix* yang diperoleh dari penerapan algoritma *Naïve Bayes* dengan Lexicon inset yang ditunjukkan pada gambar 4.25.



Gambar 4. 25 Confusion Matrix Data Lexicon Inset

Confusion matrix diatas digunakan untuk melakukan perhitungan nilai *accuracy, precission, recall,* dan *f1-score*. Berikut perhitungan manualnya:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} x100\% = \frac{172 + 0}{172 + 0 + 21 + 0} x100\%$$

$$= 0{,}89\ x\ 100\% = \mathbf{89}\%$$

- Kelas Positif

$$Precission = \frac{\text{TP}}{\text{TP} + \text{FP}} x\ 100\% = \frac{172}{172 + 21}\ x\ 100\%$$

$$= 0{,}89\ x\ 100\%$$

$$= \mathbf{89}\%$$

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} x\ 100\% = \frac{172}{172 + 0} x\ 100\%$$

$$= 1\ x\ 100\%$$

$$= \mathbf{100}\%$$

$$f1 - Score = 2\ x \frac{Precission\ x\ Recall}{Precission + Recall}\ x\ 100\%$$

$$= 2\ x \frac{0{,}89\ X\ 1}{0{,}89 + 1}\ x\ 100\% = 2\ x \frac{0{,}89}{1{,}89}\ x\ 100\%$$

$$= 0{,}94\ x\ 100\%$$

$$= \mathbf{94}\%$$

- Kelas Negatif

$$Precission \quad = \frac{\text{TN}}{\text{TN} + \text{FN}} x\ 100\% = \frac{0}{0+0}\ x\ 100\%$$

$$= \ 0\ x\ 100\%$$

$$= \mathbf{0}\%$$

$$Recall \quad = \frac{\text{TN}}{\text{TN} + \text{FP}} x\ 100\% = \frac{0}{0+21} x\ 100\%$$

$$= \ 0\ x\ 100\%$$

$$= \ \mathbf{0}\%$$

$$f1 - Score \ = 2\ x \frac{Precission\ x\ Recall}{Precission + Recall}\ x\ 100\%$$

$$= 2\ x \frac{0\ \ X\ 0}{0\ + 0}\ x\ 100\% \ = \ 2\ x \frac{0}{0}\ x\ 100\%$$

$$= \ 0\ x\ 100\%$$

$$= \ \mathbf{0}\%$$

Selanjutnya menentukan perhitungan *weighted avarage* yang merupakan rata-rata nilai *precission, recall,* dan *f1-score* dari masing-masing kelas dengan memberikan bobot kepada setiap kelas berdasarkan jumlah contoh dalam kelas tersebut. Berikut perhitungan manualnya:

$$Weighted\ Avg\ Precission = \frac{(0\ x\ 21) + (0{,}89\ x\ 172)}{(21 + 172)}$$

$$= 0{,}79$$

$$Weighted\ Avg\ Recall = \frac{(0\ x\ 21) + (1{,}00\ x\ 172)}{(21 + 172)}$$

$$= 0{,}89$$

$$Weighted\ Avg\ F1 - Score = \frac{(0\ x\ 21) + (0{,}94\ x\ 172)}{(21 + 172)}$$

$$= 0{,}84$$

Berdasarkan perhitungan diatas diketahui performa dari algoritma Naïve Bayes dengan Lexicon Inset memperoleh memperoleh *accuracy* 89%, *precision* 79%, *recall* 89%, *f1-score* 84%. Adapun hasil perhitungan yang dilakukan sistem adalah sebagai berikut:

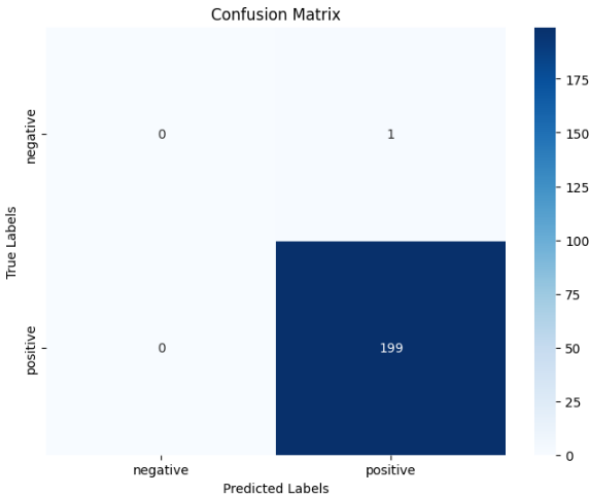
	precision	recall	f1-score	support
negative	0.00	0.00	0.00	21
positive	0.89	1.00	0.94	172
accuracy			0.89	193
macro avg	0.45	0.50	0.47	193
weighted avg	0.79	0.89	0.84	193

