PREDIKSI TINGKAT KEPERCAYAAN MASYARAKAT PILPRES TERHADAP 2024 MENGGUNAKAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE

THESIS

Oleh:
EKA RIFUT NUR MUSTAQIM
NIM. 200605210020

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGRI MAULANA MALIK IBRAHIM
MALANG
2024

PREDIKSI TINGKAT KEPERCAYAAN MASYARAKAT TERHADAP PILPRES 2024 MENGGUNAKAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE

THESIS

Diajukan kepada:
Universitas Islam Negri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Magister Komputer (M.Kom)

Oleh:
EKA RIFUT NUR MUSTAQIM
NIM. 200605210020

PROGRAM STUDI MAGISTER INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGRI MAULANA MALIK IBRAHIM
MALANG
2024

PREDIKSI TINGKAT KEPERCAYAAN MASYARAKAT TERHADAP PILPRES 2024 MENGGUNAKAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE

THESIS

Oleh:
EKA RIFUT NUR MUSTAQIM
NIM. 200605210020

Telah diperiksa dan disetujui untuk diuji:
Tanggal: 03 Juni 2024

Pembimbing I,	Pembimbing II,
Dr. Usman Pagalay,M.Si	Dr. Cahyo Crysdian
NIP. 19670118 200501 1 001	NIP. 19740424 200901 1 008

Mengetahui,
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PREDIKSI TINGKAT KEPERCAYAAN MASYARAKAT TERHADAP PILPRES 2024 MENGGUNAKAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE

THESIS

Oleh:
EKA RIFUT NUR MUSTAQIM
NIM. 200605210020

Telah Dipertahankan di Depan Dewan Penguji Thesis
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Magister Komputer (M.Kom)
Tanggal: 03 Juni 2024

Susunan Dewan Penguji		**Tanda Tangan**
Penguji I	Dr. Irwan Budi Santoso, M.Kom NIP. 19770103 200110 1 004	( )
Penguji II	Dr. Ririen Kusumawati, S.Si, M.Kom NIP. 19720309 200501 2 002	( )
Pembimbing I	Dr. Usman Pagalay, M.Si NIP. 19650414 200312 1 001	( )
Pembimbing II	Dr. Cahyo Crysdian NIP. 19740424 200901 1 008	( )

Mengetahui dan Mengesahkan
Ketua Program Studi Magister Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama	: Eka Rifut Nur Mustaqim
NIM	: 200605210020
Program Studi	: Magister Informatika
Fakultas	: Sains dan Teknologi

Menyatakan dengan sebenarnya bahwa Thesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri,kecuali dengan mencantumkan sumber cuplikan pada daftar Pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Thesis ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 03 Juni 2024
Yang membuat pernyataan,

Eka Rifut Nur Mustaqim
NIM 200605210020

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Syukur alhamdulillah penulis hanturkan kehadirat Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Thesis ini dengan baik.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan jazakumullah ahsanal jaza' kepada semua pihak yang telah membantu terselesaikannya Thesis ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Bapak Dr. Usman Pagalay, M.Si dan Bapak Dr. Cahyo Crysdian selaku dosen pembimbing Thesis, yang telah banyak memberikan pengarahan dan pengalaman yang berharga.
2. Segenap sivitas akademika Program Studi Magister Informatika, terutama seluruh Bapak/ Ibu dosen, terima kasih atas segenap ilmu dan bimbingan.
3. Ayahanda dan Ibunda tercinta yang senantiasa memberikan doa dan restunya kepada penulis dalam menuntut ilmu.
4. Semua pihak yang ikut membantu dalam menyelesaikan Thesis ini baik berupa materil maupun moril.

Penulis menyadari bahwa dalam penyusunan Thesis ini masih terdapat kekurangan dan penulis berharap semoga Thesis ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Wassalamu'alaikum Wr. Wb.

Malang, 03 Juni 2024
Penulis,

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Rifut Nur, Eka. 2024. Prediksi Tingkat Kepercayaan Masyarakat Terhadap Pilpres 2024 Menggunakan Metode Naive Bayes dan Support Vector Machine. Thesis. Program Studi Magister Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Usman Pagalay, M.Si (II) Dr. Cahyo Crysdian.

Kata kunci : Big Data, Klasifikasi, Prediksi, TF-IDF, *Naïve Bayes*, *Support Vector Machine* (SVM)

Dalam era modern ini, dunia maya telah menjadi salah satu aspek yang tak terpisahkan dari kehidupan sehari-hari kita. Dengan banyaknya jumlah pengguna maka semakin banyak pula data yang tersimpan serta pemanfaatan yang paling tepat dan optimal adalah tuntutan yang harus terslesaikan. Dalam permasalahan tersebut langkah yang paling tepat adalah melakukan pemanfaatan data untuk tujuan prediksi data dengan menggabungkan teknik TF-IDF dengan 2 metode yang berbeda, disimpulkan bahwa model pada teknik TF-IDF menggunakan metode SVM menunjukkan kinerja yang lebih unggul. Secara khusus, model SVM memiliki tingkat akurasi sebesar 81%, sedangkan model Naïve Bayes hanya mencapai 79%. Lebih lanjut, precision pada model SVM mencapai 76%, sedangkan pada model Naïve Bayes hanya sebesar 70%. Meskipun recall model SVM hanya sedikit lebih tinggi daripada Naïve bayes yaitu 83%: 80%, namun nilai F1-score yang mencapai 79% pada model SVM menunjukkan keseimbangan yang baik antara ketepatan dan keberhasilan dalam menemukan informasi, sedangkan model Naïve bayes hanya mencapai 74%. Kemudian dari hasil evaluasi penerapan kedua model yang telah dilakukan terdapat beberpa permasalahan yang ditemukan dalam penerapanya anatara lain. Pertama, ukuran korpus teks yang besar dapat memperlambat waktu processing karena memerlukan sumber daya komputasi yang signifikan untuk melatih model. Kedua, masalah ketidak seimbangan kelas dalam data sentimen dapat menyebabkan performa yang tidak optimal atau machine akan cendrung mengarah pada jumlah kelas data yang tinggi. Kesimpulan akhir dari penelitian ini mendukung penggunaan teknik TF-IDF dengan metode SVM sebagai pendekatan yang lebih efektif dalam melakukan prediksi data dibandingkan dengan teknik TF-IDF + Naïve Bayes.

ABSTRACT

Rifut Nur, Eka. 2024. Predicting the Level of Public Trust in the 2024 Presidential Election Using the Naive Bayes Method and Support Vector Machine. Thesis. Master of Informatics Study Program, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University, Malang. Supervisor: (I) Dr. Usman Pagalay, M.Si (II) Dr. Cahyo Crysdian.

Keywords: Big Data, *Classification*, *Prediction*, TF-IDF, *Naïve Bayes*, *Support Vector Machine* (SVM)

In this modern era, cyberspace has become an inseparable aspect of our daily lives. Cyberspace, or the internet, is the result of advances in information technology that have revolutionized the world over the last few decades. With the large number of users, the more data is stored and the most appropriate and optimal use of it is a demand that must be resolved. In this problem, the most appropriate step is to utilize data for data prediction purposes by combining the TF-IDF technique with 2 different methods. It was concluded that the model in the TF-IDF technique using the SVM method showed superior performance. Specifically, the SVM model has an accuracy rate of 81%, while the Naïve Bayes model only reaches 79%. Furthermore, the precision in the SVM model reached 76%, while in the *Naïve Bayes* model it was only 70%. Even though the recall of the SVM model is only slightly higher than *Naïve Bayes*, namely 83%: 80%, the F1-score value of 79% for the SVM model shows a good balance between accuracy and success in finding information, while the Naïve Bayes model only reaches 74% . Then, from the results of the evaluation of the implementation of the two models that have been carried out, there are several problems found in their implementation, among others. First, a large text corpus can slow down processing time because it requires significant computing resources to train the model. Second, the problem of class imbalance in sentiment data can cause suboptimal performance or the machine will tend to lead to a high number of data classes. The final conclusion of this research supports the use of the TF-IDF technique with the SVM method as a more effective approach in predicting data compared to the TF-IDF + *Naïve Bayes* technique.

مستخلص البحث

ريفوت نور، إيكا. 2024. التنبؤ بمستوى ثقة الجمهور في الانتخابات الرئاسية 2024 باستخدام طريقة ساذج بايز وآلة ناقل الدعم. أُطرُوحَة. برنامج الماجستير في المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانج. المشرف: (أنا) د. عثمان باجالاي، ماجستير (II) د. كاهيو كريستاليان.

الكلمات المفتاحية: البيانات الضخمة، التصنيف، التنبؤ، TF-IDF، ساذج بايز، آلة ناقل الدعم (SVM)

في هذا العصر الحديث، أصبح الفضاء الإلكتروني جانبًا لا ينفصل عن حياتنا اليومية. الفضاء الإلكتروني، أو الإنترنت، هو نتيجة التقدم في تكنولوجيا المعلومات التي أحدثت ثورة في العالم على مدى العقود القليلة الماضية. ومع تزايد عدد المستخدمين، يتم تخزين المزيد والمزيد من البيانات ويصبح الاستخدام الأمثل والأنسب مطلبًا لا بد منه يتم حلها. في هذه المشكلة، فإن الخطوة الأكثر ملائمة هي استخدام البيانات لأغراض التنبؤ بالبيانات من خلال الجمع بين تقنية TF-IDF وطريقتين مختلفتين. وقد تم التوصل إلى أن النموذج في تقنية TF-IDF باستخدام طريقة SVM أظهر أداءً متفوقًا. على وجه التحديد، يتمتع نموذج SVM بمعدل دقة يبلغ 81%، في حين يصل نموذج Naïve Bayes إلى 79% فقط. علاوة على ذلك، وصلت الدقة في نموذج SVM إلى 76%، بينما في نموذج Naïve Bayes كانت 70% فقط. على الرغم من أن استدعاء نموذج SVM أعلى قليلاً فقط من Naïve Bayes، أي 83٪: 80٪، فإن قيمة درجة F1 البالغة 79٪ لنموذج SVM تظهر توازنًا جيدًا بين الدقة والنجاح في العثور على المعلومات، في حين أن Naïve نموذج بايز يصل إلى 74% فقط. ومن ثم، فمن خلال نتائج تقييم تنفيذ النموذجين اللذين تم تنفيذهما، هناك عدة مشاكل في تنفيذهما، من بين أمور أخرى. أولاً، يمكن لمجموعة نصية كبيرة أن تبطئ وقت المعالجة لأنها تتطلب موارد حاسوبية كبيرة لتدريب النموذج. ثانيًا، يمكن أن تؤدي مشكلة عدم التوازن الطبقي في بيانات المشاعر إلى أداء دون المستوى الأمثل أو تميل الآلة إلى أن تؤدي إلى عدد كبير من فئات البيانات. الاستنتاج النهائي لهذا البحث يدعم استخدام تقنية TF-IDF مع طريقة SVM كنهج أكثر فعالية في التنبؤ بالبيانات مقارنة بتقنية TF-IDF + Naïve Bayes.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di masa maju ini, dunia maya telah menjadi bagian tak terpisahkan dari rutinitas kita. Kemajuan teknologi informasi selama beberapa dekade terakhir telah merevolusi dunia, sehingga memunculkan dunia maya atau internet. Namun, hal ini telah berkembang menjadi lebih dari sekedar teknologi, merupakan ekosistem hidup di mana miliaran orang terhubung, memproduksi dan mengonsumsi informasi, serta menciptakan budaya digital. Media sosial telah mengubah cara kita berinteraksi, berbagi informasi, dan berkomunikasi dengan dunia di sekitar kita. Dalam dekade terakhir, platform-platform seperti Facebook, Twitter, Instagram, LinkedIn, dll telah menjadi bagian tak terpisahkan dari kehidupan kita, memungkinkan kita untuk terhubung dengan teman, keluarga, rekan kerja, dan orang-orang dari seluruh penjuru dunia. Media sosial juga memberikan platform untuk berbagi informasi, berita, artikel. Ini memungkinkan akses lebih luas ke pengetahuan dan pemahaman tentang berbagai topik, penggunaan media sosial telah memiliki beragam efek, baik positif maupun negatif, tergantung pada cara, konteks, dan tujuan penggunaannya, maka semakin banyak pengguna dan aktivitas dalam aplikasi media sosial tersebut semakin banyak juga data-data yang terekam dan tersimpan, kemudian terciptalah yang dikenal dengan istilah big data.

Menurut Eko et al. (2021) Big data dapat menyimpan data yang besar dan saling terintegrasi antara data yang satu dengan yang lainya. Dengan big data kita dapat menggunakan data yang benar dan cepat dari manapun dengan aman dan nyaman. Adanya *big data* tersebut menjadi suatu hal yang sangat menguntungkan

terutama bagi masyarakat luas, para tim survey, maupun individu serta pemerintahan yang memiliki cakupan luas. Menurut hasil penelitian dari Reza et al. (2020) Ada sejumlah faktor yang berkontribusi terhadap kemajuan luar biasa yang telah dicapai dalam bidang data mining, salah satunya adalah pertumbuhan jumlah kumpulan data, serta kemajuan signifikan dalam kemampuan komputasi dan kapasitas media penyimpanan. Data berukuran besar ini sangat beragam, mulai data yang bersifat *positif* dan *negative*, penggunaan data juga sangat tergantung pada orangnya, apalagi jika data tersebut dimasukkan atau di post di dunia maya, jika kebenaran datanya akurat maka bisa membantu orang lain mendapatkan informasi yang benar, tetapi jika data tersebut tidak akurat maka data tersebut bisa menjadi hoax atau informasi yang bohong, tetapi yang paling penting bagaimana cara penggunaan data tersebut dan diolah sebaik mungkin serta bisa mengelompokkan data tersebut termasuk data asli atau bukan. Teknik yang dapat digunakan untuk mengolah dan memanfaatkan *big data* tersebut adalah dengan teknik *data mining*. Teknik yang dapat digunakan untuk mengolah dan memanfaatkan *big data* tersebut adalah dengan teknik *text mining.*

Menurut Lynda et al. (2020) *text mining* merupakan proses yang melibatkan pencarian data dimana data tersebut sebelumnya tidak diketahui, yang mungkin berguna dari sumber informasi tidak terstruktur termasuk arsip bisnis, komentar klien, halaman situs web, dan catatan XML. Dari segi tujuan dan prosedur, *text mining* mirip dengan data mining, namun input dari text mining adalah file data yang tidak terstruktur seperti Word, PDF, teks, XML, dan lain sebagainya. Penambangan teks dapat dimanfaatkan dalam lebih dari satu cara, khususnya ekstraksi data, mengikuti subjek, sinopsis, penyusunan dan pengelompokan data.

Dalam pemrosesan data ini sangat penting karena berkaitan dengan informasi yang disebar luaskan di dunia maya, dan berkaitan dengan kebenaran data tersebut, lebih – lebih jika data tersebut tersebar didunia maya, dan sangat rentang dengan informasi palsu atau *hoaxs*. Penyebaran informasi palsu atau hoaxs juga dijelaskan pada AL - Qur'an surat Al - hujarat ayat 6 dimana Allah berfirman:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِنْ جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَنْ تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْبِحُوا عَلَى مَا فَعَلْتُمْ نَادِمِينَ

Artinya: "Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti, agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu."

Tafsir Al - Hujurat ayat 6 : Allah berfirman, Allah swt memberikan petunjuk kepada kita supaya berhati - hati, agar tidak ceroboh dan tidak tergesa - gesa saat mendapatkan informasi (Ibnu Katsir 7/291).

Dipahami dari nilai-nilai keislaman serta adab dalam bersosial diperlukan kehati-hatian dan tidak mudah gegabah yang lebih untuk melakukan proses pemanfaatan pengolahan *big data* ini dalam hal *text mining* untuk memprediksi data text data guna untuk menghindari prilaku atau penyebaran berita *hoaxs*. Langkah untuk melakukan pemrosesan data secara tepat dan dihadapkan dengan jumlah yang sangat besar, data yang tidak terstruktur dan perubahan data yang sangat cepat maka dalam proses pengolahan data ini memerlukan sebuah mesin yang mempunyai kinerja cepat dan dapat melakukan prosesnya sendiri secara berkala tanpa melibatkan campur tangan manusia lagi, yaitu *machine learnig*.

Seperti dituliskan oleh Stephen (2023) *text mining* merupakan salah satu bagian dari metode information mining yang terkenal dan digunakan oleh banyak orang dibandingkan dengan information mining itu sendiri, karena 80 persen dari seluruh bisnis memiliki dokumen informasi dalam bentuk teks, penambangan teks adalah teknik penambangan data populer yang digunakan banyak orang dibandingkan penambangan data itu sendiri. Penggalian teks merupakan suatu metode untuk menghilangkan atau mencari informasi teks yang sebelumnya tidak jelas untuk menghasilkan informasi baru. Ada beberapa model yang dapat dimanfaatkan dalam *text mining*, salah satunya adalah grouping. Kemudian menurut Akbar et al. (2019) *text mining* adalah subbidang data mining yang berfokus pada analisis data berbasis teks. Penambangan teks adalah langkah pemeriksaan teks yang dilakukan secara konsekuen oleh PC untuk menghapus data berkualitas dari serangkaian teks yang dirangkum dalam sebuah catatan. Pemikiran yang mendasari pembuatan text mining adalah untuk menemukan desain data yang dapat diambil dari teks yang tidak terstruktur. Saat ini, penambangan teks telah mendapat perhatian di berbagai bidang, termasuk keamanan, biomedis, pengembangan pemrograman dan aplikasi, media online, periklanan, skolastik, dan masalah pemerintahan. Penambangan teks harus diterapkan pada studi kasus sesuai dengan prosedur analisis, seperti halnya penambangan data, pra-pemrosesan teks merupakan langkah awal sebelum menggunakan teknik text mining untuk menganalisis data teks, oleh karena itu, analisis penambangan teks dapat dilakukan setelah memperoleh data yang siap diproses.

Prediksi terhadap pemanfaatan informasi yang sangat besar melalui pencarian dan penanganan informasi dari segala jenis artikulasi atau kondisi yang mampu

dilakukan oleh klien yang dikomunikasikan dalam struktur teks melalui sosial media. Penelitian menurut Muhammad (2019) prediksi adalah estimasi sistematis atas sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi yang tersedia saat ini baik dari masa lalu maupun masa yang akan dating atau dari masa sekarang. Prediksi tidak harus memberikan jawaban pasti atas apa yang akan terjadi, melainkan harus berusaha mencari jawaban yang sedekat mungkin dengan apa yang akan terjadi.

1.2 Pernyataan Masalah

Pada proses pengolahan dan pemanfaatan *big data* dengan data berbentuk teks yang bersumber dari media sosial, dengan menerapkan teknik *data mining* menggunakan *machine learning* untuk memprediksi data. Melihat pentingnya proses prediksi data serta permasalahan yang ada, penelitian ini membahas:

a. Metode prediksi apakah yang paling optimal dalam penerapan proses klasifikasi data berupa teks.

1.3 Tujuan Penelitian

Berdasarkan latar belakang dan pernyataan masalah yang sudah dipaparkan, tujuan yang ingin di capai peneliti untuk menangani permasalahan yang ada dengan memaksimalkan proses klasifikasi menggunakan *machine learning* kususnya pada tahap prediksi data ini adalah:

a. Menganalisis proses penerapan metode pada analisis sentimen yang tepat agar mendapatkan hasil yang paling optimal serta akurat dari berbagai perbandingan yang diterapkan untuk memprediksi data.

b. Dapat mengetahui dan menangani permasalahan-permasalahan yang muncul pada saat penerapan analisis sentimen dalam meprediksi data.

