PERBANDINGAN METODE PEMBOBOTAN *TF-IDF* DENGAN *TF-RF* DALAM ANALISIS SENTIMEN KENAIKAN HARGA TIKET KONSER PASCA PANDEMI *COVID-19* DI INDONESIA BERBASIS ALGORITMA *NAÏVE BAYES*

SKRIPSI

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh
Gelar Sarjana Strata Satu (S.1)
Dalam Ilmu Teknologi Informasi

Oleh :

MUHAMMAD AMIRUL SYACHRUDIN

NIM : 1908096040

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

2024

PERNYATAAN KEASLIAN

Yang bertandatangan dibawah ini :

Nama : Muhammad Amirul Syachrudin

NIM : 1908096040

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul :

Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*

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

Semarang, 8 Mei 2024

Pembuat Pernyataan,

Muhammad Amirul S

NIM : 1908096040

KEMENTRIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl.Prof.Dr.Hamka Km.1 Semarang Telp. 024 76433366 Semarang 50185
E-mail: fst@walisongo.ac.id. Web : http://fst.walisongo.ac.id

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*

Penulis : **Muhammad Amirul Syachrudin**

NIM : 1908096040

Jurusan : Teknologi Informasi

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

Semarang, 24 Juni 2024

DEWAN PENGUJI

Penguji I,

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

Penguji II,

Dr. Khothibul Umam, S.T., M.Kom
NIP. 197908272011011007

Penguji III,

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

Penguji IV,

Mokhamad Iklil M., M.Kom
NIP. 198808072019031010

Pembimbing I,

Dr. Khothibul Umam, S.T., M.Kom
NIP. 197908272011011007

Pembimbing II,

Adzhal Arwani Mahfudh, M.Kom
NIP.199107032019031006

NOTA PEMBIMBING

Semarang, 8 Mei 2024

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

Assalamualaikum wr. wb.

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

Judul	:	Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*
Penulis	:	**Muhammad Amirul Syachrudin**
NIM	:	1908096040
Jurusan	:	Teknologi Informasi

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

Wassalamualaikum wr. wb.

Pembimbing I.

Dr. Khothibul Umam., S.T., M.Kom.
NIP. 197908272011011007

NOTA PEMBIMBING

Semarang, 8 Mei 2024

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

Assalamualaikum wr. wb.

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

Judul	:	Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*
Penulis	:	**Muhammad Amirul Syachrudin**
NIM	:	1908096040
Jurusan	:	Teknologi Informasi

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

Wassalamualaikum wr. wb

Pembimbing II.

Adzhal Arwani Mahfudh, M.Kom.
NIP.199107032019031006

LEMBAR PERSEMBAHAN

Alhamdulillahi Rabbil Aalamin, puji serta syukur saya panjatkan kepada *Allah SWT*. Terimakasih atas karunia-Nya yang telah memberikan kemudahan dan kelancaran sehingga skripsi ini dapat terselesaikan dengan baik.

Skripsi ini saya persembahkan untuk diri saya sendiri yang telah berjuang dan berusaha selama ini. Terimakasih atas kerja kerasnya untuk tetap teguh dan tidak menyerah meskipun telah diguncang oleh berbagai masalah. Mari kita tetap berdoa dan berusaha menjadi pribadi yang lebih baik serta jangan menyerah untuk kedepannya.

Halaman persembahan ini juga ditujukan sebagai ungkapan terima kasih kepada kedua orang tua dan keluarga saya yang telah mendoakan dan memberikan dukungan penuh selama perjuangan menempuh pendidikan hingga selesai. Semoga selalu diberikan kesehatan, kebahagiaan, dan panjang umur.

Terimakasih juga saya ucapkan kepada segenap dosen Prodi Teknologi Informasi, teman-teman Teknologi Informasi Angkatan 2019, dan teman-teman UKM Musik yang telah mewarnai kehidupan saya sebagai mahasiswa di UIN Walisongo Semarang.