Gambar 4. 26 Nilai Performa dari Permodelan NBC dengan Lexicon Inset

Berdasarkan hasil tersebut dapat diketahui bahwa nilai *precission* atau tingkat keberhasilan sistem dalam mencari

ketepatan antara informasi yang diminta oleh pengguna pada nilai negatif sebesar 0%, sedangkan pada nilai positif sebesar 89%. Sehingga dari angka tersebut dapat diartikan bahwa proporsi label yang diprediksi dengan positif lebih tinggi dibandingkan dengan label negatif. Nilai *recall* pada nilai negatif sebesar 0%, sedangkan pada nilai positif sebesar 100%. Artinya nilai *recall* atau tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi yang bernilai positif lebih besar dari label negatif. Untuk nilai *f1-score* dapat dihitung sebesar 0% pada nilai negatif dan 94% pada nilai positif. Sehingga diperoleh total keseluruhan dari nilai precission, recall, serta f1-score pada gambar 4.26 didapatkan nilai *precission* sebesar 79%, nilai *recall* sebesar 89%, dan nilai *f1-score* sebesar 84%.

Selanjutnya perhitungan performa algoritma *Naïve Bayes* menggunakan ahli bahasa. Perhitungan nilai *accuracy, precission, recall,* dan *f1-score* berdasarkan perhitungan dari *confusion matrix* yang diperoleh dari penerapan algoritma *Naïve Bayes* menggunakan ahli bahasa:



Gambar 4. 27 Confusion Matrix Naïve Bayes Menggunakan Ahli Bahasa

Confusion matrix diatas digunakan untuk melakukan perhitungan nilai *accuracy, precission, recall,* dan *f1-score*. Berikut perhitungan manualnya:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} x100\% = \frac{199 + 0}{199 + 0 + 1 + 0} x100\%$$

$$= 0{,}99\ x\ 100\% = \mathbf{99}\%$$

- Kelas Positif

$$Precission = \frac{\text{TP}}{\text{TP} + \text{FP}} x\ 100\% = \frac{199}{199 + 1}\ x\ 100\%$$

$$= 0{,}99\ x\ 100\%$$

$= \mathbf{99}\%$

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} x\ 100\% = \frac{199}{199 + 0} x\ 100\%$$

$= 1\ x\ 100\%$

$= \mathbf{100}\%$

$$f1 - Score = 2\ x \frac{Precission\ x\ Recall}{Precission + Recall}\ x\ 100\%$$

$$= 2\ x \frac{0{,}99\ X\ 1}{0{,}99 + 1}\ x\ 100\% = 2\ x \frac{0{,}99}{1{,}99}\ x\ 100\%$$

$= 0{,}99\ x\ 100\%$

$= \mathbf{99}\%$

- Kelas Negatif

$$Precission = \frac{\text{TN}}{\text{TN} + \text{FN}} x\ 100\% = \frac{0}{0 + 0}\ x\ 100\%$$

$= 0\ x\ 100\%$

$= \mathbf{0}\%$

$$Recall = \frac{\text{TN}}{\text{TN} + \text{FP}} x\ 100\% = \frac{0}{0 + 1} x\ 100\%$$

$= 0\ x\ 100\%$

$= \mathbf{0}\%$

$$f1 - Score = 2\,x\,\frac{Precission\ x\ Recall}{Precission + Recall}\ x\ 100\%$$

$$= 2\,x\,\frac{0\ X\ 0}{0\ +\ 0}\ x\ 100\% = 2\,x\,\frac{0}{0}\ x\ 100\%$$

$$= 0\ x\ 100\%$$

$$= \mathbf{0}\%$$

Selanjutnya menentukan perhitungan *weighted avarage* yang merupakan rata-rata nilai *precission, recall,* dan *f1-score* dari masing-masing kelas dengan memberikan bobot kepada setiap kelas berdasarkan jumlah contoh dalam kelas tersebut. Berikut perhitungan manualnya:

$$Weighted\ Avg\ Precission = \frac{(0\ x\ 1) + (0{,}99\ x\ 199)}{(1 + 199)}$$

$$= 0{,}99$$

$$Weighted\ Avg\ Recall = \frac{(0\ x\ 1) + (1{,}00\ x\ 199)}{(1 + 199)} = 0{,}99$$

$$Weighted\ Avg\ F1 - Score = \frac{(0\ x\ 1) + (1{,}00\ x\ 199)}{(1 + 199)}$$

$$= 0{,}99$$

Berdasarkan perhitungan diatas diketahui performa dari algoritma Naïve Bayes dengan Ahli Bahasa memperoleh *accuracy* 99%, *precision* 99%, *recall* 99%, *f1-score* 99%.