1.4 Manfaat Penelitian

Dalam penelitian ini untuk memaksimalkan hasil dari prediksi data menggunakan machine learning dengan data berbentuk teks yang bersumber dari media sosial juga dapat bermanfaat bagi beberapa pihak yaitu:

a. Masyarakat
   - Dapat memberikan informasi terkait isu pemilu.
   - Mendorong partisipasi masyarakat dalam ikut serta memberikan hak pilih.

b. Tim Survei Pemilu
   - Dapat melakukan evaluasi terkait prediksi data yang sesaui dengan kejadian yang sebenarnya nantinya.
   - Dapat melakukan riset kepuasan masyarakat terhadap hasil data yang diprediksi dengan data bersumber dari media soasial.

c. Tim Survei Partai
   - Dapat mengetahui pandangan masyarakat terhadap kandidat calon presiden dan wakil presiden terhadap kinerja partai.
   - Dapat memetakan terkait isu yang sedang memanas di media sosial khususnya twitter.
   - Dapat melakukan pemantauan opini masyarakat terhadap pemilu yang akan di laksanakan.
   - Dapat mengukur elektabilitas kandidat-kandidat partai yang di usulkan.

1.5 Ruang Lingkup Penelitian

Dalam ruang lingkup penelitian agar tidak menyimpang dari pokok bahasan, akan diberikan batasan yang jelas yaitu:

a. Data yang akan diambil berasal dari platform sosial media bernama twitter.
b. Data yang diambil adalah data berupa teks postingan yang dikelompokkan berdasarkan *hashtag* (#) dengan topik pemilihan presiden yang akan di analisis.
c. Waktu priode pengambilan data berdasarkan postingan yang akan diambil adalah dari 01 maret 2022 - 28 desember 2023 dengan jumlah data sebanyak 1000 data.
d. Objek dari penelitian ini adalah masyarakat yang menggunakan media social menggunakan platform twiter.

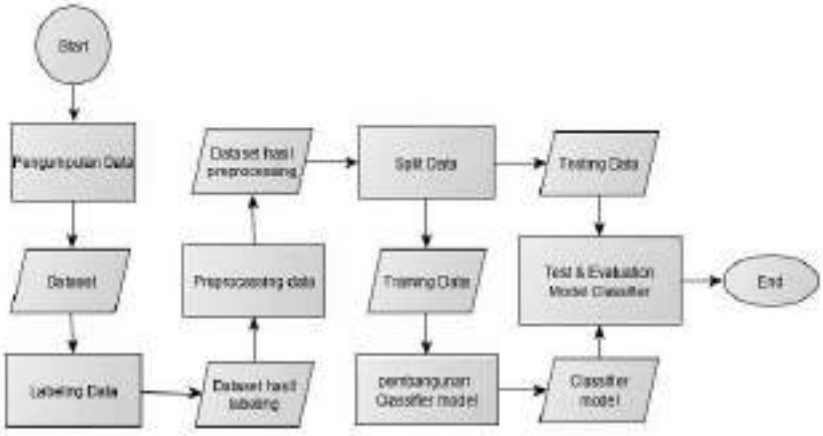
BAB II

STUDI PUSTAKA

Bagian bab ini membahas tentang penelitian terdahulu yang terkait atau memiliki tema penelitian yang sama dengan penelitian yang akan dilakukan yaitu penanganan permasalahan pada proses klasifikasi teks dan prediksi pada machine learning, dari berbagai sumber yang telah diuji kebenarannya seperti jurnal penelitian.

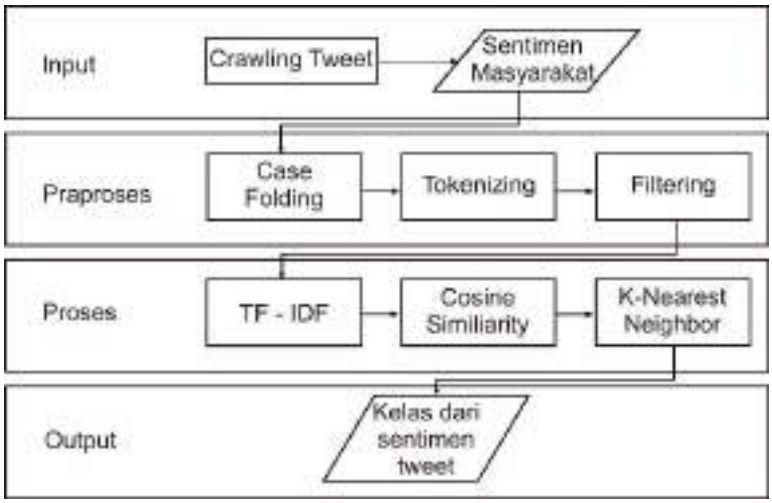
2.1 Klasifikasi

Pada paparan dari studi pustaka kali ini mengulas alur proses penelitian yang diterapkan pada klasifikasi. Peneliti mengutip dalam penelitian yang ditulis oleh Alfian et al. (2019) pada penelitian ini melakukan klasifikasi atau yang umunya pada data mining dapat disebut sebagai *supervised learning* adalah salah satu bagian pada data mining yang terdapat pada *analytical processing* dari data tersebut. Klasifikasi adalah teknik yang berfokus terhadap proses pembelajaran untuk kumpulan data yang telah dipisahkan kedalam bagian label atau grup tertentu, pada proses pembelajaran akan didapatkan model yang mampu memberi prediksi atau waktu bagian data mana yang tidak memiliki grup atau label tertentu.



Gambar 2. 1 langkah-langkah klasifikasi teks

Pada penelitian selanjutnya terkait langkah-langkah proses klasifikasi yang dilakukan oleh Akbar et al. (2019) Klasifikasi teks merupakan teknik untuk memprediksi kategori kelas dari data dan merupakan bagian dari text mining. *Naive Bayes Classifier*, KNN, *Support Vector Machines*, dan *Neural Networks* merupakan beberapa metode klasifikasi yang sering digunakan untuk mengklasifikasikan teks.



Gambar 2. 2 Alur Klasifikasi

Gambar 2.2 merupakan proses klasifikasi, pada penelitian ini juga menjelaskan poin setiap langkahnya sebagai berikut:

1. Input: suatau data inputan dari hasil crawling data tweet berdasarkan pencarian yang sesuai selanjutnya dibandingkan dengan sentiment masyarakat berupa data latih
2. Praproses, terdapat langkah - langkah yang dijalankan diantaranya :
    a. *Case Folding*: perubahan data latih dan data yang akan diuji menjadi huruf kecil kemudian selain huruf ‘a’ - ‘z’ dianggap sebagai karakter.
    b. *Tokenizing* : kalimat yang dipotong menjadi kata - kata.

c. *Filtering* : memilih kata – kata yang penting dari hasil token menggunakan stopword.

3. Proses, terdapat langkah - langkah yang dijalankan diantaranya :

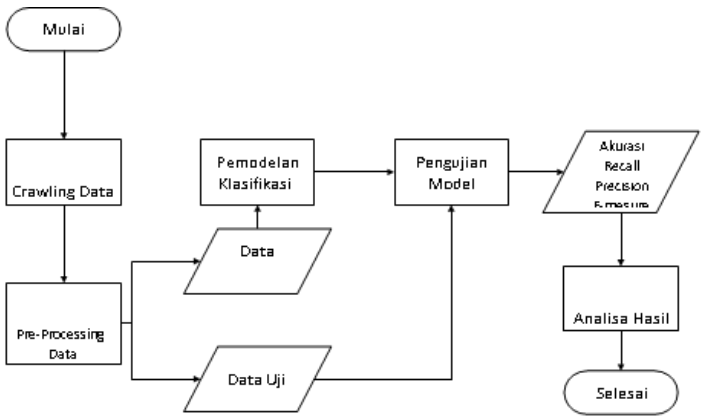
a. *TF-IDF* : proses dari hasil pencarian dari data uji kemudian dilakukan perhitungan bobot menggunakan data latih berdasarkan jumlah kata yang sama.

b. *Cosine Similiarity* : proses menghitung jarak yang mirip dari data uji dan data latih.

c. KNN : proses klasifikasi dari beberapa jarak yang ada akan dihitung dimana jarak yang lebih mendekati dan lebih banyak sejumlah nilai K.

4. Output : didapatkan hasil dari klasifikasi teks berupa data tweet.

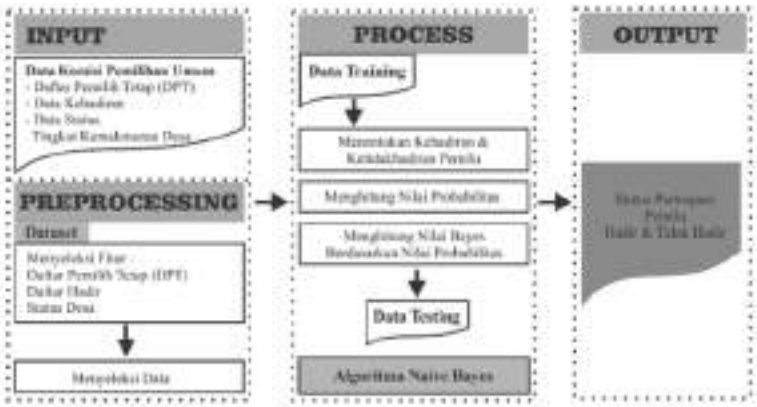
Selanjutnya penelitian yang dilakukan oleh Brata & Kemas (2018) untuk memprediksi dari setiap tweet tentang manfaat transportasi berbasis internet dan mendapatkan ketepatan model yang digunakan. Informasi diperoleh dengan bantuan API (antarmuka Pemrograman) yang diberikan oleh Twitter. Pemrosesan awal data dilakukan untuk membersihkan data setelah dikumpulkan. Setelah informasinya bersih, pilih informasi persiapan dan informasi pengujian.



Gambar 2. 3 Alur klasifikasi teks twiter

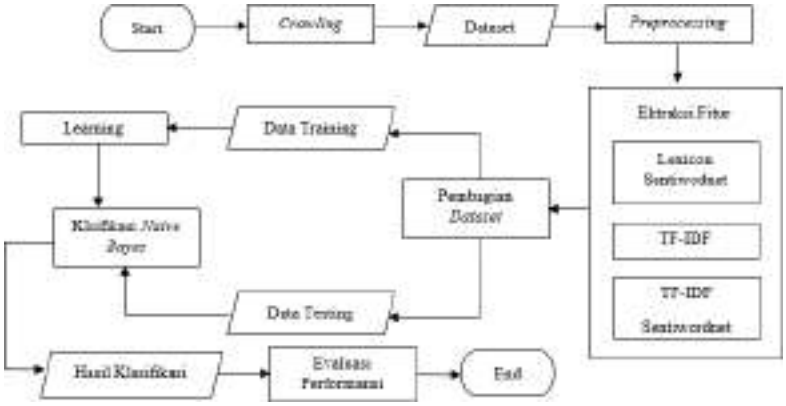
Gambar 2.3 merupakan langkah-langkah klasifikasi teks, langkah awal mengambilan data, kemudian data dibersihkan dengan menggunakan metode pre - processing dengan menggunakan tahapan *case folding, tokenizing, filtering* dan *stemming*. Jika data sudah bersih dilakukan pemilihan data latih dan data uji dimana terdiri dari 1500 dan 500 data dari total 2000 data. Selanjutnya melakukan pemodelan klasifikasi menggunakan data yang akan dilatih dan pengujian model dengan data uji. Langkah paling akhir melakukan perhitungan ketepatan dari model yang digunakan.

Kemudian pada penelitian selanjutnya yang telah dilakukan oleh (Arif, 2019) Data mining ini merupakan ekstraksi informasi atau gambaran yang begitu penting dan menarik dari sekumpulan data yang ada dalam database. Mempunyai salah satu teknik yaitu klasifikasi, yang digunakan untuk memprediksi sebuah nilai dari target variabel kategori. kemudian *Naive bayes Classifier* adalah salah suatu algoritma klasifikasi yang digunakan dalam data mining didasarkan pada keputusan *Bayes*, *Naive Bayes Classifier* mempunyai kemampuan klasifikasi seperti metode *Decision Tree* dan *Neural Network*. algoritma tersebut bisa berguna untuk memprediksi probabilitas keanggotaan suatu kelas.



Gambar 2. 4 Diagram klasifikasi

Pada penelitian ini yang dilakukan oleh Firda & Naim. (2017) sentimen masyarakat pada twitter dengan menggunakan kombinasi ektraksi fitur *Lexicon SentiWordnet* dan *Naive Bayes*, *TF-IDF* dan *Naive Bayes*. Pada proses penelitian analisis sentimen bermula dengan pengumpulan dataset menggunakan teknik crawling data yang berupa sentimen masyarakat pada twitter, penamaan dataset secara manual, preprocessing pada dataset, ekstraksi fitur, lalu klasifikasi digunakan algoritma *Naive Bayes* dan *K-fold Cross Validation*.



Gambar 2.6 Alur klasifikasi

Menurut penelitian yang dilakukan oleh Arindini.(2022) yang membahas berbagai macam perbandingan metode - metode klasifikasi yang umum digunakan, serta menjabarkan kelebihan dan kekurangan dari masing - masing algoritma yang dihasilkan dari beberapa ilmu yang telah diterapkannya dengan hasil sebagai berikut:

Tabel 2.1 Hasil dari Penelitian

Metode	Kelebihan	Kekurangan
Lexicon Based	Metode ekstraksi sentiment suatu kalimat secara otomatis sehingga tidak membutuhkan waktu lama seperti memberi sentiment secara	Metode *lexicon* bergantung pada data kamus sentiment yang digunakan, hal tersebut mempengaruhi label sentiment

	manual.	untuk sebuah data beserta akurasinya.
Naive Bayes	komputasi pada metode *Naive Bayes* sederhana dan mudah dipahami	Dalam metode ini siasumsikan bahwa setiap variable adalah independent dan tidak memiliki korelasi antar variable satu dengan yang lainya
Super Vector Machine	Metode ini bisa melakukan klasifikasi dengan jumlah sample data yang sedikit	Sulit dipakai untuk mengatasi masalah dengan jumlah data yang besar
K-Nearest Neighbor	Tangguh terhadap pengujian data yang *noisy* dan efektif apabila menggunakan data latih besar	Pembelajaran untuk dimana jarak yang tidak jelas dan atribut mana yang harus digunakan.

Penelitian terkait penggunaan beberapa metode klasifikasi yang digunakan oleh Deni et al. (2020) penggunaan jenis algoritma yaitu *Support Vector Machine* (SVM) dan *Naive Bayes* (NB) yang di tambah sebuah seleksi fitur *Genetic Algorithm* (GA) agar klasifikasi bisa meningkat. perbandingan algoritma tersebut untuk diketahui algoritma terbaik untuk diterapkan bersama dengan seleksi fitur *Genetic Algorithm* (GA). Menggunakan *Support Vector Machine* (SVM) dalam klasifikasi karena sebuah teknik machine learning yang populer untuk klasifikasi teks serta memiliki performa yang baik pada banyak domain. Kemampuan SVM dalam mengidentifikasi hyperplane secara terpisah diantara dua kelas berbeda sehingga didapati nilai yang maksimal. *Naive Bayes* (NB) adalah metode klasifikasi teks berdasarkan probilitas kata kunci dalam membandingkan data latih dan data uji. Keduanya perbandingan beberapa tahapan persamaan, yang didapati hasil probalitas tertinggi yang digunakan sebagai kategori dokumen baru. Hasil pengujian data tweet terkait calon gubernur jawa barat periode 2018-2023 dengan

Algoritma *Support Vector Machine* menghasilkan rata-rata akurasi 92,61% dengan AUC 0,950, Algoritma *Naive Bayes* menghasilkan rata-rata akurasi 93,29% dengan AUC 0,525, Algoritma *Support Vector Machine* berbasis *Genetic Algorithm* menghasilkan rata-rata akurasi 93,03% dengan AUC 0,869 dan Algoritma *Naive Bayes* berbasis *Genetic Algorithm* menghasilkan rata-rata akurasi 92,85% dengan AUC 0,543.

Pada penelitian beriktnya yang dilakukan oleh Stephen (2023) hasil dari sebuah penelitian menunjukkan penggunaan algoritma *Naive Bayes* menggunakan kata pertama "vaksinsinovac" mendapatkan hasil sentimen positif 66% dan negatif 34%, kata kunci kedua "vaksinmerahputih" mendapatkan hasil sentimen positif 89% dan negatif 11%, sedangkan metode SVM dengan kata kunci pertama mendapatkan hasil sentimen positif 96% dan negatif 4%, kata kunci kedua mendapatkan nilai sentimen positif 98% dan negatif 2%. Selain dari hasil tersebut juga dilihat bahwa metode Naive Bayes mendapatkan nilai rata rata akurasi lebih besar dengan persentase 85,59%, sedangkan SVM sebesar 84,41%. Data yang diambil dari jejaring sosial media twitter berjumlah 780 data tweet, hasil uji klasifikasi menunjukkan metode SVM memiliki tingkat yang lebih tinggi dari algoritma *Naive Bayes* yaitu sebesar 79,5%. Hasil prediksi menggunakan metode SVM menunjukkan sejumlah 144 positif dan 636 negatif, maka disimpulkan bahwa masyarakat Indonesia terhadap dampak penurunan global sebagai akibat resesi dominan beropini secara negatif.

Pada penelitian yang telah dilakukan oleh Ahmad & Wahyu (2023) *pre processing* data, dan implementasi algoritma SVM serta *Naive Bayes* dalam melakukan klasifikasi sentiment, menghasilkan perhitungan algorithma *Naive*

Bayes dengan *precision* 97%, dan perhitungan dari algorithma SVM diperoleh hasil *Precision* 80%, dapat dilihat dari sebuah hasil analisis sentimen pada diskusi twitter mengenai topik tersebut. Algoritma *Support Vector Machine* (SVM) digunakan dalam klasifikasi diperoleh hasil *Precision* 80% dan *Recall* 93%, dimana menunjukkan hasil yang akurat dan stabil dalam mengklasifikasikan teks, namun *Naive Bayes* secara perhitungan lebih baik dalam proses klasifikasi diperoleh hasil perhitungan yang di hasilkan adalah *Precision* 97% dan *Recall* 97%.

2.2 Analisis Sentimen

Pada paparan dari studi pustaka kali ini mengulas alur proses penelitian yang diterapkan pada analisis sentimen, peneliti mengutip darai penelitian terdahulu yang ditulis oleh Salim & Agung (2023) Analisis sentimen merupakan proses pengumpulan dan memahami pandangan seseorang terhadap suatu topik dalam kehidupan yang sebenarnya. Analisis sentimen pada media sosial adalah teknik atau metode yang digunakan dalam memahami pendapat seseorang melalui aplikasi jejaring sosial. Dengan menggunakan analisis sentimen pada social media, kita dapat mengidentifikasi dan menganalisis ekspresi emosional, pendapat, atau tanggapan seseorang terhadap topik tertentu yang dibagikan melalui media sosial, proses tersebut membantu kita dalam memahami dan menghasilkan informasi yang benar dari data tekstual, sehingga dapat membantu dalam mengatasi hoaxs kontroversial dengan baik.