Terimakasih banyak untuk semua pihak yang tidak dapat disebutkan satu persatu atas dukungan dan bantuannya kepada saya baik secara langsung maupun tidak langsung.

MOTTO

"Janganlah takut untuk menjadi sesuatu yang berbeda dari yang lainnya, karena sesungguhnya dunia ini dapat menjadi lebih indah karena adanya perbedaan."

ABSTRAK

Dalam era pasca pandemi *Covid-19*, industri musik di Indonesia merupakan salah satu sektor yang sangat terpengaruh oleh pandemi. Salah satu dampak yang muncul adalah kenaikan harga tiket konser yang signifikan dan menimbulkan berbagai kontroversi. Penelitian ini bertujuan untuk mengetahui pola sentimen pada data *tweet* pengguna media sosial *X* serta membandingkan keakuratan dari Metode *TF-IDF* dan *TF-RF* pada tahap ekstraksi fitur dalam analisis sentimen mengenai kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes* dengan menggunakan *dataset* yang berasal dari data *tweet* pengguna media sosial *X* sebanyak 1350 *tweet*. Alur penelitian ini dimulai dengan tahap persiapan data, *labelling, text-preprocessing*, ekstraksi fitur, pembagian *dataset*, dan perancangan model analisis sentimen berbasis Algoritma *Naïve Bayes* menggunakan 70% data latih dan 30% data uji yang berasal dari hasil pembobotan masing-masing metode pembobotan kata ke dalam bentuk klasifikasi sentimen positif dan negatif. Kemudian diakhiri dengan mengevaluasi model tersebut menggunakan *Confusion Matrix*. Hasil dari penelitian ini menunjukan bahwa masing-masing model memiliki performa yang sangat baik dan pola sentimen yang tercermin dalam data *tweet* yang diunggah di media sosial *X* memiliki kecenderungan yang negatif serta *dataset* yang diperoleh dari hasil pembobotan kata dengan menggunakan Metode *TF-RF* memiliki nilai akurasi yang lebih tinggi jika dibandingkan dengan Metode *TF-IDF*, yaitu sebesar 92,25% pada Metode *TF-RF* berbanding dengan 88,75% pada Metode T*F-IDF.*

Kata Kunci : analisis sentimen, tiket konser, *X, TF-IDF, TF-RF, Naïve Bayes, Confusion Matrix.*

KATA PENGANTAR

Assalamualaikum wr. wb.

Puji syukur peneliti ucapkan kepada *Allah SWT* atas segala nikmat dan karunia-Nya peneliti dapat menyelesaikan skripsi yang berjudul "Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*" sebagai syarat untuk mendapatkan gelar Sarjana (S.1) Program Studi Teknologi Informasi di UIN Walisongo Semarang.

Adapun penyusunan skripsi ini tidak selalu berjalan dengan mulus dan lancar sehingga banyak pihak yang terlibat dalam penyelesaiannya. Oleh karena itu, peneliti hendak menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Dekan Fakultas Sains dan Teknologi UIN Walisongo Semarang, Bapak Prof. Dr. H. Musahadi, M.Ag.
2. Ketua Program Studi Teknologi Informasi UIN Walisongo Semarang sekaligus Dosen Pembimbing I dan Dosen Wali, Bapak Dr. Khothibul Umam, S.T., M.Kom.
3. Dosen Pembimbing II, Bapak Adzhal Arwani Mahfudh, M.Kom.

4. Ibu Nurul Widiawatik, S.Pd. selaku Wali dan Guru Kelas 6B di SDI Al Madina Semarang yang telah membantu peneliti sebagai validator dalam penelitian ini.
5. Semua pihak yang tidak dapat peneliti sebutkan satu per satu yang telah terlibat dalam penyusunan skripsi ini sehingga dapat terselesaikan dengan baik.

Dalam penyusunan skripsi ini, peneliti menyadari bahwa penelitian ini masih terdapat berbagai kekurangan dan jauh dari kata sempurna. Oleh karena itu, peneliti mengharapkan adanya kritik dan saran yang bersifat membangun agar dapat menjadi bahan rujukan peneliti untuk penelitian selanjutnya. Peneliti berharap agar penelitian ini dapat bermanfaat dan memberikan dampak positif bagi semua pihak.