Adapun hasil perhitungan yang dilakukan sistem adalah sebagai berikut:

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	1
positive	0.99	1.00	1.00	199
accuracy			0.99	200
macro avg	0.50	0.50	0.50	200
weighted avg	0.99	0.99	0.99	200

Gambar 4. 28 Nilai Performa NBC dengan Ahli Bahasa

Berdasarkan hasil tersebut dapat diketahui bahwa nilai *precision* atau tingkat keberhasilan sistem dalam mencari ketepatan antara informasi yang diminta oleh pengguna pada nilai negatif sebesar 0%, sedangkan pada nilai positif sebesar 99%. Sehingga dari angka tersebut dapat diartikan bahwa proporsi label yang diprediksi dengan negatif lebih tinggi dibandingkan dengan label positif. Nilai *recall* pada nilai negatif sebesar 0%, sedangkan pada nilai positif sebesar 100%. Artinya nilai *recall* atau tingkat keberhasilan sistem dalam menentukan kembali sebuah informasi yang bernilai positif lebih besar dari label negatif. Untuk nilai *f1-score* dapat dihitung sebesar 0% pada nilai negatif dan 100% pada nilai positif. Sehingga diperoleh total keseluruhan dari nilai *precission, recall* serta *f1-score* pada gambar 4.28 Didapatkan

nilai *precission* sebesar 99%, nilai *recall* sebesar 99%, dan nilai *f1-score* sebesar 99%.

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang dilaksanakan, dapat diambil kesimpulan bahwa :

1. Analisis sentimen menggunakan metode Naïve Bayes Classifier untuk pasangan Ganjar Pranowo dan Mohammad Mahfud MD pada cuitan tweet dapat dilakukan dengan baik. Dari 1000 data komentar awal yang diberi label menggunakan lexicon InSet, setelah dilakukan text preprocessing, jumlah data menjadi 961 dengan 153 sentimen negatif dan 808 sentimen positif. Sedangkan, data yang diberi label dengan didampingi ahli bahasa setelah dilakukan text preprocessing jumlahnya menjadi 997 dengan 4 sentimen negatif dan 993 sentimen positif.
2. Hasil penelitian menunjukkan bahwa metode *Naïve Bayes Classifier* dengan Lexicon Inset memberikan performa dengan *accuracy* sebesar 89%, *precission* 89%, *recall* 89%, dan *f1-score* 84%. Sementara itu, *Naïve Bayes Classifier dengan* ahli bahasa memberikan performa dengan

accuracy sebesar 99%, *precission* 99%, *recall* 99%, dan *f1-score* 99%.

B. Saran

Berdasarkan penelitian yang dilaksanakan, penulis berharap kepada peneliti selanjutnya untuk dapat dikembangkan dan terdapat beberapa saran-saran, yaitu:

1. Algoritma yang berbeda dapat diterapkan agar dapat dilakukan perbandingan dalam mencari hasil klasifikasi yang terbaik. Algoritma lainnya seperti Support Vector Machine (SVM), K-NN, Decision Tree, Random Forest dan lain sebagainya.
2. Data dalam penelitian ini berasal dari cuitan tweet, disarankan pada penelitian berikutnya dapat mengambil data dari media sosial lainnya seperti Instagram, Youtube, Tik Tok, Google Play Store, Facebook, dan lain sebagianya.
3. Penambahan koleksi kamus pada kata yang tidak baku/gaul, karena pada media sosial banyak komentar yang berisikan bahasa yang kurang baku.