Menurut penelitian yang dilakukan oleh Fadila & Utomo. (2023) Analisis sentimen merupakan proses yang digunakan untuk memahami pendapat seseorang. Proses tersebut menggunakan sebuah teknik analisis teks pada data teks agar memahami dan pengelompokan emosi seseorang, baik berupa positif maupun

negatif. Faktor yang mempengaruhi penggunaan analisis sentimen merupakan cara pengolahan data teks yang berbeda. Analisis sentimen terdapat beberapa langkah, diantaranya *crawling* data, *pre - processing*, *feature selection*, *classification*, dan *evaluation*. Analisis sentimen dapat merubah data yang tidak terstruktur menjadi data yang terstruktur, dengan demikian kita dapat mengevaluasi dan menghasilkan inovasi di berbagai bidang. Dalam analisis sentimen twitter sering digunakan sebagai sumber data karena strukturnya yang mudah. Peneliti menggunakan berbahasa Indonesia di twitter sebagai sumber data untuk analisis sentimen.

Selanjutnya penelitian yang dilakukan oleh Dewi (2022) Analisis sentimen merupakan proses ekstraksi, secara langsung memproses dan memahami data dalam bentuk teks tidak terstruktur untuk mengambil informasi sentimen yang terkandung dalam kalimat pendapat seseorang. Analisis sentimen dapat menilai pendapat seseorang terhadap suatu topik memiliki kecenderungan negatif dan positif, analisis sentimen dapat diterapkan pada perspektif di semua bidang termasuk ekonomi, politik, masyarakat dan ekonomi hukum. Dalam bidang politik mengenai analisa sentiment tersebut dapat menggunakan algoritma *Naive Bayes* untuk identifikasi kecenderungan pendapat kalayak umum pada twitter terkait pemilihan presiden di Indonesia pada tahun 2019. Hasil dalam penelitian tersebut menghasilkan berupa positif yaitu 79.5% untuk sentimen Jokowoi-Ma'ruf dan 64% untuk Prabowo-Sandi.

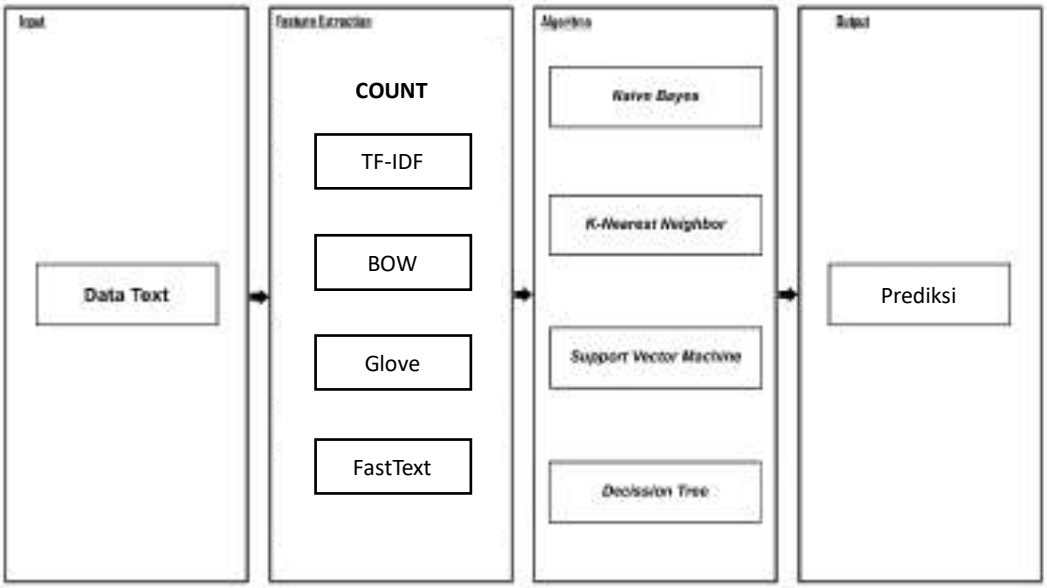
Menurut penelitian yang dilakukan oleh Oktaviami et al (2023) Analisis sentimen merupakan proses dalam *text mining* menggunakan metode data mining untuk pengklasisikasian dari data tidak terstruktur kemudian yang memberikan informasi terkait sentimen. Tujuan dari text mining ini didapatkan informasi yang

sangat penting dari sekumpulan dokumen, maka dari itu sumber data yang dipakai pada *text mining* tersebut tidak terstruktur atau semi terstruktur. Ada tugas khusus didalam yang dilakukan dalam *text mining* berupa pengkategorisasian dan pengelompokkan teks, dalam memberikan sebuah solusi, *text mining* ini mengambil dan mengembangkan berbagai teknik dari berbagai bidang lain, seperti data *mining*, *information retrieval*, statistik dan matematik, *Machine Learning*, *Linguistik*, *Natural Language Processing* dan visualisasi. Kegiatan yang dilakukan dalam *text mining* meliputi ekstraksi dan penyimpanan teks, *preprocessing* teks, pengumpulan data statistic dan analisis sentimen. Analisis sentimen merupakan proses yang berguna untuk evaluasi dan identifikasi pendapat, perasaan, atau emosi yang terkandung dalam teks, seperti dokumen, artikel, ulasan, atau pesan media sosial. Analisis sentimen digunakan untuk memahami dan mengukur reaksi emosional atau pendapat orang terhadap suatu topik, produk, layanan, atau peristiwa tertentu. Dari hasil tersebut dapat berupa pendapat positif, negatif, atau netral. Analisis Sentimen dapat dibagi menjadi 3 level, yaitu :

a. Level Dokumen: pengklasifikasian keseluruhan data ke dalam kelas positif, netral, atau negative.
b. Level Kalimat: menentukan sentimen positif atau negatif dari kalimat tertentu dengan mempertimbangkan susunan kata dalam suatu kalimat tersebut.
c. Level Aspek: menentukan sentimen positif, negatif, atau netral berdasarkan atribut dari suatu entitas.

2.3 Kerangka Teori

Penyusunan kerangka teori ini beracuan dari beberapa penelitian terdahlu yang telah dipaparkan, terkait permasalahan maupun penanganan pada proses klasifikasi menggunakan macnie learning dengan tujuan prediksi data.



Gambar 2.7 Kerangka teori

Pada Gambar 2.7 kerangka teori yang dijelaskan sebagai berikut:

a. Pada pembahasan kali ini adalah mencari proses pengambilan data teks dan membandingkan metode yang digunakan pada setiap penelitian yang pernah dilakukan guna melakukan penyelesaian analisis sentimen seperti yang sudah dipaparkan pada pembahasan sebelumnya, Setelah menentukan proses pengambilan dan pengolahan data twitter, selanjutnya bagian ini adalah mencari metode yang akan diterapkan pada penelitian. Guna mendapatkan hasil yang optimal untuk melakukan proses klasifikasi pada machine learning, berikut adalah paparan hasil perbandingan algoritma yang bisa digunakan sebagai acuan

pemilihan dari penelitian-penelitian terdahulu, berikut adalah tabel beberapa proses metode yang digunakan pada peneliti terdahulu:

Tabel 2.2 Perbandingan metode klasifikasi

No	Judul & Penulis	Dataset	Metode	Hasil
1	Analisis sentimen untuk memprediksi hasil calon pemilu presiden menggunakan *Lexicon Based* dan *Random Forest* (Oktaviami et al. 2023)	Data Teks Twitter	- *Lexicon Based* - *Random Forest*	Metode Random Forest menunjukkan tingkat akurasi yang tinggi, dengan nilai akurasi mencapai 94%.
2	Analisis sentimen masyarakat pada twitter terhadap pemilihan umum 2024 menggunakan Algoritma *Naive Bayes* (Salim et al 2023)	Data Teks Twitter	- *Naive Bayes* - KDD	menunjukkan bahwa terdapat 331 label dengan sentimen positif, sedangkan 261 label dengan sentimen negatif, dan 825 label dengan sentimen netral
3	Analisis sentimen masyarakat Indonesia terhadap dampak penurunan global sebagai akibat resesi di twitter (Stephen 2023)	Data Teks Twitter	- SVM - *Naive Bayes*	Dari 780 data *tweet*, hasil uji klasifikasi menunjukkan algoritma SVM memiliki akurasi lebih tinggi dibandingkan dengan algoritma *Naive Bayes* yaitu sebesar 79,5%.
4	Implementasi algoritma metode *Naive Bayes* dan *Support Vector Machine* tentang pembobolan dan kebocoran data di twitter (Ahmad & Wahyu 2022)	Data Teks Twitter	- SVM - *Naive Bayes*	Algoritma *Support Vector Machine* (SVM) memperoleh hasil *Precision* 80% dan *Recall* 93%, kemudian algoritma *Naïve bayes* secara hitungan lebih unggul dalam proses klasifikasi dengan hasil

				Precision 97% dan *Recall* 97%
5	Komparasi algoritma *Support Vector Machine* dan *Naive Bayes* dengan algoritma genetika pada analisis sentimen calon gubernur Jabar 2018 - 2023 (Deni et al 2020)	Data Teks Twitter	- SVM - *Naive Bayes*	Algoritma SVM berbasis *Genetic Algorithm* memperoleh hasil rata - rata akurasi 93,03% dengan AUC0,869 kemudian Algoritma *Naive Bayes* berbasis memperoleh hasil rata - rata akurasi 92,85% dengan AUC 0,543.

Mengacu pada Tabel 2.2 tentang proses pengambilan data teks yang digunakan oleh peneliti terdahulu dan dipaparkan pada tabel diatas didapat kesimpulan, bahwa tentang perbandingan dan penerapan metode-metode klasifikasi pada machine learning dengan dataset berbentuk dokumen dan teks yang digunakan oleh peneliti terdahulu, maka penelitian ini memilih dan menerapkan metode *Naive Bayes* dan *Super Vector Machine* (SVM) ditinjau berdasakan hasil keunggulan akurasi dan proses klasifikasi yang dihasilkan.

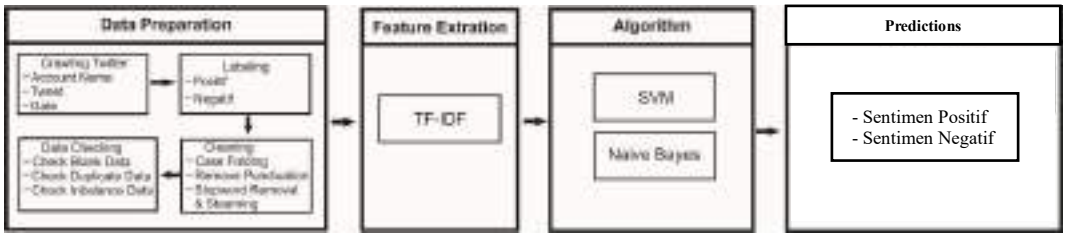
BAB III

DESAIN SISTEM DAN DATA PREPARATION

Bagian bab ini membahas representasi dari sebuah sistem secara rinci pada setiap langkah penelitian serta membahas proses persiapan data mentah hingga siap diproses untuk dilakukan prediksi data menggunakan *machine learning.*

3.1 Desain Sistem

Berikut merupakan representasi alur proses penelitian untuk memprediksi data menggunakan *machine learning* secara utuh diterapkan pada penelitian kali ini, yang dipaparkan pada Gambar 3.1 dibawah disertai penjelasan singkat pada setiap alur berdasarkan cara kerja serta proses didalamnya.



Gambar 3. 1 Desain Sistem

Langkah pertama dari penelitian ini adalah *data preparation* atau mempersiapkan data sebelum diproses yang terdiri mulai dari *crawling* atau penambangan data mentah berbentuk teks yang bersumber dari aplikasi Twitter berisi nama pengguna, isi komentar dan tanggal unggah dengan jumlah data yang akan diambil yaitu sebanyak 1000 data cuitan. Kemudian *labelling* atau pelabelan data dan diproses sebagai syarat untuk pengolahan data *supervised* serta proses terakhir adalah *cleanning*, data tersebut dibersihkan dari fitur-fitur yang tidak penting yang bisa mempengaruhi kinerja dari *machine* jika diproses secara

langsung dan setelah semua siap akan dilanjutkan kedalam pengolahan kata atau *feature extraction*.

Langkah kedua adalah *feature extraction*, dalam proses ini data yang sudah melewati langkah pertama atau *data preparation* maka data sudah memenuhi syarat yaitu bersih dan berlabel yang akan diolah dengan cara kerja menggambarkan kata kedalam bentuk angka berupa vektor, penerapan pada proses klasifikasi ini dengan tujuan agar komputer dapat menangkap dan memproses data berupa teks yang akan diolah pada proses selanjutnya, tidak hanya dengan merepresentasikan kata kedalam angka saja, banyak aspek yang perlu diperhatikan dalam pemrosesan pada tahap *feature extraction* ini seperti hubungan antar kata, kesamaan kata dengan satu makna, dan lain sebagainya. Dikarenakan banyaknya jenis teks berupa bahasa yang dimiliki manusia maka sangat diperlukan teknik yang paling sesuai untuk melakukan pemrosesan pada tahap ini.

Langkah ketiga setelah data melewati proses *cleaning* dan proses *feature extraction,* saat ini data menjadi bentuk numerik berupa vektor lalu pada tahap inilah jug dilakukan proses modeling atau penerapan metode yang telah dipilih dengan mempelajari kata yang sudah di training sebelumnya atau proses utama dari klasifikasi teks dengan menerapkan metode SVM. Algoritma pembelajaran ini merupakan system pembelajaran menggunakan hipotesis berupa fungsi - fungsi linier, yang dilakukan dengan ketelitian yang tinggi untuk medapatkan model hasil akurat serta optimal. Pada dasarnya konsep dan cara kerja dari algoritma SVM adalah berusaha mencari dan menemukan fungsi garis pemisah (*hyperplane*) yang terbaik diantara beberapa fungsi. Formulasi algoritma ini dapat dibedakan menjadi beberapa fungsi atau karnel. terdapat 4 fungsi dan masing-masing memiliki garis

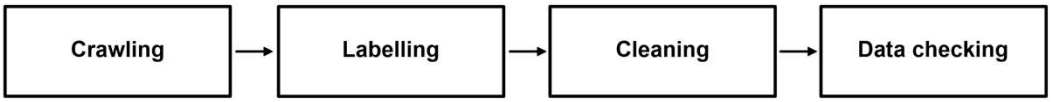
pemisah yang berbeda-beda pada setiap fungsinya. Algoritma SVM mencari dan memilih dari salah-satu garis/*hyperplane* yang paling baik atau paling optimal dalam melakukan klasifikasi dan pemilihan tersebut sangat bergantung pada jenis data dengan tujuan dapat memisahkan antar kelas yang berbeda dengan sempurna.

Langkah keempat atau yang terakhir untuk proses analisis sentimen menggunakan *machine learning* ini adalah melakukan pengukuran dan evaluasi model yang telah dibuat sebelumnya menggunakan acuan hasil pengukuran *confusion matrix* dalam mengevaluasi performa algoritma *machine learning* dengan melihat hasil *true positive* (TP), *true negative* (TN), *false positive* (FP) dan *false negative* (FP). Setelah melakukan proses *evaluation* pada *confusion matrix* acuan yang digunakan selanjunya adalah pengukuran nilai *accuracy* yaitu persentase prediksi benar dari *true positive* dan *true negative*, nilai *precission* yaitu persentase nilai *true positiv*e dari keseluruha nilai positif yang diprediksi, nilai persentase prediksi positif dibandingan dengan *true positive recall* dan nilai *Fl score* yaitu perbandingan ratarata *precission* dan *recall*. Kedua teknik yang diterapkan pada *evaluation* ini bekerja dengan merepresentasikan prediksi dan kondisi dari data *training* atau data pelatihan dengan data *testing* atau data percobaan yang sudah di pisahkan berdasarkan persentase umum yaitu 80 : 20%.

3.2 Data Preparation

Serangkaian proses mempersiapkan data awal atau data primer pada analisis sentimen dengan karakteristik data berbagai atribut yang tidak diperlukan serta tidak terstruktur diubah menjadi data yang terstruktur dengan beberapa langkah yang harus dilakukan agar dapat diproses dan olah oleh komputer. Tahapan ini

merupakan tahap paling awal yang harus dilakükan untuk proses analisis sentimen menggunakan *machine learning*.



Gambar 3. 2 Alur data *preparation*

Pada Gambar 3.2 diatas merupakan langkah secara berurutan untuk menghasilkan data yang siap diproses dan dilakukan analisis, mulai dari *crawling* atau penambangan data mentah berupa cuitan teks kalimat dari Twitter untuk disimpan kemudian *labelling* adalah pelabelan data yang akan diproses sebagai syarat untuk data supervised dan kemudian *cleaning* setelah mendapatkan data dan melabelinya data tersebut dibersihkan dari fitur-fitur yang tidak penting yang bisa mempengaruhi kinerja dari *machine*. Dalam rangkaian proses data *preparation* tersebut penambahan langkah terahir yaitu data *checking* dengan data yang sudah melewati proses *labelling* dan *cleaning* harus di cek terlebih dahulu seperti keseimbangan antar labelnya, adanya data kosong dan data duplikat sebelum dilanjutkan ke pemrosesan, agar mendapatkan data yang benar benar baik untuk diproses. Berikut merupakan penjelasan detail dari setiap langkah pada proses data *preparation*.

3.2.1 Crawling

Pengambilan data atau *crawling* adalah proses menggali lebih jauh dan mengimpor informasi atau data yang telah ditemukan kedalam file lokal komputer pribadi, tahapan paling awal yang harus dilakukan cara ini menggunakan bantuan *google colab* dengan bahasa pemrograman *python* dari

pengguna aplikasi Twitter berupa *tweet* dengan hastag (#pilpres2024) yang berfungsi sebagai penanda untuk *tweet* yang saling berhubungan dan menyampaikan ekspresi terkait dengan apa yang akan diteliti, karakteristik data berupa teks yang masih mentah dan berisikan angka, simbol, emoticon serta huruf di dapatkan dengan sistem *Application Programming Interface* (API) yang secara legal disediakan oleh pihak Twitter. Pengambilan data berdasarkan postingan yang akan diambil adalah mulai dari 1 maret 2022 - 28 desember 2023. Batasan pengambilan hasil data mencakup 1000 entri yang berisi variabel nama akun (*action name*), isi teks yang di unggahan (*tweet),* serta waktu unggahan (*date*). Proses ini dilakukan menggunakan *source code* yang telah disusun sebelumnya, kemudian menggunakan library *Tweepy* untuk mengakses API Twitter, mencari dan mengambil 1000 *tweet* dalam bahasa Indonesia yang berdasarkan kata 'pilpres2024' antara tanggal 1 maret 2022 - 28 desember 2023. Data akun, teks *tweet*,tanggal dan disimpan dalam sebuah *DataFrame* dengan *Pandas* untuk menghasilkan sekumpulan data *excel* yang sudah tersusun berformat .xlsx dengan hasil berikut:

Tabel 3. 1 Hasil *crawling* data Twitter

No	Account Name	Tweet	Date
0	@dedy_pram	Pastilah, Prabowo Anies nggak punya prestasi...…□	15/12/2023 04:04
1	@pengawasan_86	Suasana, pedesaan memang selalu menyenangkan..@pengawasan_86	15/12/2023 14:35
…	…	…	…
890	@andre_rosiade	Prabowo, Gibran berkomitmen untuk mengoptimalkan…□	04/12/2023 15:40
891	@supersaiyaindo	Program-program Prabowo-Gibran, jembatan menuju kemajuan tanpa batas. #02UntukIndonesiaMaju 17Program PrioritasPraGibs	04/12/2023 16:45

3.2.2 Labelling

Dalam hasil data yang sudah di *crawling* atau ditambang pada aplikasi Twitter yang berupa data teks berbentuk kalimat akan diberi label menjadi dua variabel yaitu positif dan negatif. Pada fase ini dilakukan secara manual terkait *tweet* yang bersifat mengarah pada pendapat yang positif atau negatif, dengan dilakukan oleh orang yang berkompeten dibidangnya pada kasus penelitian ini adalah tim survei masing – masing kandidat calon presiden maka akan dilakukan oleh orang yang bekerja di bidang tersebut. Kemudian pada pelabelan data ini menggunakan teknik *crowdsourcing*. Adalah teknik melakukan suatu pekerjaan dengan bantuan beberapa orang (lebih dari 1 orang dengan jumlah ganjil) untuk menentukan suatu pencapaian atau keputusan yang valid dari jumlah voling terbanyak antar keputusan, dengan teknik ini peneliti mengunakan tiga orang tim survei untuk melakukan pelabelan data, dengan hasil dari pelabelan yang telah dilakukan oleh ketiga orang tersebut maka akan mendapatkan suatu keputusan dengan jumlah kesamaan hasil label berjumlah dua berbanding satu, dengan hasil ahir ketentuan label yang berjumlah dua persamaan, maka itulah hasil keputusan label yang akan dipakai.