Semarang, 8 Mei 2024

Muhammad Amirul S
NIM : 1908096040

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR KODE PROGRAM

DAFTAR PERSAMAAN

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pandemi *Covid-19* telah dinyatakan berakhir menganut pada kebijakan yang telah diumumkan oleh Pemerintah Pusat yang secara resmi mencabut status pandemi Covid-19 di Indonesia pada tanggal 21 Juni 2023 (Humas Kemensetneg, 2023). Dalam era pasca pandemi *Covid-19*, industri hiburan, khususnya industri musik di Indonesia yang sebelumnya merupakan salah satu sektor yang sangat terpengaruh oleh pandemi telah mengalami transformasi yang signifikan. Diantaranya yaitu mulai gencarnya para seniman musik dan penyelenggara konser dalam mengadakan konser seolah hal tersebut dilakukan demi memuaskan hasrat para penikmat musik di Indonesia yang haus akan hiburan selama pandemi berlangsung (Tim Kreatif CNBC Indonesia, 2022).

Namun adanya pandemi *Covid-19* telah mengubah pola perilaku dan aktivitas masyarakat di Indonesia secara radikal dalam hal menghadiri acara hiburan, terutama konser musik. Salah satu dampak yang muncul dari perubahan yang terjadi adalah kenaikan harga tiket konser yang sangat signifikan jika dibandingkan dengan era sebelum pandemi *Covid-19* akibat semakin tingginya

inflasi, devaluasi mata uang, kenaikan biaya operasional untuk menggelar sebuah konser yang sangat drastis, hingga kenakalan promotor konser dalam mematok harga tiket konser demi keuntungan pribadi (Batubara, 2023).

Sebagai contoh, untuk membeli sebuah tiket konser *BTS* di situs penjualan tiket resmi penonton diharuskan membayar sebesar Rp11.000.000,00 (sebelas juta rupiah) dari harga awalnya yang hanya sebesar Rp3.000.000,00 (tiga juta rupiah) akibat adanya program "harga dinamis" yang diberlakukan oleh pihak promotor konser (Sandy, 2023). Dan untuk menikmati konser *Coldplay* yang baru-baru ini digelar di Jakarta pada 15 November 2023, penonton dikenakan Harga Tiket Masuk (HTM) mulai dari Rp800.000,00 (delapan ratus ribu rupiah) hingga Rp11.000.000,00 (sebelas juta rupiah) di situs penjualan tiket resmi, namun apabila penonton ingin membeli tiket melalui calo bahkan dapat melonjak dengan harga tertingginya mencapai Rp60.000.000,00 (enam puluh juta rupiah) untuk sebuah tiket konser (Tim DetikHot, 2023).

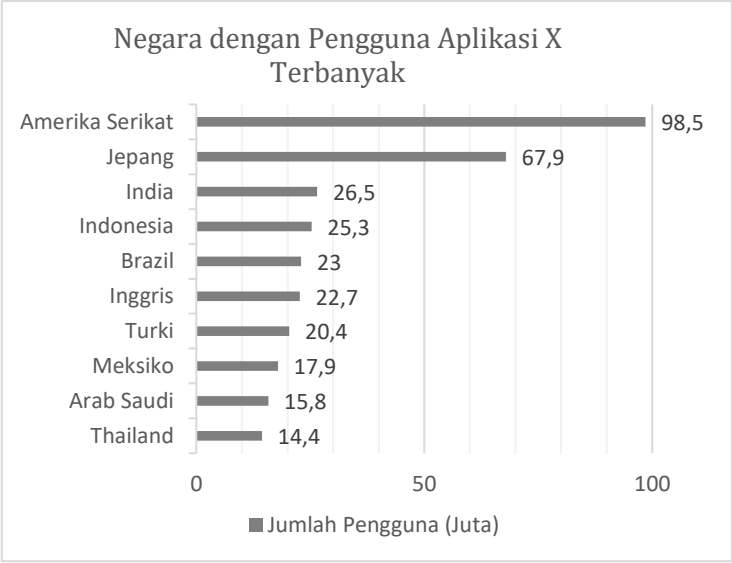
Adanya fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* yang signifikan dapat menimbulkan perubahan dalam pandangan masyarakat terhadap harga tiket konser, terutama yang tercermin dalam media sosial. Untuk dapat mengamati respon dari publik terhadap