DAFTAR PUSTAKA

Adhi Putra, A.D. (2021). Analisis Sentimen Pada Ulasan Pengguna Aplikasi Bibit Dan Bareksa Dengan Algoritma Kkn. *Jatisi (Jurnal Teknik Informatika Dan Sistem Informasi), 8*(2), 636-646. Https://Doi.Org/10.35957/Jatisi.V8i2.962

Akbar, M. N., Darmatasia, D., Mustikasari, M., & Syahwal, M. (2021). Analisis Clustering Teks Tanggapan Masyarakat Di Twitter Terhadap Pembatasan Sosial Berskala Besar Menggunakan Algoritma K-Means. *Jurnal Insypro (Information System And Processing), 6*(1), Https://Doi.Org/10.24252/Insypro.V6i1.23325

Amin, F., & Alfa Razaq, J. (2018). Implementasi Stemmer Bahasa Jawa Dengan Metode Rule Base Approach Pada Sistem Temu Kembali Informasi Dokumen Teks Berbahasa Jawa. *Prosiding Sendi_U*, 199-206.

Arini, A.-, Wardhani, L. K., & Octaviano, D.-. (2020). Perbandingan Seleksi Fitur Term Frequency & Trigram Character Menggunakan Algoritma Naï Ve Bayes Classifier (Nbc) Pada Tweet Hashtag #2019gantipresiden. Kilat, 9(1), 103–114. Https://Doi.Org/10.33322/Kilat.V9i1.878

Aronoff, M., & Fudeman, K. (2011). "What Is Morphology?" Malden, Ma: Wiley-Blackwell.

Astari, N. M. A. J., Divayana, D., & Indrawan, G. (2020). “Analisis Sentimen Dokumen Twitter Mengenai Dampak Virus Corona Menggunakan Metode Naive Bayes Classifier,” Jurnal Sistem Dan Informatika (Jsi), Vol. 15, No. 1, Pp. 27–29, 2020, Doi: 10.30864/Jsi.V15i1.332.

Bbc News Indonesia. (2023) “Mahfud Md Resmi Menjadi Bakal Cawapres Pendamping Ganjar Pranowo,” Bbc News Indonesia, Oct. 18, 2023. Https://Www.Bbc.Com/Indonesia/Articles/C9wlxmmw3zgo (Accessed Feb. 07, 2024)

Bustami. (2014). Penerapan Algoritma Naive Bayes. Jurnal Informatika, 8(1), 884–898

Chaer, A., & Agustina, L. (2010). "Morfologi: Suatu Tinjauan Deskriptif." Jakarta: Rineka Cipta.

Conger, K., & Kate, C. (2023). “So What Do We Call Twitter Now Anyway?”. The New York. Issn 0362-4331.

Cucus, A., Endra, R. Y., & Naralita, T. (2019). Chatter Bot Untuk Konsultasi Akademik Di Perguruan Tinggi. *Explore: Jurnal Sistem Informasi Dan Telematika, 10*(1). Https://Doi.Org/10.36448/Jsit.V10i1.121

Dale, R. (2010). Classical Approaches To Natural Language Processing. In *Handbook Of Natural Language Processing, Second Edition.*

David E., & Wes, T. (2023). "Twitter Is Being Rebranded As X" [Twitter Berganti Nama Menjadi X]. The Verge.

Engelhart, M. D., & Moughamian, H. (1968). Book Reviews : Book Reviews. *Educational And Psychological Measurement, 28*(2), 619-620. Https://Doi.Org/10.1177/001316446802800256

Erfina, A. & Lestari, R. A. (2023). "Sentiment Analysis Of Electric Vehicles Using The Naïve Bayes Algorithm," Jurnal Sistem Informasi (Sistemasi), Vol. 12, No. 1, Pp. 178–185, 2023, Doi: 10.32520/Stmsi.V12i1.2417.

Fatchan, M., & Sugeng, H. (2021). "Analisa Terpilihnya Tri Rismaharini Sebagai Menteri Sosial Dengan Pendekatan Algorithma Naïve Bayes," Journal Of Practical Computer Science, Vol. 1, No. 2, Pp. 50–57, 2021, Doi: 10.37366/Jpcs.V1i2.942.

Fatmawati, M. (2017). *Pengklasteran Laporan Tugas Akhir Berdasarkan Abstrak Menggunakan Metode Rapid Automatic Keyphrase Extraction Dan Average Linkage Hierarchical Clustering*.