Tabel 3. 2 Hasil labeling menggunakan teknik *crowdsourcing*

No	Tweet	Label 1	Label 2	Label 3	Hasil
0	pasti anies punya prestasi@dedy_pram	Positif	Positif	Positif	Positif
1	main aman gimana jauh pikir awam saya jelas tahu, wkwkwkwkw…	Negatif	Negatif	Positif	Negatif
…	…	…	…	…	…
890	jir striker prabowo …	Negatif	Negatif	Negatif	Negatif
891	2019 emank milu paling ngeri2 sedapcebong kamp...	Negatif	Negatif	Negatif	Negatif

3.2.3 Cleaning

Dalam *Natural Language Processing* (NLP), sebagian besar data mengandung kata yang berlebihan seperti *stopwords*, *miss-spellings*, *slangs*, dan lain-lain. Pada bagian ini, menjelaskan beberapa teknik dan metode untuk pembersihan teks dan pra - pemrosesan dokumen teks, metode pembelajaran statistik, noise kemudian fitur yang tidak digunakan dapat mempengaruhi proses keseluruhan secara negatif, jadi penghapusan fitur-fitur ini sangat berpengaruh.

a. Case Folding/Capitalization

Pada suatu kalimat yang terdapat pada cuitan *tweet* biasanya dapat berisi huruf besar/kapital dan huruf kecil beberapa campuran huruf tersebut berbentuk kalimat dokumen teks yang dapat mempengaruhi kinerja proses berjalannya *machine learning* jika tidak ditangani, penanganan pada permasalahan ini adalah dengan merubah atau menyamakan semua isi teks tersebut menjadi huruf kecil secara keseluruhan. Pada tahap ini menggunakan teknik *lowercase*, untuk melakukan pembersihan data teks berupa *tweet* dengan mengubah semua hurufnya menjadi huruf kecil atau *lowercase*, dan hasilnya disimpan pada kolom terbaru bernama '*clean1*' kedalam *DataFrame* dengan hasil berikut:

Tabel 3. 3 Hasil *case folding*

No	Tweet	Clean 1
0	pasti anies punya prestasi…	pasti anies punya prestasi…
1	main aman gimana jauh pikir awam saya jelas ba..@pengawasan_86	main aman gimana jauh pikir awam saya jelas ba..@pengawasan_86
…	…	…
890	jir striker prabowo …🕯	jir striker prabowo …🕯

891	2019 emank milu paling ngeri2 sedapcebong kamp...	2019 emank milu paling ngeri2 sedapcebong kamp...

b. Remove Punctuation

Setelah proses penyamarataan huruf atau *case folding* selesai, dokumen teks yang didapat umumnya masih ada berisikan sebuah karakter tanda baca atau karakter khusus dan tidak digunakan seperti url, angka, emotion dan spasi berlebih. *Remove punctuation* ini berperan untuk menghapus sebagian tanda baca yang sangat mengganggu dan memperlambat proses *machine* berjalan, meskipun terkadang tanda baca sebagian juga penting untuk memahami arti dan konteks, dengan melakukan pembersihan teks *tweet* dari elemen seperti nama pengguna, tautan, karakter numerik, tanda baca, dan karakter spesial lainnya, lalu menyimpan hasilnya dalam kolom '*clean2*' dalam *DataFrame* 'df' dengan hasil berikut:

Tabel 3. 4 Hasil *remove punctuation*

No	Clean 1	Clean 2
0	pasti anies punya restasi	anis prestasi
1	main aman gimana jauh pikir awam saya jelas ba... @pengawasan_86	main aman gimana pikir awam banget sikap anies...
…	…	…
890	jir striker prabowo …🔥	jir striker prabowo
891	2019 emank milu paling ngeri2 sedapcebong kamp...	emank milu sedapcebong kampret tebar hasil mil...

c. Stopword Removal

Langkah selanjutnya adalah *stopword removal*, dimana *stopword removal* adalah sekumpulan kata yang tidak saling berhubungan (*irrelevant*). Kata

yang dimaksud adalah jenis kata penghubung seperti di, ke. Berikut merupakan algoritma yang digunakan untuk menghapus *stopwords* dari kata kunci pencarian yang diinputkan oleh pengguna:

- Memasuka *stopword* dari database kedalam *array stoplist* di variable.
- mengurai variabel *string* dengan menggunakan fungsi *string split* ke dalam *array*.
- Inisialisasi kata yang sudah didapatkan berupa variabel yang berisikan nilai-nilai *boolean false*.
- Apakah elemen pada array katakunci samadengan elemen pada *stoplist*? Jika *true*, lakukan langkah 6, jika tidak terpenuhi maka ubahlah nilai menjadi *true*.
- Melakukan langkah ke 4 dan 5 hingga keseluruhan elemen pada *array stoplist* sampai habis.
- Apakah variabel bernilai *false*? Jika kondisi terpenuhi, maka dilakukan langkah 9, dan jika tidak terpenuhi maka lakukan langkah 10.
- Pada array hasil berisikan berisikan elemen kata kunci array.
- Lakukan langkah selanjtnnya yaitu langkah ke 4 sampai dengan langkah ke 9 sehingga seluruh elemen pada array kata kunci habis.

Secara umum teknik ini akan menyeleksi dan menghapus kata-kata penghubung yang kurang penting dalam kalimat atau dokumen, yang jika tidak dilakukan akan menghambat kinerja *machine*. hasil dari proses *stopword removal* menghapus kata - kata penghubung umum yang tidak relevan dari teks *tweet* seperti di, dan, ke, yang dll. Kedalam kolom '*clean2*' disimpan pada '*clean3*' dengan hasil pemrosesan berikut:

Tabel 3. 5 Hasil *stopword removal*

No	Clean 2	Clean 3
0	anies prestasi	anies prestasi
1	main aman gimana pikir awam banget sikap anies	main aman awam banget sikap anies
…	…	…
890	jir striker prabowo	striker prabowo
891	emank milu sedap cebong kampret tebar hasil mil	Sedap cebong kampret tebar hasil

d. Stemming

Langkah yang paling terhakhir untuk proses *cleaning* data atau proses pembersihan data ini adalah *stemming*, yang merupakan proses menghilangkan kata imbuhan (*affixes*) pada kata berimbuhan seperti awalan (*prefixes*), akhiran (*suffixes*), sisipan (*infixes*), dan kombinasi (*confixes*). Contohnya kata lanjutan, keberlanjutan, kemudian di *stem* ke *root word* yaitu "sama". Secara umum *stemming* juga bisa dikatakan sebagai proses atau teknik dalam menemukan kata dasar dari suatu kata umum sebelumnya yang sudah banyak menggunakan variasi kata yang sangatlah banyak persamaan antara saku kata dengan kata yang lain, dengan tanpa menghilangkan makna. *Stemming* ini sendiri juga bisa memiliki fungsi agar menghapus variasi - variasi *morfologi* yang terdapat pada kata, dengan cara menghapus imbuhan - imbuhan pada kata tersebut, kemudian di dapatkan kata yang benar sesuai struktur *morfologi* dalam bahasa indonesia yang baik, dengan tujuan pengurangan jumlah huruf dalam kata yang hanya memiliki satu makna guna mempercepat dan mempermudah saat melakukan proses modeling meggunakan *machine learning*, dengan hasil teknik *stemming*:

- text = df['clean3']: mengambil atau memuat data teks yang telah dibersihkan dan disimpan dalam kolom 'clean3' DataFrame 'df' dan menyimpannya dalam variabel 'text'.
- factory = StemmerFactory(): Ini adalah bagian dari library Sastrawi sebagai *stemming* pada teks bahasa Indonesia. Kode ini membuat objek "factory" dari StemmerFactory.
- stemming = factory.create_stemmer(): Kode ini menggunakan objek "factory" untuk membuat objek "stemming" yang akan digunakan untuk melakukan proses stemming.
- output = [(stemming.stem(token)) for token in text]: Kode ini menggunakan objek "stemming" untuk menerapkan proses stemming pada setiap token (kata) dalam "text". Hasil stemming dari setiap kata disimpan dalam sebuah list yang disebut "output".
- df['clean4'] = output: Hasil stemming yang telah disimpan dalam "output" kemudian dimasukkan ke dalam kolom baru 'clean4' DataFrame 'df'. Ini berarti bahwa kolom 'clean4' akan berisi teks-teks dari kolom 'clean3' yang telah di-stem (dibawa ke bentuk dasarnya).

Tabel 3. 6 Hasil *stemming*

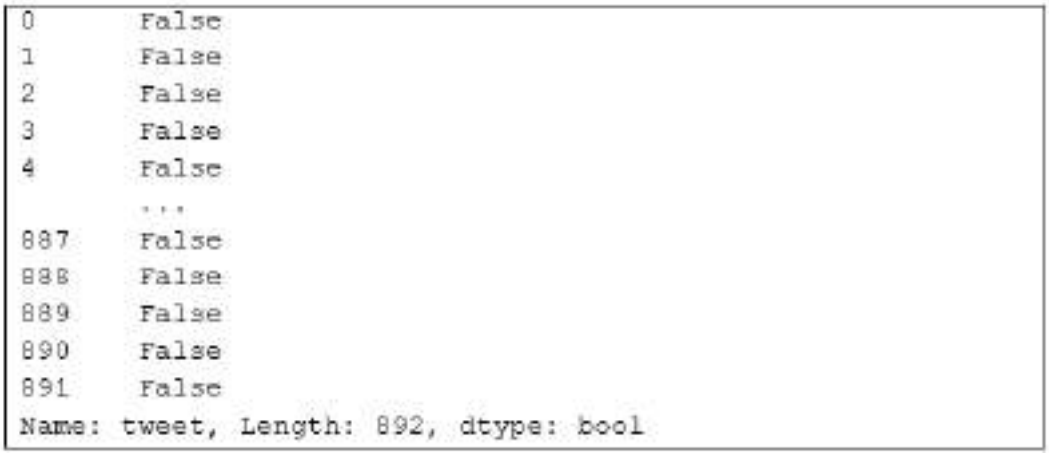
No	Clean 3	Clean 4
0	anies prestasi	Anies prestasi
1	main aman awam banget sikap anies	Main aman awam banget sikap anies
…	…	…
891	striker prabowo	Striker prabowo
891	Sedap cebong kampret tebar hasil	Sedap cebong kampret tebar hasil

3.2.4 Checking

Setelah semua proses terlewati kemudian sebelum data benar-benar diproses data harus dipastikan dulu kevalidtannya dengan melihat jumlah label yang dihasilkan data tersebut harus seimbang dengan ketentuan maksimal kelas data adalah 60%:40% kemudian juga data harus dipastikan tidak ada yang eror dilihat apakah terdapat data duplikat dan data yang kosong pada *tweet* yang ada:

a. Check and Remove Blank Data

   Proses ini merupakan pengecekan data *tweet* keseluruhan yang sudah melewati proses mulai dari *labelling* dan *cleaning*. Proses pengecekan ini dengan melihat apakah terdapat data kosong dari jumlah 892 *tweet* yang dihasilkan dengan proses menggunakan *google colab* dengan bahasa pemrograman *phyton*. Jika terdapat data kosong atau *blank data* maka *machine* akan otomatis melakukan proses penghapusan pada kolom data yang kosong tersebut agar tidak mendapatkan eror atau *missing data* pada saat dilakukan pemrosesan berisikan (df.dropna(inplace=True)) menghapus baris dengan nilai-nilai NaN dalam DataFrame 'df', mengubah DataFrame awal. Ini penting untuk membersihkan data yang tidak lengkap. Kemudian, kode kedua (*df['tweet'].isna()*) memeriksa apakah nilai dalam kolom '*tweet*' adalah NaN, menghasilkan Series dengan nilai *True* jika NaN, *False* jika tidak. Hal ini berguna untuk melakukan validasi data dan memastikan integritasnya sebelum analisis lebih lanjut dengan hasil berikut:

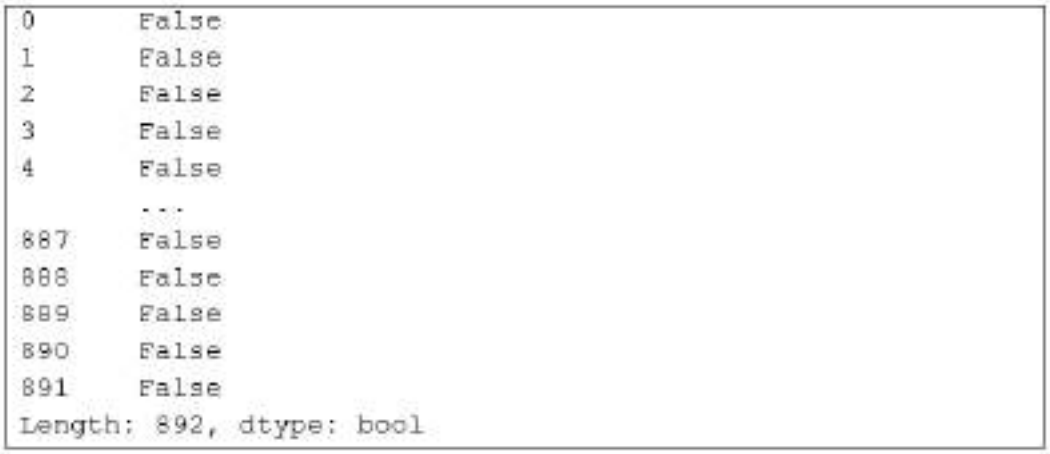


Gambar 3. 3 Hasil *check and remove blank data*

Terlihat hasil pemrosesan pada Gambar 3.3 diatas dari 892 data yang ada, sudah dilakukan proses pengecekan data yang kosong atau *check blank* data tidak ditemukan data kosong pada setiap barisnya, dan jika menemui data yang kosong otomatis program akan menghapus baris yang terdeteksi.

b. Check and Remove Duplicate Data

Selanjutnya setelah semua kolom data dipastikan sudah terisi atau tidak ada data yang kosong, maka dilakukan pengecekan lagi untuk data yang ada lebih dari satu dengan isi teks yang sama atau *duplicate data*, dan jika terdapat data yang *duplicate* makan akan dilakukan penghapusan salah satu data *tweet* tersebut guna mendapatkan proses klasifikasi yang benar-benar valid dan proses juga dapat berjalan dengan cepat. penjelasan pada df.drop_duplicates(*inplace=True*), kita hapus duplikat dalam DataFrame. Kemudian, df.*duplicated*() membantu identifikasi baris duplikat dengan Series *True* dan *False* dengan hasil berikut:



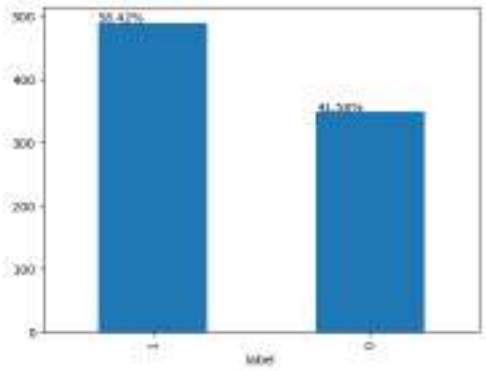
Gambar 3. 4 Hasil *check and remove duplicate data*

Terlihat hasil pemrosesan pada Gambar 3.4 hasil proses pengecekan dan penghapusan data awal setelah dilakukan proses *remove blank data* adalah sejumlah 1000 data kemudian setelah dilakukan proses *remove duplicate* data berkurang menjadi 986 dengan artian masih terdapat data kosong sejumlah 14 baris dan sudah dihapus secara otomatis oleh program. Dengan begitu maka data yang sudah didapatkan tidak lagi berisi data yang kosong (*blank data*) maupun data yang ganda (*duplicate data*) dalam artian data sudah benar-benar bersih dan siap untuk dilakukan pemrosesan tahap pengecekan data yang terahir keseimbangan antar label (*imbalance data*).

c. Check Imbalance Data

Proses ini sangat harus diperhatikan dikarenakan jika tetap dilakukan proses klasifikasi pada label data yang tidak seimbang mesin yang membaca data akan mengklasifikasikan data secara berpihak pada salah satu kelas yang memiliki jumlah terbanyak saja. Fenomena ini dalam *machine learning* dikenal dengan istilah *imbalance dataset* yang mempunyai dampak buruk bagi model yang dibuat mengakibatkan model *overfitting* dengan beberapa penanganan yang dapat dilakukan yaitu *undersampling* atau *oversampling*.

Dan menghasilkan grafik batang yang menunjukkan distribusi data dalam kolom 'label' dari DataFrame df, dengan persentase relatif disertakan sebagai anotasi di atas setiap batang hasil berikut:



Gambar 3. 5 Hasil *check imbalace data*

Terlihat hasil pada pemrosesan Gambar 3.5 hasil melakukan pengecekan otomatis berupa tampilan *interface* plot data dengan menggunakan pemrograman *python* dan perbandingan label positif digambarkan angka 1 yang mempunyai jumlah data sebanyak 489 data dengan persentase sebesar 53.96%, kemudian untuk label negatif digambarkan dengan angka 0 data berjumlah 348 dengan persentase sebesar 44.32%. Kesimpulan hasil dari pengecekan keseimbangan data didapatkan perbedaan antar label tidak terpaut jauh antara label positif dan label negatif, maka data tersebut masih tergolong data yang *balance* dan sudah memenuhi syarat untuk dilakukan pemrosesan ke tahap klasifikasi selanjutnya.

BAB IV

PREDIKSI DATA MENGGUNAKAN METODE *NAIVE BAYES*

Bagian bab ini membahas tentang penerapan dan hasil pengujian pertama teknik yaitu TF-IDF yang dipadukan dengan algoritma klasifikasi *Naïve Bayes* dengan data yang sudah dipersiapkan pada tahap *data preparation* sebelumnya.

4.1 Deskripsi Penelitian

Pemrosesan menggunakan *google colab* serta *python* sebagai bahasa pemrograman yang digunakan, *library tensorflow* untuk pembuatan model *machine learning* dan dataset teks sebagai bahan uji dengan alur berikut:

Gambar 4. 1 Prediksi data TF-IDF dan *Naïve Bayes*

Terlihat pada Gambar 4.1 alur pengujian pada prediksi data berbasis TF-IDF dan *Naïve Bayes* diatas akan dijelaskan secara umum pada langkah pertama adalah dengan *import library* yang dibutuhkan dan dataset yang sudah melalui proses *data preparation* kemudian membagi data tersebut menjadi dua bagian yaitu dengan perbandingan data *training* sebesar 80% dan data *testing* sebesar 20%, kemudian data yang sudah di bagi dua sebelumnya akan diubah kedalam nilai biner berbentuk vektor agar data dapat dibaca dan diproses oleh algoritma *Naïve Bayes* pada *machine learning* proses inilah yang dinamakan *feature extraction* dengan keberhasilan dari proses tersebut akan diukur pada tahap akhir yaitu v*alidation* menggunakan *confusion matrix*.