fenomena tersebut, analisis sentimen melalui media sosial dapat menjadi solusi dalam memahami pandangan dan emosi masyarakat mengenai kenaikan harga tiket konser. Dalam penelitian ini, metode pembobotan kata seperti *Term Frequency-Inverse Document Frequency (TF-IDF)* dan *Term Frequency-Relevance Frequency (TF-RF)*, memainkan peran penting dalam mengekstraksi fitur yang terdapat di dalam kumpulan data dengan jumlah yang sangat besar, serta Algoritma *Naïve Bayes* dalam menilai sentimen yang tercermin dalam konten media sosial.

Metode *TF-IDF* merupakan metode ekstraksi fitur tanpa pengawasan *(Unsupervized Term Weighting)* yang telah digunakan dengan baik dalam berbagai penelitian dan pencarian informasi, namun tidak terlalu baik jika digunakan dalam klasifikasi teks dikarenakan tidak menggunakan informasi kelas atau label yang ada untuk menghasilkan bobot kata (Norindah Sari et al., 2023). Sedangkan Metode *TF-RF* merupakan metode pembobotan kata yang diawasi *(Supervized Term Weighting)* dan cukup baru diantara metode pembobotan kata yang lainnya dimana metode ini menggunakan informasi kelas dari *dataset* untuk menghitung nilai bobot (Norindah Sari et al., 2023).

Diantara berbagai *platform* media sosial yang beredar di internet, salah satu sosial media dimana para

penggunanya memiliki kebebasan dalam menuangkan opini mereka dan dapat digunakan dalam analisis sentimen adalah *X*. Menurut *DataReportal*, Indonesia sendiri telah menduduki peringkat ke-4 di seluruh dunia sebagai negara dengan pengguna aplikasi *X* terbanyak yaitu sebesar 25,3 juta pengguna (OBERLO, 2023).



Gambar 1.1 Negara dengan Pengguna *X* Terbanyak

Dalam beberapa tahun terakhir, *X* memiliki peran yang cukup besar dalam memberikan sumber informasi terkini karena banyak hal menarik yang dapat ditemukan pada *X*. Hal ini dapat terjadi dikarenakan banyaknya opini masyarakat yang tercurahkan baik terhadap suatu fenomena hingga mengenai suatu kebijakan pemerintah

(Hikmawan et al., 2020). Salah satunya yaitu fenomena kenaikan harga tiket konser musik pasca pandemi *Covid-19* yang dapat menjadi bahan pertimbangan dan rujukan untuk instansi terkait seperti Kementrian Pariwisata dan Ekonomi Kreatif (Kemenparekraf) ataupun pihak penyelenggara konser dalam menanggapi respons dari masyarakat.

Dalam perspektif *Unity of Sciences*, terdapat pemahaman bahwa pengetahuan dari berbagai disiplin ilmu dapat saling melengkapi dan membantu dalam memahami penciptaan Allah SWT. Dalam *Surah Al-Mujadilah* (58:11), Allah berfirman;

يَرْفَعِ اللّٰهُ الَّذِيْنَ اٰمَنُوْا مِنْكُمْۙ وَالَّذِيْنَ اُوْتُوا الْعِلْمَ دَرَجٰتٍۗ

Artinya: *Allah akan meninggikan derajat orang-orang yang beriman di antara kalian dan orang-orang yang diberi ilmu pengetahuan. (Q.S. Al-Mujadilah, 58:11)*

Ayat ini menegaskan mengenai pentingnya ilmu pengetahuan dan pemahaman, serta Allah SWT akan memberikan penghargaan kepada mereka yang berusaha mendapatkan pengetahuan. Dalam konteks kenaikan harga tiket konser pasca pandemi *Covid-19*, penyatuan atau kesatuan ilmu dari perspektif *Al-Qur'an* dapat diinterpretasikan sebagai upaya pemanfaatan berbagai disiplin ilmu seperti ilmu komputer dan ilmu sosial dalam

memahami dampak dari kenaikan harga tiket konser terhadap masyarakat.