Fitri, E. (2020). Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine. Jurnal Transformatika, 18(1), 71. Https://Doi.Org/10.26623/Transformatika.V18i1.2317

Furqan, M., & Shidqi, M. N. (2023). Chatbot Telegram Menggunakan Natural Language Processing. Walisongo Journal Of Information Technology, 5(1), 15-26. Https://Doi.Org/Https://Doi.Org/10.21580/Wjit.2023.5.1.14793

Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics For Multi-Class Classication: An Overview.* 1-17. Https://Doi.Org/10.1007/978-1-4842-8149-9_2

Gridin, I. (2022). Hyperparameter Optimization. In *Automated Deep Learning Using Neural Network Intelligence*. Https://Doi.Org/10.1007/978-1-4842-8149-9_2

Harpizon, H., Kurniawan, R., Iskandar, I., Salambue, R., Budianita, E., & Syafria, F. (2022). "Analisis Sentimen Komentar Di Youtube Tentang Ceramah Ustadz Abdul Somad Menggunakan Algoritma Naïve Bayes," Jurnal Nasional Komputasi Dan Teknologi Informasi, Vol. 5,

No. 1, Pp. 131–140, 2022, Doi: 10.32672/Jnkti.V5i1.4008.

Isaac, Mike, Hirsch, & Lauren. (2022). "Musk's Deal For Twitter Is Worth About $44 Billion". The New York Times. Issn 0362-4331.

Joseph, V. R., & Vakayil, A. (2022). Split: An Optimal Method For Data Splitting. Technometrics, 64(2), 166-176. Htpps://Doi.Org/10.1080/00401706.2021.1921037

Khder, M. A. (2021). Web Scraping Or Web Crawling: State Of Art, Techniques, Approaches And *Application. International Journal Of Advances In Soft Computing And Its Applications, 13*(3), 144-168. Https://Doi.Org/10.15849/Ijasca.211128.11

Lestari, A. R., Perdana, R. S., & Fauzi, M. A. (2017, Desember). Analisis Sentimen Tentang Opini Pilkada Dki 2017 Pada Dokumen Twitter Berbahasa Indonesia Menggunakan Naive Bayes Dan Pembobotan Emoji. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 1*(12), 1718-1724.

Lisangan, E. A., Gormantara, A. & Carolus, R. Y. (2022). "Implementasi Naive Bayes Pada Analisis Sentimen Opini Masyarakat Di Twitter Terhadap Kondisi New Normal Di Indonesia," Konvergensi Teknologi Dan

Sistem Informasi (Konstelasi), Vol. 2, No. 1, Pp. 23–32, 2022, Doi: 10.24002/Konstelasi.V2i1.5609.

Mahfudh, A. A., & Mustofa, H. (2019). Klasifikasi Pemahaman Santri Dalam Pembelajaran Kitab Kuning Menggunakan Algoritma Naïve Bayes Berbasis Forward Selection. *Walisongo Journal Of Information Technology, 1*(2), 101. Https://Doi.Org/10.21580/Wijt.2019.1.2.4529

Marga, N., Isnain, A., & Alita, D. (2021). "Sentimen Analisis Tentang Kebijakan Pemerintah Terhadap Kasus Corona Menggunakan Metode Naive Bayes," Jurnal Informatika Dan Rekayasa Perangkat Lunak (Jatika), Vol. 2, No. 4, Pp. 453–463, 2021, Doi: 10.33365/Jatika.V2i4.1602.

Molina M., & Brett. (2017). "Twitter Overcounted Active Users Since 2014, Shares Surge On Profit Hopes". Usa Today.

Monte, D. & Leslie. (2009). "Swine Flu's Tweet Tweet Causes Online Flutter". Business Standard.

Rahman, A. (2022) "Popularitas Tokoh Politik Di Indonesia," Drone Emprit Publications, Dec. 09, 2022. Https://Pers.Droneemprit.Id/Popularitas-Tokoh-Politik-Di-Indonesia-3/ (Accessed Feb. 07, 2024).

Ratniasih, N. L., Larasati, I., Putri, N., & Kepentingan, K. P. (2023). *Analisis Sentimen Kepuasan Pemangku Kepentingan Menggunakan Metode Naïve Bayes Classifier Dan K-Nearest. 2*, 103-109.