4.2 Import Data dan Library

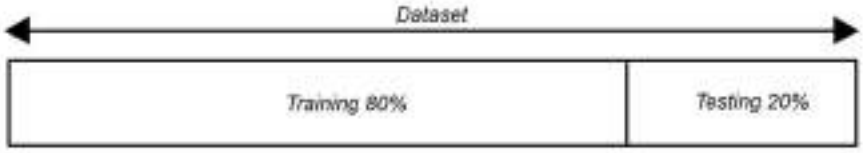
Langkah pertama adalah mempersiapkan data teks yang sudah melewati proses *data preparation* atau pembersihan dan validasi data, yang sudah benar-benar siap untuk dilakukan pemrosesan lebih lanjut yaitu klasifikasi teks pada tahap *feature extraction* menggunakan teknik TF - IDF dengan memasukan beberapa *library* yang dibutuhkan dalam setiap rangkaian proses klasifikasi melakukan proses *import* beberapa *library* yang dibutuhkan yaitu pada baris pertama hingga baris ke tujuh, kemudian mengambil data bernama *clean.csv* dalam file komputer lokal yang diberi nama *data* dan selanjutnya file *data* tersebut dalam kolom *tweet* semua data dipastikan isinya bertipekan *string* (*str*), menampilkan hasil lima data teratas berikut:

Tabel 4. 1 Hasil *import data*

No	label	tweet
0	1	pasti anies punya prestasi
1	0	main aman gimana jauh pikir awam saya jelas ba...
2	0	ayoviralkan rekam jejak anies rasyid baswedan ...
3	1	suasana desa memang selalu senang udara segar ...
4	0	program anies hanya ubah nama rumah sakit se i...

4.3 *Data Partition*

Langkah kedua adalah membagi data yang sudah bersih dan berlabel tersebut kedalam dua bagian yaitu berupa data *training* dan data *testing*, pada pembagian data ini peneliti menggunakan aturan umum (*Rule of Thumb*) dalam persentase data *training* sebesar 80% dan data *testing* sebesar 20%, dengan ilustrasi pada gambar dibawah:



Gambar 4. 2 Pembagian *training* data dan *testing* data

Training pada dataset yang digunakan untuk melatih dan memprediksi data dari sebuah algoritma klaisikasi serta sebagai sumber data mesin yang kita latih untuk mecari korelasi pembelajaran pada pola data tersebut, sedangkan *testing* merupakan bagian dataset yang digunakan untuk menguji dan melihat keakuratan mesin yang sudah melakukan proses pembelajaran dan pengenalan pola yang dilakukan pada data *training*, proses pembagian data '*tweet*' dan kemudian disimpan kedalam variabel X, sementara untuk data *'label'* disimpan kedalam variabel y. Dengan hasil akhir pembagian yang didapatkan data *training* sebanyak 713 dan data *testing* sebanyak 178 dengan perbandingan 80% : 20%.

4.4 *Term Frequency – Inverse Document Frequency* (TF-IDF)

adalah sebuah teknik yang umum digunakan dalam pengolahan teks untuk mengevaluasi seberapa penting sebuah kata dalam sebuah dokumen relatif terhadap seluruh koleksi dokumen. TF - IDF merupakan kombinasi dari dua matrix, *term frequency* (TF) dan *Inverse document frequency* (IDF). Teknik ini dikembangakan oleh Gerard Salton dan rekannya, yang dikenal sebagai bapak pengambilan informasi modern. Gerard Salton adalah seorang ilmuwan komputer yang bekerja di bidang pengambilan informasi dan pencarian teks.

Langkah pertama dalam penerapan teknik TF-IDF ini adalah dengan menginport kelas terlebih dahulu, setelah menginport kelas dilakukan konversi kolom tweet menjadi tipe data berupa string agar sistem bisa mengenali data

tersebut, setelah dilakukan perubahan data tersebut maka langkah selanjutnya mengubah data tweet menjadi vector menggunakan teknik TF-IDF. Untuk membuat kelas dan mengubah data tweet menjadi data string agar dapat dibaca oleh sistem untuk mengekstraksi fitur dari data teks, seperti menghitung frekuensi kata, *term frequency – inverse document frequency* (TF-IDF), kelas ini digunakan sebagai kumpulan dokumen teks menjadi representasi fitur numerik. TF-IDF adalah teknik yang digunakan untuk mengevaluasi berapa penting sebuah kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya. Singkatnya, ini membantu menormalisasi frekuensi kata sehingga kata-kata yang umum tidak terlalu ditekankan dibandingkan kata-kata yang lebih jarang namun mungkin lebih signifikan, dengan hasil sebagai berikut :

```
(0, 1687)   0.937325011732628
  (0, 104)  0.3484563421440753
  (1, 571)  0.31557894213586246
  (1, 498)  0.30202058600516063
  (1, 1181) 0.2915039079542727
  (1, 780)  0.2829111563231096
  (1, 2156) 0.2915039079542727
  (1, 1924) 0.26935280019240776
  (1, 195)  0.2285857714733914
  (1, 150)  0.3346883718179135
  (1, 1634) 0.27564608833651105
  (1, 670)  0.26380172664105855
  (1, 76)   0.30202058600516063
  (1, 1213) 0.27564608833651105
  (1, 104)  0.09806991552131511
  (2, 889)  0.37784630730102403
  (2, 493)  0.30206256911495316
  (2, 1640) 0.1587206040084363
  (2, 215)  0.2553615262995225
  (2, 1751) 0.421476266605111
  (2, 873)  0.3597382380978181
  (2, 1759) 0.38932248772312716
  (2, 163)  0.4469981567059919
  (2, 104)  0.1309787705746457
  (3, 1534) 0.20622308307048462
  :   :
  (889, 2215)     0.3979624485433575
  (889, 666)      0.25608497071010233
```

```
  (889, 1287)     0.23899197407115036
  (889, 1675)     0.14861053824122536
  (890, 1985)     0.6901247076286362
  (890, 888)      0.6901247076286362
  (890, 1675)     0.21784346637202331
  (891, 360)      0.22234899255586157
  (891, 855)      0.22234899255586157
  (891, 505)      0.22234899255586157
  (891, 341)      0.22234899255586157
  (891, 1410)     0.22234899255586157
  (891, 2068)     0.22234899255586157
  (891, 944)      0.44469798511172315
  (891, 1858)     0.22234899255586157
  (891, 545)      0.22234899255586157
  (891, 437)      0.187950989302612
  (891, 2100)     0.1936595523402048
  (891, 1312)     0.3578870902480314
  (891, 736)      0.2974529295451647
  (891, 898)      0.1535529860493624
  (891, 2128)     0.16897262763837115
  (891, 2164)     0.17525570976528537
  (891, 1675)     0.07018626452186194
  (891, 882)      0.17195682862428185
```

Gambar 4. 3 Hasil data tweet menjadi *vector*

Dalam Gambar 4.3 menggambarkan hasil representasi dari sebuah *sparse matrix* yang berisi nilai TF-IDF dari kata-kata dalam dokumen, hasil ini menunjukkan pasangan indeks dan nilai dalam bentuk koordinat (dokumen, kata) diikuti oleh nilai TF-IDF yang sesuai. Ini adalah cara penyimpanan yang efisien untuk matriks yang sebagian besar elemennya adalah nol, karena hanya menyimpan elemen yang tidak nol.

Kemudian proses selanjutnya yaitu menampilkan data dari hasil represntasi *sparse matrix* yang digunakan untuk mendapatkan daftar fitur atau kata-kata yang diekstraksi atau dihasilkan oleh vektorisasi teks menggunakan TF-IDF, untuk mengonversi teks atau dokumen menjadi representasi vektor berdasarkan bobot TF-IDF, Kemudian hasil dari vektorisasi merupakan atribut atau hasil dari metode yang mengembalikan daftar fitur atau kata-kata yang diekstraksi dari teks. Kata-kata ini

merupakan hasil tokenisasi teks awal, yang kemudian dijadikan fitur dalam representasi vektor TF-IDF , yang sesuai dengan indeks kolom dalam matriks TF-IDF atau yang dihasilkan dalam proses vektorisasi teks menggunakan TF-IDF. Dengan cara ini, dapat melihat kata-kata apa saja yang digunakan dalam analisis atau pemodelan berdasarkan TF-IDF .Setiap elemen dalam array tersebut mewakili suatu kata, token, atau string tertentu, array tersebut adalah representasi data yang berisi sejumlah string, di mana setiap string merepresentasikan entitas teks tertentu. Langkah ketiga atau langkah terakhir yang dilakukan setelah memuat vektor kata TF-IDF, ini digunakan untuk menghasilkan matrix. maka akan menghasilkan metrix TF-IDF seperti berikut:

```
matrix([[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., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

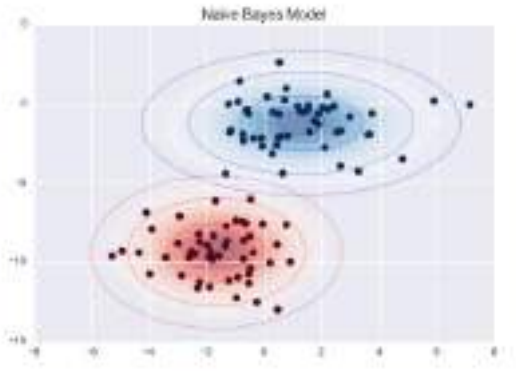
Gambar 4. 4 Hasil matrix TF-IDF

Dari hasil matrix TF-IDF pada Gambar 4.4 menunjukkan bahwa sebagian besar elemen dalam matriks adalah nol. Ini umum dalam matriks TF-IDF karena banyak kata dalam korpus mungkin tidak muncul di sebagian besar dokumen, menghasilkan banyak entri nol, matriks tersebut adalah hasil dari operasi numerik atau pembelajaran mesin, melalui perhitungan TF-IDF.

4.5 *Naïve Bayes*

Setelah data melewati *Naive Bayes Classifier*. Naive Bayes classifier merupakan perhitungan pengelompokan kemungkinan dasar yang menerapkan hipotesis dengan kebebasan tinggi, berdasarkan penyebaran kata-kata dalam data teks. Data pelatihan untuk pengklasifikasi *Naive Bayes* digunakan untuk

memperkirakan probabilitas setiap kategori dalam karakteristik dokumen pengujian. Data latih dan data uji akan dipakai untuk melatih sistem, dan kemudian akan diminta untuk mengetahui nilai fungsi target dari sebuah data.



Gambar 4. 5 Algoritma *Naïve Bayes*

Proses klasifikasi dengan menggunakan *Naive Bayes* didasarkan pada perhitungan probabilitas posterior dalam setiap kelas, kemudian langkah selanjutnya memilih kelas dengan probabilitas terbesar. Persamaan umumnya adalah:

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)}$$

Keterangan :
$P(C|X)$ = merupakan probabilitas posterios dari kelas C diberikan fitur X.
$P(X|C)$ = likelihood merupakan probabilitas fitur X diberikan kepada kelas C.
$P(C)$ = merupakan probabilitas prior dari kelas C.
$P(X)$ = merupakan probabilitas dari fitur X.

Naive Bayes mengasumsikan bahwa fitur-fitur dalam dataset saling independen. Dengan asumsi ini, likelihood *P(X|C)* dapat dipecah menjadi:

$$P(XC) = P(x_1, x_2, \cdot \ldots \cdot x_n|C) = P(x_1|C).P(x_2|C) \ldots P(x_n|C)$$

Sehingga persamaan *Naïve Bayes* menjadi :

$$P(C|X) \propto P(C) \cdot \pi_{i=1}^{n} P(x_i|C)$$

Disini, kita hanya perlu menghitung *P(C)* dan $P(x_i|C)$ dari data pelatihan untuk semua kelas *C,* kemudian langkah selanjutnya melakukan perhitungan :

1. Probabilitas Prior *P(C)* = Probabilitas prior *P(C)* adalah rasio jumlah data dari kelas *C* terhadap total data.
2. Likehood $P(x_i|C)$ = Likelihood $P(x_i|C)$ adalah probabilitas fitur x_i diberikan kelas *C*. ini bisa dihitung dari frekuensi fitur dalam kelas *C*. Dimana , $P(x_i = 1|C)$ probabilitas bahwa x_i muncul dalam kelas *C* dimana terdapat 2 kelas yaitu kelas negatif yang di beri label dengan angka 0 dan kelas positif diberi label dengan angka 1.
3. Probabilitas Posterior *P(C|X)* = Untuk setiap kelas, hitung probabilitas posterior *P(C|X)* dengan mengalikan prior dengan likelihood dari semua fitur.
4. Probabilitas Posterior terbesar = Kelas dengan nilai *P(C|X)* terbesar akam menjadi prediksi.

Perhitungan menggunaka data kelas, dengan menggunakan dua kelas yaitu C_1 dan C_2 dan tiga fitur x_1, x_2, x_3 , data pelatihan

- Data kelas $C_1 = (1,0,1), (1,1,0), (0,1,1)$
- Data kelas $C_2 = (0,0,0), (1,0,0), (0,1,0)$

Langkah 1: hitung probabilitas prior *P(C)*

$$P(C_1) = \frac{3}{6} = 0.5$$
$$P(C_2) = \frac{3}{6} = 0.5$$

Langkah 2: hitung likelihood $P(x_i|C)$ menggunakan smoothing laplace untuk menghindari probabilitas nol :

- $P(x_1 = 1|C_1) = \frac{2+1}{3+1} = 0.6$
- $P(x_2 = 1|C_1) = \frac{2+1}{3+1} = 0.6$
- $P(x_3 = 1|C_1) = \frac{2+1}{3+1} = 0.6$
- $P(x_1 = 1|C_2) = \frac{1+1}{3+2} = 0.4$
- $P(x_2 = 1|C_2) = \frac{1+1}{3+2} = 0.4$
- $P(x_3 = 1|C_1) = \frac{0+1}{3+2} = 0.2$

Langkah 3: hitung probabilitas posterior *P(C|X)* untuk data baru *X* =(1,1,0) :

$$P(C_1|X) \propto P(C_1) \cdot P(x_1 = 1|C_1) \cdot P(x_2 = 1|C_1) \cdot P(x_3 = 0|C_1)$$
$$P(C_1|X) \propto 0.5 \cdot 0.6 \cdot 0.6 \cdot 0.4 = 0.072$$

$P(C_2|X) \propto P(C_2) \cdot P(x_2 = 1|C_2) \cdot P(x_2 = 1|C_2) \cdot P(x_3 = 0|C_2)$
$P(C_2|X) \propto 0.5 \cdot 0.4 \cdot 0.4 \cdot 0.8 = 0.064$

Langkah 4: Klasifikasi

Karena $P(C_1|X) > P(C_2|X)$, maka data X = (1,1,0) diklasifikasikan ke kelas C_1.

Pada tahap berikutnya teks dibaca secara berurutan kemudian menentukan kriteria kategori berdasarkan suatu kata tertentu (kata kunci) yang akan muncul. Algoritma *Naïve Bayes* ini menggunakan asumsi independensi fitur dan distribusi probabilitas tertentu (Gaussian, Multinomial, atau Bernoulli) untuk melakukan klasifikasi. Langkah awal sebelum melakukan *tuning* data menggunakan model *Naïve Bayes* dalam *machine learning* adalah proses spil data, pengubahan tek menjadi numerik, dan sparse matrix menjadi array agar bisa digunakan oleh *machine learning*. langkah-langkah dalam menggunakan *Naïve Bayes :*

a. Split data, Membagi data teks dan label menjadi set pelatihan dan pengujian menggunakan *train_split_data,* agar nanti dapat diolah oleh *machine learning,* pembagian data akan sangat berpengaruh terhadap hasil yang didapat.
b. Pengubahan teks menjadi Numerik, Menggunakan *TfidfVectorizer* dimana nantinya akan digunakan pada tahap tuning data agar dapat merubah data teks menjadi fitur numerik yang bisa digunakan oleh model *machine learning*
c. Perubahan *sparse matrix* menjadi array, Parameter ini Mengubah hasil TF-IDF yang berupa sparse matrix menjadi dense array agar bisa digunakan oleh *Naive Bayes*. Data yang diolah dari TF-IDF berupa sparse matrix agar bisa diolah oleh machine learning maka harus dirubah terlebih dahulu menjadi data berupa array.

Langkah berikutnya sebelum melakukan tuning pada penerapan model *Naïve Bayes* maka dilakukan konfersi data *matrix* terlebih dahulu agar nanti dapat diolah oleh machine learning. Setelah melakukan konfersi matrix maka langkah berikutnya

yaitu tuning data yang nantinya menentukan parameter terbaik dari metode *Naïve Bayes.* Hasil dari menentukan parameter terbaik dengan menggunakan inisialisasi nilai x berisikan data test dan nilai y sebagai hasil dari prediksi data yang diolah oleh mechine learning dalam penerapan metode *Naïve Bayes*, langkah berikutnya adalah menampilkan hasil nilai parameter terbaik. Berikut adalah hasil dari tunning yang telah dilakukan model *Naïve Bayes* dengan mencari hasil nilai parameter yang terbaik:

```
Parameter terbaik: {'var_smoothing': 0.01873817422860384}
```

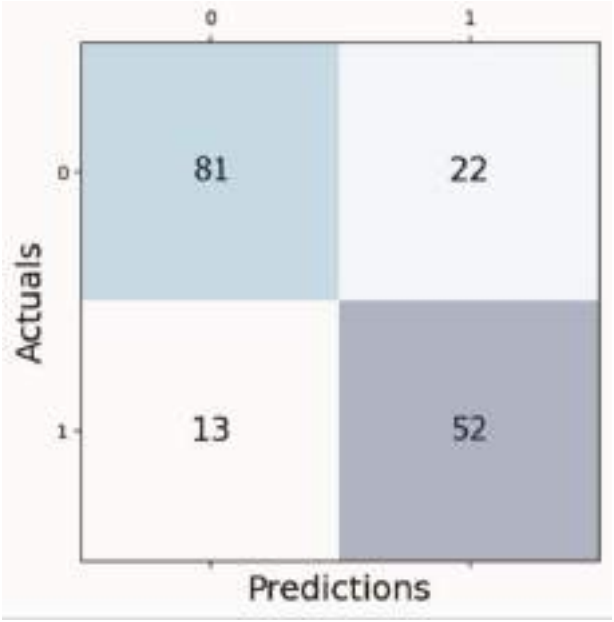
Gambar 4. 6 Hasil parameter *Naïve Bayes*

Hasil dari Gambar 4.6 menunjukkan bahwa proses pencarian telah dilaksanakn untuk menentukan nilai parameter yang hasilnya kinerja model terbaik. Dalam hal ini, parameter yang dimaksud adalah *var_smoothing*, Ini adalah parameter spesifik dalam model *Naive Bayes*.

4.6 *Validation*

Pada langkah terahir proses klasifikasi analisis sentimen ini adalah melakukan pengukuran. *Confusion matrix* merupakan sebuah tabel yang dapat digunakan dalam pemahaman performa model klasifikasi pada *machine learning* dengan cara kerjanya menggambarkan jumlah prediksi *machine* yang benar terhadap data *real* dan jumlah prediksi *machine* yang salah terhadap dara *real* yang dibuat oleh model terhadap data uji, memungkinkan untuk mengevaluasi model tersebut efektif dalam mengklasifikasikan data berdasarkan empat komponen utama yaitu TP (*true positive*), TN (*true negative*), FN (*false negative*) dan FP (*false positive*). Dari empat komponen tersebut, dapat menghitung berbagai *metrix* evaluasi model akurasi, presisi, recall dan nilai *F1 - Score*, yang memberikan pandangan yang mendalam perihal kinerja klasifikasi untuk mengatasi kasus analisis sentimen

menggunakan proses *machine learning* pada studi kasus sentimen positif dan negatif. digunakan untuk menghasilkan dan menampilkan *predictions* dari hasil kerja model *machine learning* terhadap data uji. *predictions* ini memberikan gambaran visual perihal berapa baik model dapat mengklasifikasikan data menjadi kategori yang benar dan seberapa sering terjadi kesalahan prediksi, yang digunakan sebagai acuan atau untuk menentukan hasil dari perhitungan *accuracy, presission, recall* dan *f1 - score* dengan hasil *predictions* sebagai berikut ini:



Gambar 4. 7 Hasil tabel *Predictions*

Hasil tabel *predictions* yang didapatkan pada Gambar 4.7 dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom yaitu:

- TP (*True Positive*), sebanyak 52 data yang positif terklasifikasi benar.
- TN (*True Negative*), sebanyak 81 data yang negatif terklasifikasi benar.
- FP (*false positive*), yaitu sebanyak 22 data positif terklasifikasi salah.
- FN (*False Negative*), yaitu sebanyak 13 data negatif terklasifikasi salah.

Dari informasi tabel *predictions* yang sudah didapat, kemudian adalah melakukan perhitungan *accuracy, precission, recall* dan *f1 – score,* dalam mengukur berapa jauh keberhasilan model dalam mengklasifikasikan data tersebut.

A*ccuracy* (A) persentase prediksi benar dari *true positive* dan *true negative.*

$$A = \frac{(TP+TN)}{(TP+FP+FN+TN)} = \%$$

$$A = \frac{(52+81)}{(52+22+13+81)} = 0.7917$$

$$Accuracy = 79\%$$

Precission (P) persentase prediksi benar dari dari keseluruhan nilai positif.

$$P = \frac{(TP)}{(TP+FP)} = \%$$

$$P = \frac{(52)}{(52+22)} = 0.7027$$

$$Precission = 70\%$$

Recall (R) persentase prediksi *positif* dibandingan dengan *true* positif.

$$R = \frac{(TP)}{(TP+FN)} = \%$$

$$R = \frac{(52)}{(52+13)} = 0.8$$

$$Recall = 80\%$$

F1-Score (F) merupakan pembandingan rata - rata *precission* dan *recall*.

$$F = \frac{2(P\ x\ R)}{P+R} = \%$$

$$F = \frac{2(0.7027\ \ x\ 0.8)}{0.7027\ +\ 0.8} = 0.7483$$

$$F1 - Score = 74\%$$

BAB V

PREDIKSI DATA MENGGUNAKAN METODE SVM

Bagian bab ini membahas tentang penerapan dan hasil pengujian kedua yaitu teknik TF-IDF yang dipadukan dengan metode klasifikasi SVM dengan data yang sudah dipersiapkan pada tahap awal yaitu tahap *data preparation* sebelumnya.

5.1 Deskripsi Penelitian

Pemrosesan menggunakan *google colab* serta *python* sebagai bahasa pemrograman yang digunakan, *library tensorflow* untuk pembuatan model *machine learning* dan dataset teks sebagai bahan uji dengan alur berikut:

Gambar 5. 1 Prediksi data TF-IDF dan SVM

Terlihat pada Gambar 5.1 alur pengujian pada analisis sentimen berbasis TF-IDF dan SVM diatas akan dijelaskan secara umum pada setiap alur yang dilewati oleh data yang diproses. Untuk penjelasan secara umum adalah TF-IDF mengekstrak data berbentuk teks dari proses data *preparation* sebelumnya kedalam nilai biner berbentuk vektor untuk dapat diproses pada algoritma SVM, kemudian data numerik berbentuk vektor kemudian dibagi menjadi *training* dan *testing.* Tujuan data *taining* untuk melatih *machine* dan data *testing* untuk menguji keberhasilan *machine* dengan pengukuran validation *predictions* berdasarkan nlai *accuracy, precission, recall* dan *f1-score,* berikut merupakan langkah-langkah dan paparan hasil pengujian pada tahap kedua:

5.2 Import data dan Library

Langkah pertama adalah mempersiapkan data teks yang sudah melewati proses *data preparation* dan validasi, yang sudah benar-benar siap untuk dilakukan

pemrosesan lebih lanjut yaitu klasifikasi teks analisis sentimen menggunakan teknik *FastText* pada tahap *feature extraction* dengan memasukan beberapa *library* yang dibutuhkan, kemudian mengambil *Clean.csv* dalam file komputer lokal yang diberi nama *df* , terdapat dua kolom yaitu label dan tweet dengan hasil berikut:

Tabel 5. 1 Hasil *import data*

No	label	tweet
0	1	pasti anies punya prestasi
1	0	main aman gimana jauh pikir awam saya jelas ba...
2	0	ayoviralkan rekam jejak anies rasyid baswedan ...
3	1	suasana desa memang selalu senang udara segar ...
4	0	program anies hanya ubah nama rumah sakit se i...

5.3 *Term Frequency – Inverse Document Frequency* (TF – IDF)

sebuah teknik umum berguna dalam pengolahan teks untuk mengevaluasi berapakah penting kata tersebut dalam suatu dokumen relatif terhadap seluruh koleksi dokumen. TF - IDF merupakan gabungan dari dua *matrix*, *term frequency* (TF) dan *inverse document frequency* (IDF). Teknik ini dikembangakan oleh Gerard Salton dan rekannya, yang dikenal sebagai bapak pengambilan informasi modern. Gerard Salton adalah seorang ilmuwan komputer yang bekerja di bidang pengambilan informasi dan pencarian teks.

Langkah pertama dalam penerapan teknik TF-IDF ini adalah dengan menginport kelas terlebih dahulu, setelah menginport kelas dilakukan konversi kolom tweet menjadi tipe data berupa string agar sistem bisa mengenali data tersebut, setelah dilakukan perubahan data tersebut maka langkah selanjutnya mengubah data tweet menjadi vector menggunakan teknik TF-IDF, untuk membuat kelas dan mengubah data tweet menjadi data string agar dapat dibaca oleh sistem untuk mengekstraksi fitur dari data teks, seperti menghitung frekuensi kata, *term*

frequency inverse document document frequency (TF-IDF), kelas tersebut berfungsi untuk merubah kumpulan dokumen teks menjadi representasi fitur numerik. TF - IDF merupakan teknik yang berguna untuk evaluasi berapa penting sebuah kata dalam sebuah dokumen relatif terhadap kumpulan dokumen lainnya. Singkatnya, ini membantu menormalisasi frekuensi kata sehingga kata-kata yang umum tidak terlalu ditekankan dibandingkan kata-kata yang lebih jarang namun mungkin lebih signifikan, dengan hasil sebagai berikut :

```
  (0, 1687) 0.937325011732628
  (0, 104)  0.3484563421440753
  (1, 571)  0.31557894213586246
  (1, 498)  0.30202058600516063
  (1, 1181) 0.2915039079542727
  (1, 780)  0.2829111563231096
  (1, 2156) 0.2915039079542727
  (1, 1924) 0.26935280019240776
  (1, 195)  0.2285857714733914
  (1, 150)  0.3346883718179135
  (1, 1634) 0.27564608833651105
  (1, 670)  0.26380172664105855
  (1, 76)   0.30202058600516063
  (1, 1213) 0.27564608833651105
  (1, 104)  0.09806991552131511
  (2, 889)  0.37784630730102403
  (2, 493)  0.30206256911495316
  (2, 1640) 0.1587206040084363
  (2, 215)  0.2553615262995225
  (2, 1751) 0.421476266605111
  (2, 873)  0.3597382380978181
  (2, 1759) 0.38932248772312716
  (2, 163)  0.4469981567059919
  (2, 104)  0.1309787705746457
  (3, 1534) 0.20622308307048462
  :   :
  (889, 2215)     0.3979624485433575
  (889, 666)      0.25608497071010233
  (889, 1287)     0.23899197407115036
  (889, 1675)     0.14861053824122536
  (890, 1985)     0.6901247076286362
  (890, 888)      0.6901247076286362
  (890, 1675)     0.21784346637202331
  (891, 360)      0.22234899255586157
  (891, 855)      0.22234899255586157
  (891, 505)      0.22234899255586157
```

```
  (891, 341)      0.22234899255586157
  (891, 1410)     0.22234899255586157
  (891, 2068)     0.22234899255586157
  (891, 944)      0.44469798511172315
  (891, 1858)     0.22234899255586157
  (891, 545)      0.22234899255586157
  (891, 437)      0.187950989302612
  (891, 2100)     0.1936595523402048
  (891, 1312)     0.3578870902480314
  (891, 736)      0.2974529295451647
  (891, 898)      0.1535529860493624
  (891, 2128)     0.16897262763837115
  (891, 2164)     0.17525570976528537
  (891, 1675)     0.07018626452186194
  (891, 882)      0.17195682862428185
```

Gambar 5. 2 Hasil data tweet menjadi *vector*

Dalam Gambar 5.2 menggambarkan hasil dari representasi dari sebuah *sparse matrix* yang berisi nilai TF-IDF dari kata yang ada dalam dokumen, hasil ini menunjukkan pasangan indeks dan nilai dalam bentuk koordinat (dokumen, kata) diikuti oleh nilai TF-IDF yang sesuai. Ini adalah cara penyimpanan yang efisien untuk matriks yang sebagian besar elemennya adalah nol, karena hanya menyimpan elemen yang tidak nol.

Kemudian proses selanjutnya yaitu menampilkan data dari hasil represntasi *sparse matrix* yang digunakan untuk mendapatkan daftar fitur atau kata-kata yang diekstraksi atau dihasilkan oleh vektorisasi teks menggunakan TF-IDF, Kemudian hasil dari vektorisasi merupakan atribut atau hasil dari metode yang mengembalikan daftar fitur atau kata-kata yang diekstraksi dari teks. Kata-kata ini merupakan hasil tokenisasi teks awal, yang kemudian dijadikan fitur dalam representasi vektor TF-IDF , yang sesuai dengan indeks kolom dalam matriks TF-IDF atau yang dihasilkan dalam proses vektorisasi teks menggunakan TF-IDF. Dengan cara ini, dapat melihat kata-kata apa saja yang digunakan dalam analisis atau pemodelan berdasarkan TF-IDF.

```
array(['aamiin', 'ab', 'abang', ..., 'zulhas', 'zxs', 'zxy'],
dtype=object)
```

Gambar 5. 3 Hasil vektorisasi kata

Hasil dari Gambar 5.3 adalah sekumpulan elemen dengan tipe data string. Setiap elemen dalam array tersebut mewakili suatu kata, token, atau string tertentu, array tersebut adalah representasi data yang berisi sejumlah string, di mana setiap string merepresentasikan entitas teks tertentu.

Langkah ketiga atau langkah terakhir yang dilakukan setelah memuat vektor kata TF-IDF yaitu menggunakan matrix. maka akan menghasilkan metrix TF-IDF seperti berikut:

```
matrix([[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., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

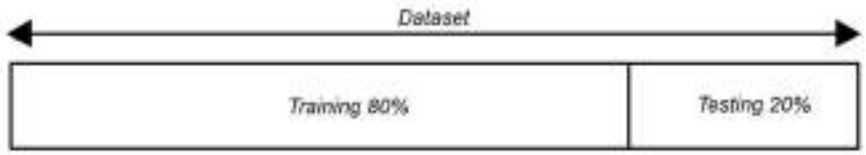
Gambar 5. 4 Hasil matrix TF-IDF

Dari hasil matrix TF-IDF pada Gambar 5.4 menggambarkan bahwa sebagian besar elemen dalam matriks adalah nol. Ini umum dalam matriks TF-IDF karena banyak kata dalam korpus mungkin tidak muncul di sebagian besar dokumen, menghasilkan banyak entri nol, matriks tersebut adalah hasil dari operasi numerik atau pembelajaran mesin, melalui perhitungan TF-IDF.

5.3 *Data Partition*

Langkah kedua adalah membagi data yang sudah bersih dan berlabel tersebut menjadi dua bagian berupa data *training* dan data *testing*, pada pembagian ini peneliti menggunakan aturan umum (*Rule of Thumb*) dalam persentase yaitu 80%

sebagai data *training* dan 20% digunakan sebagai data *testing* dengan ilustrasi pada gambar dibawah:

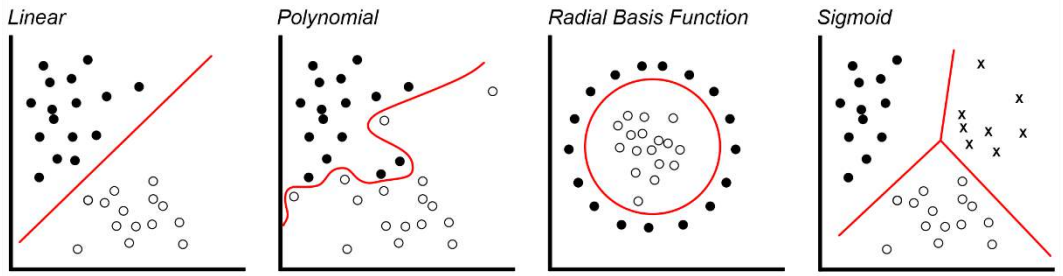


Gambar 5. 5 Pembagian *training data* dan *testing data*

Training data bagian dataset yang digunakan untuk melatih dan membuat prediksi untuk menjalankan fungsi dari sebuah algoritma klaisikasi serta sebagai sumber data mesin yang kita latih untuk mecari korelasi pembelajaran pada pola data tersebut, sedangkan *testing* merupakan bagian dataset yang digunakan untuk menguji dan melihat keakuratan mesin yang sudah melakukan proses pembelajaran dan pengenalan pola yang dilakukan pada data *training*, proses pembagian data '*tweet*' dan kemudian disimpan kedalam variabel X, kemudian untuk data '*label*' disimpan kedalam variabel Y. Dengan hasil akhir pembagian yang didapatkan yaitu data *training* sebanyak 713 dan data *testing* sebanyak 178 dengan perbandingan 80% dan 20%.

5.5 *Support Vector Machine* (SVM)

Setelah data melewati proses *splitting* dan proses *feature extraction*, saat ini data menjadi bentuk numerik berupa vektor lalu pada tahap inilah dilakukan proses modeling klasifikasi dengan menerapkan metode SVM. Algoritma pembelajaran ini dikembangkan oleh Boser, Guyon dan Vapnik pada tahun 1992 merupakan sistem pembelajaran dengan menggunakan hipotesis berupa fungsi – fungsi linier dalam sebuah fitur yang memiliki dimensi tinggi dan dilatih menggunakan teori yang optimal.



Gambar 5. 6 Algoritma *Support Vector Machine* (SVM)

Pada dasarnya konsep dan cara kerja dari algoritma *Support Vector Machine* (SVM) adalah berusaha mencari dan menemukan fungsi garis pemisah (*hyperplane*) yang terbaik diantara beberapa fungsi. Seperti pada Gambar 5.6 terdapat 4 fungsi atau karnel dengan tipe data dan pola masing-masing dan masing-masing memiliki jenis garis pemisah yang berbeda-beda pada setiap fungsinya. Metode *Support Vector Machine* (SVM) mencari dan memilih dari salah-satu garis/*hyperplane* yang paling baik dan pemilihan tersebut sangat bergantung pada jenis data dengan tujuan dapat memisahkan antar kelas yang berbeda dengan sempurna. Formulasi algoritma ini dapat dibedakan menjadi beberapa fungsi atau karnel yang digunakan yaitu *linear*, *polynomial, radial basis function* dan *sigmoid* dengan cara mencaru dan mengoptimalkan pemrosesan yaitu proses *tunning*. Langkah awal adalah *tuning Support Vector Machine* (SVM) dalam *machine learning* adalah proses untuk mengoptimalkan parameter-parameter model SVM agar memberikan kinerja yang terbaik untuk tugas yang sedang dihadapi. Berikut adalah langkah-langkah dalam proses tuning SVM :

a. Pemilihan karnel, memilih jenis kernel function yang akan digunakan untuk menentukan cara SVM akan memetakan data ke dalam fitur yang tinggi serta yang digunakan *linear*, *polynomial*, *radial basis function* (RBF) dan *sigmoid*. Pemilihan kernel sangat berpengaruh besar pada hasil model didapat.

b. Penentuan parameter C, dalam mengatur *trade-off* antara margin yang lebih besar dan kesalahan klasifikasi yang kecil. Nilai C yang lebih besar akan memberikan margin yang kecil tetapi kesalahan klasifikasi yang lebih kecil, sementara nilai C yang lebih kecil akan memberikan margin yang lebih besar tetapi kesalahan klasifikasi yang lebih besar. Ini adalah langkah penting dalam tuning SVM, dengan metode *GridSearch* untuk menemukan nilai terbai
c. Penentuan parameter gamma, Parameter ini mengontrol fleksibilitas model terhadap data pelatihan. Dengan berpatokan pada hasil dengan gambaran nilai gamma yang besar dapat menghasilkan model yang rumit, sementara nilai gamma yang kecil dapat menghasilkan model yang sederhana. Sama seperti parameter C, serta dapat mencoba berbagai nilai gamma untuk menemukan hasil yang optimal.