Penelitian perbandingan antara metode pembobotan *TF-IDF* dan *TF-RF* dalam analisis sentimen terkait kenaikan harga tiket konser pasca *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes* ini sejalan dengan konsep kesatuan ilmu tersebut. Hal ini merupakan upaya peneliti untuk menggabungkan pendekatan dari berbagai disiplin ilmu guna memahami perubahan sentimen dan persepsi masyarakat yang tercermin melalui data *tweet* yang diunggah di media sosial *X*.

Adanya urgensi dari fenomena kenaikan harga tiket konser musik pasca pandemi *Covid-19* di Indonesia telah memunculkan berbagai perubahan dalam pandangan masyarakat sehingga menjadi suatu hal yang harus diketahui oleh instansi terkait seperti Kementrian Pariwisata dan Ekonomi Kreatif (Kemenparekraf), pihak penyelenggara konser, maupun publik. Melalui penelitian ini, diharapkan akan tercipta pemahaman yang lebih menyeluruh mengenai kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia, serta kontribusi terhadap pemahaman terkait peran konser musik dalam masyarakat pasca pandemi *Covid-19*. Perbandingan antara Metode Pembobotan Kata *TF-IDF* dan *TF-RF* berbasis Algoritma

Naïve Bayes diharapkan juga dapat memberikan wawasan yang lebih dalam mengenai keefektifan dan keakuratan dari penggunaan metode tertentu dalam menganalisis sentimen masyarakat melalui media sosial. Oleh karena itu, berdasarkan permasalahan tersebut mendorong peneliti untuk melakukan penelitian dengan mengangkat judul **"Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*".**

1.2 Identifikasi Masalah

Dari latar belakang masalah yang telah dipaparkan, dapat diidentifikasikan beberapa masalah yang muncul, diantaranya adalah :

1. **Pola Sentimen Masyarakat di Media Sosial *X***

   Bagaimana pola sentimen dan pandangan masyarakat di Indonesia mengenai kenaikan harga tiket konser ini dapat tercermin di dalam *platform* media sosial *X* ?

2. **Efektivitas dari Metode Pembobotan Kata dalam Analisis Sentimen**

   Pertanyaan kunci lainnya adalah apakah penggunaan dari Metode Pembobotan Kata *TF-IDF* dan *TF-RF*, secara efektif dapat mencerminkan dan menilai

sentimen masyarakat terkait kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes* ? Bagaimana perbandingan keakuratan dari kedua metode tersebut dalam menangkap sentimen yang bervariasi di dalam media sosial *X* ?

1.3 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan di atas, maka dapat diuraikan permasalahan sebagai berikut :

1. Bagaimanakah pola sentimen yang tercermin dalam data *tweet* yang diunggah di media sosial *X* mengenai kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia ?
2. Bagaimanakah keakuratan dari Metode Pembobotan Kata *TF-IDF* dan *TF-RF* pada tahap ekstraksi fitur dalam analisis sentimen kenaikan harga tiket konser pasca pandemi *Covid-19* di *platform* media sosial *X* berbasis Algoritma *Naïve Bayes* ?

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Untuk mengidentifikasi pola sentimen yang tercermin dalam data *tweet* yang diunggah di media sosial *X*

mengenai kenaikan harga tiket konser pasca pandemi *Covid-19* di kalangan masyarakat Indonesia.

2. Untuk membandingkan keakuratan dari Metode Pembobotan Kata *TF-IDF* dan *TF-RF* pada tahap ekstraksi fitur dalam analisis sentimen mengenai kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*.