Rifano, E. J., Fauzan, A. C., Makhi, A., Nadya, E., Nasikin, Z., & Putra, F. N. (2020). Text Summarization Menggunakan Library Natural Language Toolkit (Nltk) Berbasis Pemrograman Python. *Ilkomnika: Journal Of Computer Science And Applied Informatics, 2*(1), 8-17. Https://Doi.Org/10.28926/Ilkomnika.V2i1.32

Rosid, M. A., Fitrani, A. S., Astutik, I. R. I., Mulloh, N. I., & Gozali, H. A. (2020). Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi. *Iop Conference Series: Materials Science And Engineering, 874*(1). Https://Doi.Org/10.1088/1757-899x/874/1/012017

Sarjana, P. S., Statistika, D., Matematika, F., Ilmu, D. A. N., & Alam, P. (2017). 291465666.

Sato & Mia. (2022). "Hate Speech Is Soaring On Twitter Under Elon Musk, Report Finds". The Verge.

Soer, D. & Sutrisno, S. (2022). "Analisis Sentimen Terhadap Pemerintahan Ridwan Kamil Sebagai Gubernur Jawa Barat Menggunakan Algoritma Naïve Bayes," In

Prosiding Seminar Nasional Sains Dan Teknologi (Saintek), 2022, Pp. 77–82.

Syah, H., & Witanti, A. (2022). Analisis Sentimen Masyarakat Terhadap Vaksinasi Covid-19 Pada Media Sosial Twitter Menggunakan Algoritma Support Vector Machine (Svm). *Jurnal Sistem Informasi Dan Informatika (Simika), 5*(1), 59-67. Https://Doi.Org/10.47080/Simika.V5i1.1411

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

Valinsky & Jordan. (2023). "Twitter X Logo: Elon Musk Rebrands Social Media Platform| Cnn Business". Cnn

Widowati, T., & Sadikin, M. (2020) "Analisis Sentimen Twitter terhadap Tokoh Publik dengan Algoritma Naive Bayes dan Support Vector Machine," Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer, vol. 11, no. 2, pp. 626–636, 2020, doi: 10.24176/simet.v11i2.4568.

Witten, I. H. (2004). Text Mining. *The Practical Handbook Of Internet Computing*, 14-1-14-22. Https://Doi.Org/10.1201/9780203507223

Wojcik, Stefan, Hughes, & Adam. (2019). “Sizing Up Twitter Users”. Pew Research Center.

Yasar, A., & Saritas, M. M. (2019). Performance Analysis Of Ann And Naïve Bayes Classification. *International Journal Of Intelligent Systems And Applications In Engineering, 7*(2), 88-91. Https://Doi.Org/10.18201/Ijisae.2019252786

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 : Contoh Dokumen Hasil Crawling Data Cuitan *Tweet*

No	Cuitan Tweet
1.	Keberanian untuk berubah dan adaptasi cepat menjadi ciri k3pemimpinan Ganjar-Mahfud yang progresif . Ganjar Pranowo terbukti lebih baik @bekbekbek101 #GanjarPresidenRakyat #GanjarMahfud2024 #Coblos3 https://t.co/PmH5JM9lUC
2.	Kampanye Ganjar-Mahfud kayak karnaval penuh warna dan meriah banget . Ganjar Pranowo terbukti lebih baik @m1lkt3a___ #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/inv1l8Fm4y
3.	Ganjar-Mahfud pasangan yang memiliki kekuatan untuk mengubah harapan menjadi kenyataan yang terp3rcaya. Ganjar Pranowo terbukti lebih baik @honexycat #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/ObdCVtvZbP
4.	Ganjar - Mahfud menjadi p3mimpin yang dikenal dengan kepeduliannya terhadap kesejahteraan rakyat khususnya terkait dengan kepemilikan rumah Ganjar Pranowo terbukti lebih baik @Gerry3239 #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/zfPKJI57gZ
5.	TPN capres-cawapres Ganjar Pranowo-Mahfud MD mengajak relawan memenangkan unggulan mereka dengan tujuan mencegah dinasti politik. https://t.co/DstvhA8IKI