Berikut merupakan hasil tuning yang dilakukan *machine* terhadap data yang telah diproses dengan berbagai parameter yang telah di masukan sebagai pilihan untuk mencari model SVM yang paling optimal:

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
[CV 1/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.615 total time=   0.1s
[CV 2/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.608 total time=   0.1s
[CV 3/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.608 total time=   0.1s
[CV 4/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.613 total time=   0.1s
[CV 5/5] END ........C=0.1, gamma=1, kernel=rbf;, score=0.613 total time=   0.1s
[CV 1/5] END .C=1000, gamma=0.0001, kernel=poly;, score=0.615 total time=   0.1s
[CV 2/5] END .C=1000, gamma=0.0001, kernel=poly;, score=0.608 total time=   0.1s
[CV 3/5] END .C=1000, gamma=0.0001, kernel=poly;, score=0.608 total time=   0.1s
[CV 4/5] END .C=1000, gamma=0.0001, kernel=poly;, score=0.613 total time=   0.1s
[CV 5/5] END .C=1000, gamma=0.0001, kernel=poly;, score=0.613 total time=   0.1s
```

Gambar 5. 7 Hasil *tunning* SVM

Pada penerapan model SVM yang digunakan dan telah dilakukan tunning dengan hasil pemrosesan *machine* seperti pada Gamba 5.7 telah didapatkan hasil yang paling optimal pada nilai C=10, gamma=1 dan karnel=polynomial. Dengan hasil output berikut:

```
{'C': 10, 'gamma': 1, 'kernel': 'poly'}
```

Gambar 5. 8 *Output best tunning*

Setelah memperoleh hasil *tuniing* untuk model yang telah ditetapkan yaitu *Support Vector Machine* (SVM), langkah berikutnya adalah menampilkan hasil nilai visual berupa grafik berbentuk tabel dengan melibatkan pemahaman visual yang mendalam terhadap model yang telah disusun. Ini mencakup analisis nilai C, evaluasi hasil nilai gamma, dan pemahaman terperinci terkait jenis kernel yang digunakan pada parameter ini dapat memahami dampaknya terhadap kinerja keseluruhan model *Support Vector Machine* (SVM). Pada dasarnya, *Support Vector Machine* (SVM) adalah suatu *linear classifier*, namun SVM dapat dikembangkan lagi menjadi *non linear classifier*. terdapat fungsi kernel *trick* pada SVM yang sudah didapatkan pada proses *tunning* sebelumnya adalah karnel *polynomial* dengan persamaan berikut:

$$K\,(x_i, x_j) = ((x_i, x_j)+1)^d$$

Dengan:

K = Fungsi karnel

x_i, x_j = Fektor dari dataset

d = Pangkat *polynomial*

Kemudian untuk langkah-langkah persamaan secara umum menggunakan algoritma SVM sebagai berikut:

1. Merepresentasikan kata - kata dalam bentuk vektor numerik berdasarkan distribusi kata-kata cuitan twitter dalam konteks korpus teks *GloVe*;
2. Inisialisai beberapa parameter yang diperlukan dalam perhitungan manual menggunakan SVM seperti a_i, y, C, s, λ, dan $i_{max;}$

Keterangan:

a_I = *Alpha* untuk mencari *support vector*

γ = *Learning rate* berfungsi sebagai pengontrol kecepatan

C = *Cost* yang berfungsi sebagai minimalisir nilai *error* saat *training*

s = *Epsilon* yang berfungsi untuk mengukur tingkat *error* klasifikasi

λ = merupakan turunan batas teoritis

i_{max} = untuk iterasi maksimum

3. Dapat melakukan perhitungan kernel *polynomial* lebih dahulu
4. Dapat melakukan perhitungan kedalam nilai y atau label dari kelas yang telah ditentukan dimana kelas positif (1) dan kelas negatif(0)
5. Perhitungan matriks *Hessian* dengan menggunakan persamaan sebagai berikut:

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2)$$

nilai i dan $j = 1, 2, 3, \ldots, n$

Keterangan:

D = matriks *Hessian*

y_i = kelas data ke - i

y_j = kelas data ke - j

(x_i, x_j) = fungsi kernel yang digunakan

6. Menghitung nilai *error rate* dengan tujuan untuk mencerminkan seberapa sering model membuat kesalahan dalam memprediksi kelas atau nilai target.

$$E_{i=} \sum_{j=1}^{n} a_i D_{ij}$$

Keterangan:

E_i = nilai *error rate* data ke - i

7. Menghitung delta alpha untuk mengetahui perubahan fungsi yang disebut dengan *Lagrange Multiplier*. Jika data *training* telah mencapai nilai konvergen ($max(|\delta a_i|) < \varepsilon$) dan ketika maksimum iterasi mencapai nilai yang ditentukan, maka iterasi akan berhenti;

   $\delta a_i = mi(max[\gamma(1 - E_i), a_i], C - a_i)$

   Keterangan:

   δa_I = nilai delta alfa data ke-*i*

8. Menghitung nilai a_i baru, koefisien yang baru dihitung sebagai bagian dari vektor bobot (*w*) yang menggambarkan *hyperplane* pemisah agar dapat digunakan pada iterasi selanjutnya;

   $a_i = a_i + \delta a_I$

9. Menghitung nilai $w.\ x^+$ dan $w.\ x^-$ untuk mendapatkan nilai bias *b* atau parameter yang bertanggung jawab untuk menentukan posisi *hyperplane*;

   $w.\ x^+ = a_i y_i(x_i, x^+)$

   $w.\ x^- = a_i y_i(x_i, x^-)$

   $b = -\frac{1}{2}(w.\ x^+ + w.\ x^-)$

   Keterangan:

   $w\ .\ x^+$ = nilai kernel data *x* dengan data *x* kelas positif

   $w\ .\ x^-$ = nilai kernel data *x* dengan data *x* kelas negatif

   b = nilai bias

10. Selanjutnya jika sudah didaptkan nilai bias, maka langkah selanjutnya menghitung nilai bobot dan kernel data *testing* supaya dapat di inputkan ke dalam fungsi klasifikasi data *testing* untuk menentukan data uji

$$f(x) = sign \sum_{i=0}^{n} (a_i y_i K(x, x_i) + b)$$

Keterangan:

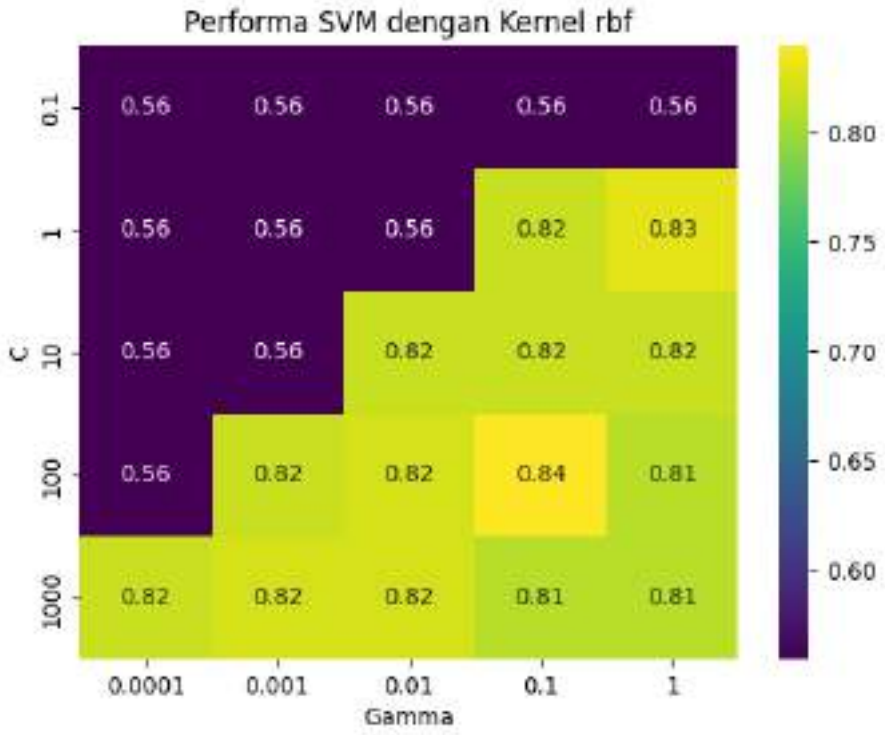
w = parameter *hyperplane* yang dicari garis tegak lurus antara garis *hyperplane* dan titik *support vector*

x = titik data masuk SVM

a_I = adalah nilai bias

(x, x_i) = fungsi kernel

Dengan berbagai kombinasi parameter C, gamma, dan jenis kernel. Melalui *nested loops*, *source code* ini menginisialisasi dan melatih model SVM pada setiap kombinasi parameter yang telah ditentukan menggunakan SVC dari library *sklearn*. Performa model dievaluasi dan disimpan dalam matriks performa. Kemudian hasilnya disajikan sebagai serangkaian *heatmap* menggunakan *seaborn* dan *matplotlib*, menampilkan hasil dari performa SVM untuk setiap jenis kernel berdasarkan variasi parameter C dan gamma. Untuk membantu analisis visual performa model terhadap kombinasi parameter yang berbeda, berikut visualisasi performa dari pemodelan yang telah didapatkan dengan menggunakan karnel *rbf* terhadap nilai-nilai dari parameter C, dan juga nilai-nilai dari parameret *Gamma* dengan hasil visualisasi seperti gambar:



Gambar 5. 9 *Hasil visualisasi params*

Dari visualisasi hasil *tunning* data teks cuitan dari media sosial Twitter pada Gambar 5.9 dapat dibaca untuk hasil terbaik dengan berpatokan pada akurasi tertinggi yaitu yang ditepati pada kolom ketiga untuk C dengan nilai 100 dan baris kelima untuk gamma dengan nilai 1 kemudian dengan hasil yang paling tinggi didapat 0.84. setelah mendapatkan nilai-nilai tersebut, selanjutnya adalah melakukan penggambaran untuk visualisasi karnel yang telah didapat yaitu karnel *rbf* dengan tampilan persebaran data serta dapat melihat pemisah antar data yang didapat atau *hyperplane.* Dengan menggunakan analisis PCA (*Principal Component Analysis*) yang mereduksi dimensi fitur ke 2D, dikarenakan hasil dari *feature extraxtion* yang dihasilkan oleh TF-IDF adalah berdimensi 256D. Setelah reduksi, model *Support Vector Machine* (SVM) dilatih dengan parameter terbaik dari *GridSearchCV* menggunakan data yang sudah direduksi. Selanjutnya, kode memvisualisasikan hyperplane dari model SVM pada data 2D, menampilkan garis

keputusan yang dibuat oleh model untuk memisahkan kelas, dengan warna dan pola garis yang merepresentasikan batas keputusan yang dihasilkan oleh model terhadap kelas-kelas pada data yang direduksi ke 2 dimensi. Untuk data dengan dimensi *n*, hyperplane dapat didefinisikan sebagai berikut :

$w \cdot x + b = 0$

Dimana :

- *w* adalah vektor bobot
- x adalah vektor fitur
- *b* adalah bias

Untuk menemukan hyperplane optimal, kita perlu memaksimalkan margin sambil memastikan bahwa semua titik data terklasifikasi dengan benar. Ini dapat diformulasikan sebagai masalah optimisasi, kemudian bisa diminimalkan dengan persamaan menggunakan *Hard* margin SVM dan *Soft* Margin SVM :

1. Hard Margin SVM, Untuk kasus dimana data sepenuhnya terpisahkan,

$$\frac{1}{2}||w||^2$$

Dengan kendala :

$$y_i(w \cdot x_i + b) \geq 1, \forall_i$$

Dimana :

- y_i adalah label label kelas untuk data x_i yaitu $y_i \in \{-1,1\}$.

2. *Soft* Margin SVM, ata tidak sepenuhnya terpisahkan, kita memperkenalkan variabel slack ε_i untuk memungkinkan beberapa pelanggaran margin :

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\varepsilon_i$$

Dengan kendala :

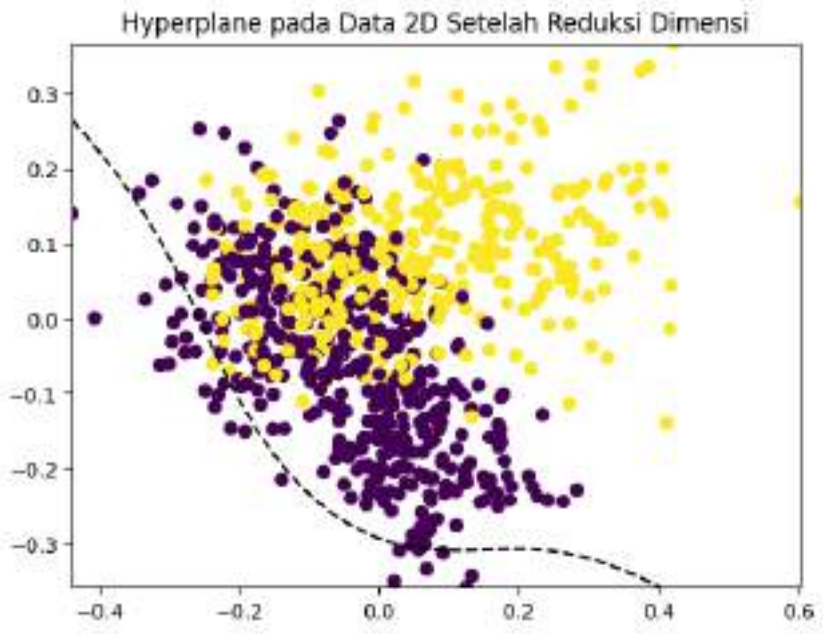
$$y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0, \quad \forall_i$$

C adalah parameter yang berfungsi sebagai pengontrol *trade-off* antara memaksimalkan margin dan meminimalkan kesalahan klasifikasi. Kemudian jika data yang tidak linier dapat menggunakan fungsi kernel untuk memetakan data kedalam ruang dimensi yang tinggi dengan persamaan :

$$K(x_{i,}x_j) = \emptyset(x_i) \cdot \emptyset(x_j)$$

Dari gambar 5. 17 menggunakan rumus *Radial Basis Function* (RBF) dengan menggunakan persamaan $K(x_{i,}x_i) = \exp(-\gamma||x_i - x_j||^2)$

Berikut adalah hasil visualsasi dengan menggunakan karnel *polynomial* dataset yang digunakan mempunyai karakter data yang tidak terstruktur karnel ini cocok untuk memecahkan masalah klasifikasi pada dataset pelatihan yang ternomanilasis sesuai dengan namanya, pada klasifikasi model ini adalah dengan mencari garis pemisah antar kelas dengan hasil berikut:



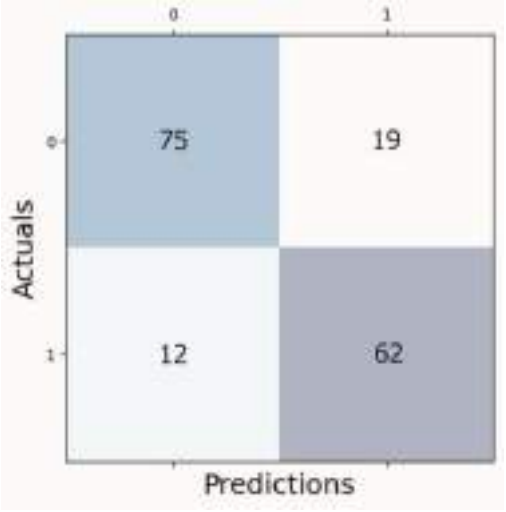
Gambar 5. 10 Hasil *hyperplane*

Dapat dijelaskan pada Gambar 5.10 dalam pengklasifikasian dan penarikan garis *hyperplane* memliki dua kelas yang diwakili dalam kelas ungu adalah kelas positif dan kelas kuning adalah kelas negatif yang dipisahkan oleh garis *polynomial* berwarna hitam (*hyperplane*), kemudian area antar garis putus-putus adalah *margin* yang diperoleh berdasarkan jarak terdekat antara *hyperlane* dengan kelas yang ingin dipisahkan dan setiap kelas yang berperan sebagai penentu *margin* dikenal dengan istilah *support vector*. Dalam menentukan *hyperplane* metode *Support Vector Machine* (SVM) akan memilih margin yang paling besar *maximum margin* sebagai hasil akhir dengan hasil yang cukup baik untuk perpisahan kedua data tersebut.

5.6 *Validation*

Pada langkah terahir proses klasifikasi analisis sentimen ini adalah melakukan pengukuran dan evaluasi model menggunakan *predictions*. *predictions* merupakan table yang berfungsi dalam pemahaman performa model klasifikasi pada *machine learning* dengan cara kerjanya menggambarkan jumlah prediksi *machine* yang benar terhadap data *real* dan jumlah prediksi *machine* yang salah terhadap dara *real* yang dibuat oleh model terhadap data uji, memungkinkan untuk mengevaluasi sejauh mana model kita efektif dalam mengklasifikasikan data berdasarkan komponen utama yaitu TP (*true positife)*, TN (*true negativ)*, FN (*false negativ*) dan FP (*false positiv*). Dari empat komponen tersebut, bisa menghitung metriks evaluasi model seperti akurasi, presisi, recall dan nilai *F1 - Score*, dimana memberikan pandangan yang lebih mendalam tentang kinerja model klasifikasi dalam mengatasi kasus analisis sentimen menggunakan proses *machine learning* pada studi kasus sentimen positif dan negatif. Untuk menghasilkan dan menampilkan *predictions* dari hasil kerja model *machine learning* terhadap data uji. *predictions* ini

memberikan gambaran visual seberapa baik algoritma dapat mengklasifikasikan data tersebut menjadi kategori yang benar dan seberapa sering terjadi kesalahan prediksi, yang digunakan sebagai acuan atau untuk menentukan hasil dari perhitungan *accuracy, precission, recall* dan *f1 – score* dengan hasil *predictions* sebagai berikut:



Gambar 5. 11 Hasil tabel *predictions*

Hasil tabel *predictions* yang didapatkan pada Gambar 5.11 dapat dijelaskan berdasarkan empat komponen utama yang didapatkan pada masing-masing kolom dengan penjelasan yaitu:

- TP (*true positive*), sebanyak 62 data yang positif terklasifikasi benar
- TN (*true negative*), sebanyak 75 data yang negatif terklasifikasi benar
- FP (*false positive*), yaitu sebanyak 19 data positif terklasifikasi salah
- FN (*false Negative*), yaitu sebanyak 12 data negatif terklasifikasi salah

Dari informasi tabel *predictions* yang sudah didapat, selanjutnya adalah melakukan perhitungan *accuracy, precission, recall* dan *f1 – score,* digunakan mengukur berapa jauh keberhasilan model dalam mengklasifikasikan data.

Accuracy (A) persentase prediksi yang benar dari *true positive* dan *true negative.*

$$A = \frac{(TP+TN)}{(TP+FP+FN+TN)} = \%$$

$$A = \frac{(62+75)}{(62+12+19+75)} = 0.8155$$

$$Accuracy = 81\%$$

Precission (P) persentase prediksi benar dari dari seluruh nilai *positive*

$$P = \frac{(TP)}{(TP+FP)} = \%$$

$$P = \frac{(62)}{(62+19)} = 0.7654$$

$$Precission = 76\%$$

Recall (R) persentase prediksi *positif* dibandingan dengan *true positif*.

$$R = \frac{(TP)}{(TP+FN)} = \%$$

$$R = \frac{(62)}{(62+12)} = 0.8378$$

$$Recall = 83\%$$

F1-Score (F) merupakan pembandingan rata - rata *precission* dan *recall*.

$$F = \frac{2(P\ x\ R)}{P+R} = \%$$

$$F = \frac{2(0.7654\ \ x\ 0.8378)}{0.7654+\ 0.8378} = 0.7998$$

$$F1 - Score = 79\%$$

BAB VI

PEMBAHASAN

Pada bab ini membahas hasil pengujian dari kedua model yang telah dilakukan, peneliti akan fokus pada presentasi dan analisis faktor yang mempengaruhi pemrorsesan dari hasil uji yang telah dilakukan terhadap kedua metode yang dibandingkan yaitu pemrosesan data dengan metode *Naive Bayes* dan SVM untuk memprediksi data berbentuk teks dari cuitan Twitter.

6.1 Hasil Pengujian

Untuk mengevaluasi hasil proses penerapan teknik pada *feature extraction* yang tepat agar mendapatkan hasil yang paling optimal serta akurat dari berbagai perbandingan yang diterapkan yaitu dengan melakukan pengukuran dengan menggunakan model *confusion matrix* sebagai acuan penentuan nilai *accuracy, precission, recall* dan *f1-score* yang telah dilakukan sebelumnya dengan mendapatkan perbandingan hasil kedua teknik sebagai berikut berikut:

Tabel 6. 1 Perbandingan hasil *confusion matrix*

Confusion Matrix	TF-IDF + Naïve Bayes	TF-IDF + SVM
TP (*true positive*)	52	62
TN (*true negative*)	81	75
FP (*false positive*)	22	19
FN (*false negative*)	13	12

Dari perbandingan hasil *confusion matrix* yang telah didapatkan pada Tabel 6.1 dapat dijelaskan secara terperinci untuk perolehan hasil nilai yang didapat dari kedua algoritma pada masing-masing parameter untuk menentukan dan mengetahui jenis keunggulan pada setiap teknik yang telah dilakukan pengujian. Pendekatan ini memberikan landasan yang kuat untuk memahami kinerja dan potensi setiap algoritma yang di ujicobakan dengan penjelasan sebagai berikut:

- TP (*True Positive*): *Machine* dapat memprediksi bahwa sentimen tersebut positif dan memang benar dibuktikan dalam data aslinya tersebut adalah sentimen positif, dengan perolehan hasil untuk percobaan algoritma *Support Vector machine* (SVM) yang lebih unggul dengan selisih delapan angka yaitu 62 berbanding 52. Dengan catatan semakin besar nilai TP maka semakin akurat *machine* dalam prediksi data.
- TN (*True Negative*): *Machine* dapat memprediksi bahwa sentimen tersebut negatif dan memang benar dibuktikan secara *real* atau dalam data aslinya sentimen tersebut adalah sentimen negatif, dengan perolehan hasil untuk percobaan *Support Vector machine* (SVM) lagi yang lebih unggul dengan selisih 75 berbanding 81.
- FP (*False Positive*): *Machine* memprediksi bahwa sentimen tersebut positif dan ternyata prediksi salah sentimen tersebut adalah sentimen negatif, dengan perolehan hasil untuk percobaan *Support Vector machine* (SVM) lagi yang lebih unggul dengan selisih dua angka yaitu 19 berbanding 22. Dengan catatan semakin kecil nilai FP maka semakin sedikit *machine* dalam melakukan kesalahan untuk hasil klasifikasi. Dalam keadaan ini disebut dengan kesalahan (*Type Eror 1*).
- FN (*False Negative*): *Machine* memprediksi bahwa sentimen tersebut negatif dan ternyata prediksi salah, sentimen tersebut adalah sentimen positif, dengan perolehan hasil untuk percobaan teknik TF-IDF menggunakan metode *SVM* yang lebih unggul dibandingan dengan teknik TF-IDF + *Naïve Bayes* dengan selisih yaitu TF-IDF+*Naïve Bayes* 13 berbanding TF-IDF+SVM 12.

Dengan catatan semakin kecil nilai FN maka semakin sedikit *machine* dalam melakukan kesalahan klasifikasi. Dalam keadaan ini disebut dengan kesalahan (*Type Eror 2*).

Seperti yang telah dijelaskan dengan detail terkait hasil dari *confusion matrix,* dari beberapa hasil yang telah dipaparkan bahwa FN merupakan kesalahan tipe 2 (*Type Eror 2*) dimana kesalahan ini sangat berbahaya dibandingkan dengan kesalahan tipe 1 (*Type Eror 1*) pada kasus ini karena jika *machine* melakukan klasifikasi teks untuk sentimen analisis memberikan label negatif dan pada nyatanya kalimat tersebut adalah kalimat positif maka akan timbul kesalahan yang fatal dengan melalaikan tujuan utama yaitu *menjadikan* suatu program yang berguna bagi maslahat serta untuk menghindari prilaku atau kejadian prasangka buruk pada hasil penentuan label keputusan pada setiap komentar.Dengan hasil penerapan teknik TF-IDF dengan SVM menjadi model yang memiliki nilai terbaik untuk hasil *confusion matrix* dalam studi kasus analisis sentiment dalam memprediksi data dari beberapa instrument yang ada pada table *confusion matrix,* dibandingkan dengan hasil teknik TF-IDF dengan algoritma *Naïve Bayes*.

Dari hasil dan kesimpulan *confusion matrix* yang telah didapatkan, selanjutnya nilai-nilai tersebut digunakan untuk menghitung berapakah *performance matrix* guna untuk mengetahui kinerja dari model yang telah dibuat. Pada bagian ini dapat dipahami beberapa *performance matrix* yang digunakan yaitu *accuracy*, *precission*, *recall* dan *f1-score* yang telah dilakukan sebelumnya dengan mendapatkan perbandingan hasil kedua teknik sebagai berikut berikut:

Tabel 6. 2 Perbandingan hasil *performance matrix*

Parameter	TF-IDF + Naïve Bayes	TF-IDF + SVM
Accuracy	79%	81%
precission	70%	76%
Recall	80%	83%
f1-score	74%	79%

Dari perbandingan hasil *performance matrix* yang telah didapatkan pada Tabel 6.2 dapat dijelaskan secara terperinci untuk perolehan nilai dari kedua metode pada masing-masing parameter untuk menentukan dan mengetahui jenis keunggulan pada setiap teknik yang telah dilakukan pengujian. Pendekatan ini merupakan hasil akhir yang memberikan hasil kinerja *machine* secara mnyeluruh:

- *Accuracy* menggambarkan berapa akuratkah model bisa mengklasifikasi data dengan benar, maka *accuracy* merupakan rasio prediksi benar (sentimen positif dan sentimen negatif) dengan keseluruhan data, *accuracy* merupakan sebuah tingkat kedekatan terhadap nilai prediksi menggunakan nilai yang actual atau yang sebenarnya. Dari nilai *accuracy* dapat diperoleh dengan hasil yang lebih tinggi pada teknik TF-IDF+SVM dengan perbedaan dari teknin TF-IDF+*Naïve Bayes* yakni 81% berbanding dengan 79%.
- Nilai *precision* dapat diperoleh masih dengan hasil yang lebih tinggi pada teknik TF-IDF+SVM menghasilkan 76%, sedangkan TF-IDF+*Naïve Bayes* menghasilkan 70%.
- *Recall* menggambarkan seberapa berhasilkah untuk menemukan kembali sebuah informasi yang ada pada klisifikasi yang telah dilakukan pemrosesan untuk analisis sentimen. Kemudian pada dasarnya *recall* adalah rasio prediksi benar positif yang dibandingkan dengan keseluruhan sebuah data yang benar – benar

positif. Nilai *recall* tersebut mendapatkan hasil yang lebih tinggi lagi pada teknik TF-IDF menggunakan metode SVM dengan perbedaan tipis hanya tiga angka yakni 83% berbanding dengan 80%.

- *F1 - score* adalah kombinasi dari *precision* dan *recall* yang memberikan sebuah gambaran keseluruhan tentang kinerja model tersebut. Nilai *f1-score* yang tinggi menunjukkan keseimbangan yang baik antara ketepatan dan keberhasilan model dalam menemukan informasi. Dalam kasus ini *f1-score* juga masih menunjukkan hasil yang lebih tinggi pada teknik TF-IDF+SVM menghasilkan 79%, dengan perbandingan angka dari teknik TF-IDF+*Naïve Bayes* yakni 74%. *F1-score* berguna untuk mengevaluasi performa model secara keseluruhan, terutama ketika keseimbangan antara *precision* dan *recall* sangat penting.

Berdasarkan hasil paparan pengujian performa model pada prediksi data dengan data teks berasal dari cuitan Twitter menggunakan teknik TF-IDF dan *Naïve Bayes* dibandingkan dengan TF-IDF dan SVM, dengan semua perbandingan hasil yang telah didapatkan dan dibahas sebelumnya dapat disimpulkan bahwa teknik tersebut yang disandingkan dengan algoritma SVM menunjukkan kinerja yang unggul dari semua teknik yang digunakan dalam penilaian yaitu *confusion matrix* dan *performance matrix*. Meskipun perbedaan hasil nilai antara kedua metode tersebut tidak terlalu signifikan namun nilai *accuracy, precission, recall* dan *f1 - score* menggunakan teknik TF - IDF menggunakan metode SVM selalu lebih bagus dibandingkan TF - IDF dengan menggunakan metode *Naive Bayes*. Bahwa hal ini menandakan kombinasi antara metode vektorisasi teks dengan TF-IDF dan model SVM mampu memberikan hasil klasifikasi dalam hal memprediksi data lebih akurat dan seimbang dalam mengenali sentimen positif dan negatif pada data dalam

bentuk teks yang bersumber dari media social cuitan *Twitter*. Oleh karena itu, untuk tugas analisis sentiment terkait prediksi data pada studi kasius ini teknik TF-IDF dan SVM dapat dianggap sebagai pilihan sangat optimal dan unggul dalam mendapatkan hasil yang lebih baik serta akurat dibandingkan dengan penggunaan model TF-IDF dengan *Naïve Bayes*.

6.2 Hasil Evaluasi

Beberapa faktor yang dapat mempengaruhi kinerja dari penerapan *feature extraction* pada proses analisis sentiment untuk memprediksi data, seperti yang di lakukan dengan metode *Naïve Bayes* dan SVM, melibatkan pemilihan teknik *dalam* memprediksi data, sumber data dan karakteristik dataset. Berikut beberapa faktor utama:

- Ukuran dan Kualitas Dataset:

  Ukuran dataset, jumlah data yang cukup besar dapat membantu model untuk memahami variasi dan kompleksitas dalam bahasa. Dataset yang kecil dapat menyebabkan *overfitting*. Kualitas labeling, keakuratan label pada dataset sangat penting dan kesalahan dalam labeling menyebabkan model menghasilkan prediksi yang tidak akurat.

- Tuning Parameter:

  Parameter algoritma, pada metode SVM, tuning parameter C dapat mempengaruhi kinerja model serta pengaturan parameter yang tepat dapat meningkatkan kemampuan model untuk menangkap pola-pola dalam data.

- Preprocessing Teks:

  Pembersihan teks, proses pembersihan teks, seperti penghapusan tanda baca atau *stemming*, dapat memengaruhi representasi kata-kata dan kinerja model.

- Karakteristik Bahasa dan Domain:

  Spesifik domain, karakteristik teks dalam domain tertentu (misalnya, bahasa informal *Twitter*) dapat memerlukan penyesuaian khusus dalam memprediksi data untuk meningkatkan kinerja.

Pada proses evaluasi akhir adalah dengan menentukan dan menyimpulkan model yang terbaik yang sudah dilakukan dari perbandingan beserta dengan pengujian dan evaluasi hasil, didapatkan paduan antara teknik prediksi data menggunakan TF-IDF dan dipadukan menggunakan algoritma SVM memiliki hasil dan performa yang unggul dalam sengala aspek pengukuran. Dimana salah satu keunggulan metode SVM dapat menangani dataset yang kompleks dan tidak linier dengan baik, karena dapat membangun hyperplane yang optimal untuk memisahkan kelas-kelas, kemudian SVM juga memiliki kemampuan untuk menangani *overfitting* dengan baik, terutama ketika digunakan dengan parameter penalti C yang tepat. Kesimpulannya akhirnya adalah, pemodelan dengan teknik TF-IDF dan algoritma SVM memberikan solusi yang handal dalam analisis sentimen dalam hal memprediksi data teks dengan potensi untuk peningkatan kinerja dan generalitas.

Dari hasil penelitian yang sudah dijabarkan maka kita bisa mengambil kesimpulan bahwa jika mendapat informasi, kita harus mengecek kebenaran dari mana sumber informasi tersebut didapatkan, karena jika banyak pengguna yang membagikan informasi tanpa melakukan verifikasi, sehingga hoax dapat dengan mudah menyebar. Penyebaran informasi palsu atau hoaxs juga dijelaskan pada AL - Qur'an surat Al - hujarat ayat 6 Allah berfirman

يَا أَيُّهَا الَّذِينَ آمَنُوا إِنْ جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَنْ تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْبِحُوا عَلَى مَا فَعَلْتُمْ نَادِمِينَ

Artinya: “Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti, agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu.”

Dari tafsir surat Al – hujarat tersebut dapat dipahami dari nilai keislaman dalam hal bersosialisai secara langsung maupun yang berasal dari social media kita dianjurkan untuk, yang pertama kita harus mengecek terlebih dahulu kebenaran dari berita atau data tersebut, yang kedua kita diharapkan mengetahui siapa atau dari mana informasi tersebut bersal aau siapa yang menyebarkanya, yang ketiga kita harus bisa mensikapinya jika mendapatkan informasi dan yang terakhir atau yang ke empat kita harus mengetahui dampak dari penyebaran informasi tersebut bagaimana, karena setiap informasi harus jelas dan benar.

BAB VII

PENUTUP

7.1 Kesimpulan

Berdasarkan hasil dari analisis yang telah dilakukan dapat disimpulkan bahwa model pada teknik TF-IDF menggunakan metode SVM menunjukkan kinerja yang lebih unggul. Secara khusus algoritma SVM memiliki tingkat akurasinya sebesar 81%, sedangkan algoritma *Naive Bayes* hanya mencapai 79%. Lebih lanjut, *precision* pada model SVM mencapai 76%, sedangkan pada model *Naïve Bayes* hanya sebesar 70%. Meskipun *recall* model SVM hanya sedikit lebih tinggi daripada *Naïve bayes* yaitu 83%: 80%, namun nilai *F1-score* yang mencapai 79% pada model SVM menunjukkan keseimbangan yang baik antara ketepatan dan keberhasilan dalam menemukan informasi, sedangkan model *Naïve bayes* hanya mencapai 74%.

Kemudian dari hasil evaluasi penerapan kedua model yang telah dilakukan terdapat beberpa permasalahan yang ditemukan dalam penerapanya anatara lain. Pertama, ukuran korpus teks yang besar dapat memperlambat waktu *processing* karena memerlukan sumber daya komputasi yang signifikan untuk melatih model, penanganannya dengan melibatkan penggunaan *subset* data atau penyesuaian parameter untuk mempercepat proses pelatihan. Kedua, masalah ketidak seimbangan kelas dalam data sentimen dapat menyebabkan performa yang tidak optimal atau *machine* akan cendrung mengarah pada jumlah kelas data yang tinggi. Ketiga, metode SVM dapat sensitif terhadap skala fitur dan parameter yang detail, sehingga perlu dilakukan normalisasi atau proses *tunning* data untuk mendapatkan hasil optimal.

7.2 Saran

Ada beberapa saran yang dapat berikan oleh peneliti untuk penelitian berikutnya, saran-saran ini bertujuan untuk memperdalam dan meningkatkan efektivitas analisis sentimen dalam memprediksi data, guna menangani beberapa tantangan yang mungkin muncul dalam implementasi data teks Twitter kedepannya:

1. Penggunaan metode *Support Vector Machine* (SVM) umumnya memberikan prediksi yang akurat, tetapi sulit untuk menginterpretasikan alasan di balik prediksi tersebut. Penelitian berikutnya untuk dapat menggunakan teknik *cross-validation* untuk memastikan keandalan hasil prediksi SVM, atau melibatkan ahli domain untuk menguji kebenaran prediksi pada situasi nyata.
2. Penelitian mendatang dapat mengeksplorasi lebih lanjut metode untuk menangani ketidak seimbangan kelas secara efektif, mengeksplorasi bagaimana integrasi SVM dengan teknik lain dapat meningkatkan kinerja dan *interpretasi* model.

DAFTAR PUSTAKA

Arsi, P., & Waluyo, R. (2021). Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma Support Vector Machine (SVM). *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *8*(1), 147–156. https://doi.org/10.25126/jtiik.202183944

Cindo, M., & Rini, D. P. (2019). Metode Klasifikasi Pada Sentimen Analisis. *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*, *1*(1), 66–70. https://seminar-id.com/semnas-sainteks2019.html

Dikih Arif Wibowo, B. (2020). Analisis Sentimen Tweet Berbahasa Indonesia Tentang Vaksin Covid-19 Menggunakan Fasttext Embedding Dan Support Vector Machine. *Universitas Gadjah Mada*. http://etd.repository.ugm.ac.id/

Duei Putri, D., Nama, G. F., & Sulistiono, W. E. (2022). Analisis Sentimen Kinerja Dewan Perwakilan Rakyat (DPR) Pada Twitter Menggunakan Metode Naive Bayes Classifier. *Jurnal Informatika Dan Teknik Elektro Terapan*, *10*(1). https://doi.org/10.23960/jitet.v10i1.2262

Faiq, M., Putro, A., & Setiawan, E. B. (2022). Analisis Sentimen Terhadap Kebijakan Pemerintah dengan Feature Expansion Metode GloVe pada Media sosial Twitter. *E-Proceeding of Engineering*, *9*(1), 54–66.

Fluorida Fibrianda, M., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(9), 3112–3123. http://j-ptiik.ub.ac.id

Fransiska, S., & Irham Gufroni, A. (2020). Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method. *Scientific Journal of Informatics*, *7*(2), 2407–7658. http://journal.unnes.ac.id/nju/index.php/sji

Giovani, A. P., Ardiansyah, A., Haryanti, T., Kurniawati, L., & Gata, W. (2020). Analisis Sentimen Aplikasi Ruang Guru Di Twitter Menggunakan Algoritma Klasifikasi. *Jurnal Teknoinfo*, *14*(2), 115. https://doi.org/10.33365/jti.v14i2.679

Hikmawan, S., Pardamean, A., Nur Khasanah, S., Mandiri, N., Damai No, J., Jati Barat, W., & Selatan, J. (2020). Sentimen Analisis Publik Terhadap Joko Widodo Terhadap Wabah Covid 19 Menggunakan Metode Machine Learning.

Jurnal Kajian Ilmiah (JKI), *20*(2), 167–176. http://ejurnal.ubharajaya.ac.id/index.php/JKI

Indriani, A. (2020). Analisa Perbandingan Metode Naïve Bayes Classifier Dan K-Nearest Neighbor Terhadap Klasifikasi Data. *Sebatik*.

Jagdale, R. S., Shirsat, V. S., & Deshmukh, S. N. (2019). Sentiment analysis on product reviews using machine learning techniques. *Advances in Intelligent Systems and Computing*, *768*, 639–647. https://doi.org/10.1007/978-981-13-0617-4_61

Maryanto, B. (2017). Big Data Dan Pemanfaatannya Dalam Berbagai Sektor. *Media Informatika*, *16*(2).

Muara Sains, J., Ilmu Kesehatan, dan, Trisari Harsanti Putri, W., & Hendrowati, R. (2018). Penggalian Teks Dengan Model Bag Of Words Terhadap Data Twitter. *Jurnal Muara Sains, Teknologi, Kedokteran, Dan Ilmu Kesehatan*, *2*(1), 129.

Nurdin, A., Anggo, B., Aji, S., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal Teknokompak*, *14*(2), 74.

Octaviani, A., & Dewi, P. (2020). Big Data di Perpustakaan dengan Memanfaatkan Data Mining. *ANUVA*, *4*(2), 223–230.

Permatasari, P. A., Linawati, L., & Jasa, L. (2021). Survei Tentang Analisis Sentimen Pada Media Sosial. *Majalah Ilmiah Teknologi Elektro*, *20*(2), 177. https://doi.org/10.24843/mite.2021.v20i02.p01

Putri, I. E., Rahmawati, D., & Yufis Azhar, ; (2020). Comparison Of Data Mining Classification Methods To Detect Heart DiseaSE. *Jurnal PILAR Nusa Mandiri*, *16*(2), 213–218. https://doi.org/10.33480/pilar.v16i2.1481

Rohanah, A., Rianti, D. L., Sari, B. N., Informatika, T., & Karawang, U. S. (2021). Perbandingan Naïve Bayes Dan Support Vector Machine Untuk Klasifikasi Ulasan Pelanggan Indihome. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, *6*(1), 23–30.

Sabrila, T. S., Sari, V. R., & Minarno, A. E. (2021). Analisis Sentimen Pada Tweet Tentang Penanganan Covid-19 Menggunakan Word Embedding Pada Algoritma Support Vector Machine Dan K-Nearest Neighbor. *Fountain of Informatics Journal*, *6*(2), 69. https://doi.org/10.21111/fij.v6i2.5536

Sari, F. V., & Wibowo, A. (2019). Analisis Sentimen Pelanggan Toko Online Jd.Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi. *Jurnal SIMETRIS*, *10*(2).

Wahyono, T. (2018). *Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan* (Vol. 1). https://www.researchgate.net/publication/330441937

Wahyu Kurniawan, F., & Maharani, W. (2020). Analisis Sentimen Twitter Bahasa Indonesia dengan Word2Vec. *E-Proceeding of Engineering*, *7*(2), 7821–7828. https://code.google.com

Wibawa, A. P., Guntur, M., Purnama, A., Fathony Akbar, M., & Dwiyanto, F. A. (2018). Metode-metode Klasifikasi. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, *3*(1).