6.	Keberhasilan Ganjar Pranowo - Mahfud dalam menjalankan program bagi-bagi sembako menjadi alasan kuat mengapa rakyat mempercayakan masa depan pada mereka . Bersama Ganjar kita sejahtera @LawrenK95 #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/D7a9ohxhLs
7.	Ganjar-Mahfud punya solusi cerdas untuk pendidikan dan kemiskinan. Ganjar Pranowo - Hanya dia yang betul merakyat dan akan berbuat lebih demi rakyat @DhipaAdrian #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/cg64k46asz
.....	
998	Ganjar Pranowo dan Mahfud MD pantang menyerah untuk membangun Indonesia yang adil dan aman. Mereka adalah p3mimpin yang penuh dedikasi. Bersama Ganjar kita sejahtera @RVrncy9671 #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/I5VzOY3C1n
999	Rakyat merasa didengar dan dihargai melalui Hajatan Rakyat yang digelar oleh Ganjar-Mahfud . Ganjar Pranowo karena kamu berhak mendapatkan yang terbaik @cororong15 #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/K0baXa5ru3
1000	Ganjar-Mahfud dan Ibu Atikoh mengajarkan bahwa kebesaran sebuah negara dapat dicapai melalui kerjasama dan kebersamaan antara pemimpin dan rakyat. Ganjar Pranowo - Hanya dia yang betul merakyat dan akan berbuat lebih #GanjarMahfud2024 #GanjarPresidenRakyat #KratonBersamaRakyat https://t.co/eus55Dvf8D

Lampiran 2 : Contoh Dokumen yang Sudah Diberi Label Lexicon Inset

No	Cuitan Tweet	Sentimen
1.	Keberanian untuk berubah dan adaptasi cepat menjadi ciri kpemimpinan GanjarMahfud yang progresif Ganjar Pranowo terbukti lebih baik	Positif
2.	harapanharapan yang dititipkan di hajatan rakyat adalah pijakan ganjarmahfud untuk menapaki jalan perubahan bersama ganjar pranowo hanya dia yang betul merakyat dan akan berbuat lebih demi rakyat	Negatif
3.	GanjarMahfud pasangan yang memiliki kekuatan untuk mengubah harapan menjadi kenyatan yang terprcaya Ganjar Pranowo terbukti lebih baik	Positif
4.	indonesia membutuhkan pmimpin seperti ganjar pranowo dan mahfud md yang tidak hanya berbicara tapi juga bertindak untuk keamanan bersama bersama ganjar kita sejahtera	Negatif
5.	Kampanye kreatif GanjarMahfud itu kayak obat penawar stres politik bikin kita percaya bahwa perubahan itu bisa datang dengan ceria Ganjar Pranowo karena kamu berhak mendapatkan yang terbaik	Positif
...	...	...
961	gas pol dukung pak ganjar pranowo dan mahfud md	Negatif

Lampiran 3 : Contoh Dokumen yang Sudah Diberi Label Ahli Bahasa

No	Cuitan Tweet	Sentimen
1.	Siap dukung Ganjar Pranowo Mahfud setulus hati.	Positif
2.	Ganjar Mahfud tampaknya lebih memilih gaya ketimbang menunjukkan keberanian dan ketegasan dalam menegakkan keadilan dan integritas. Ganjar Pranowo terbukti lebih baik.	Negatif
3.	Keberanian untuk berubah dan adaptasi cepat menjadi ciri kpemimpinan GanjarMahfud yang progresif Ganjar Pranowo terbukti lebih baik.	Positif
4.	Ganjar Mahfud terlihat lebih antusias memilih pakaian daripada memberikan solusi konkrit membuat warganet semakin skeptis terhadap kemampuan kepemimpinan mereka. Ganjar Pranowo terbukti lebih baik.	Negatif
5.	Slankers hadir sebagai kekuatan moral dalam mendukung Pak Ganjar Pak Mahfud. Ganjar Pranowo terbukti lebih baik.	Positif
...	...	...
997	Ganjar Mahfud seakan berpikir bahwa dukungan terhadap jenama lokal dapat memenangkan hati pemilih tanpa menyadari bahwa substansi kebijakan tetap menjadi penentu utama dalam pilpres. Ganjar Pranowo terbukti mensejahterakan.	Negatif

Lampiran 4 : Daftar Riwayat Hidup

RIWAYAT HIDUP

A. Identitas Diri

Nama	:	Ainun Fatimah
TTL	:	Sengkang, 07 April 2002
Alamat	:	Jalan Selat Makassar Rt.25 Kel. Tanjung Laut, Kec. Bontang Selatan, Kalimantan Timur
No Wa	:	0895359602487
Email	:	ainun_fatimah_2008096016@walisongo.ac.id

B. Riwayat Pendidikan

1. Sekolah Umum

a. Sekolah Dasar (SD) Negeri 013 Bontang Selatan

b. Sekolah Menengah Pertama (SMP) Islam Ar-Riyadh Bontang

c. Sekolah Menengah Atas (SMA) Negeri 1 Bontang

Semarang, 19 Juni 2024

Ainun Fatimah
NIM. 2008096016