1.5 Batasan Masalah

Agar penelitian dapat dilaksanakan secara ilmiah dan efektif, maka peneliti menerapkan beberapa batasan masalah yang diperlukan pada penelitian ini. Adapun batasan masalah yang diterapkan pada penelitian ini agar adalah sebagai berikut :

1. Data yang digunakan merupakan data *tweet* berbahasa Indonesia yang dikumpulkan dari media sosial *X* dengan kata kunci "harga tiket konser" sejumlah 1350 *tweet* yang di-*crawling* sejak tanggal 21 Juni 2023.
2. Tahap ekstraksi fitur dilakukan dengan membandingkan performa antara Metode Pembobotan Kata *TF-IDF* dan *TF-RF*.
3. Analisis sentimen dilakukan dengan memanfaatkan Algoritma *Naïve Bayes* dalam tahap perancangan model klasifikasi.

4. Hasil dari model klasifikasi akan dikelompokkan menjadi sentimen positif dan negatif.
5. Evaluasi model yang telah dirancang dilakukan dengan memanfaatkan *Confusion Matrix* untuk menghitung nilai akurasi, presisi, *recall,* dan *f1-score*.
6. Pengolahan data menggunakan bahasa pemrograman *Python* dan aplikasi *code compiler* yaitu *Google Colaboratory*.

1.6 Manfaat Penelitian

1. Manfaat Teoritis

Penelitian ini diharapkan dapat mengisi celah pengetahuan dalam bidang analisis sentimen, khususnya terkait efektivitas dari metode pembobotan kata dalam konteks sosial dan budaya pasca pandemi *Covid-19* di Indonesia. Selain itu, adanya perbandingan antara Metode Pembobotan Kata *TF-IDF* dan *TF-RF* diharapkan dapat menghasilkan pemahaman yang lebih dalam mengenai keterbatasan dan keunggulan masing-masing metode pembobotan kata dalam konteks analisis sentimen berbasis Algoritma *Naïve Bayes*.

2. Manfaat Praktis

Penelitian ini diharapkan dapat memberikan pandangan kepada lembaga maupun instansi yang bergerak di industri seni dan hiburan seperti Kementrian Pariwisata dan Ekonomi Kreatif (Kemenparekraf) maupun promotor konser terkait dampak yang ditimbulkan dari kenaikan harga tiket konser musik pasca pandemi *Covid-19* di Indonesia, serta dapat membantu lembaga dan instansi terkait untuk menyesuaikan strategi harga yang lebih sesuai dengan kebutuhan pasar. Selain itu, peneliti berharap hasil dari penelitian ini dapat digunakan sebagai dasar untuk merekomendasikan kebijakan publik terkait industri seni dan hiburan serta perlindungan konsumen, khususnya terkait regulasi harga tiket konser pasca pandemi *Covid-19* di Indonesia.

1.7 Sistematika Penelitian

Sistematika penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini merupakan pendahuluan yang memuat latar belakang, identifikasi masalah, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian dan sistematika penelitian.

BAB II KAJIAN TEORI DAN LANDASAN PUSTAKA

Bab ini memuat penelitian-penelitian terdahulu dan memuat penjelasan terkait landasan teori terkait penelitian ini seperti : Konser Musik, *X*, Analisis Sentimen, *Term Weighting,* Algoritma *Naïve Bayes*, dan lain-lain.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang gambaran proses penelitian dan metode yang dipakai dalam penelitian seperti diagram blok dan perencanaan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil penelitian yang dilakukan dan disertai dengan penjelasannya.

BAB V KESIMPULAN

Bab ini berisi kesimpulan berdasarkan dari hasil penelitian yang telah diperoleh dan beberapa saran penulis dalam pengembangan sistem.

BAB II
LANDASAN PUSTAKA

2.1 Kajian Penelitian yang Relevan

Agar penelitian ini dapat lebih terarah pada suatu masalah penelitian serta menghasilkan suatu kebaruan dalam penelitian, peneliti perlu melakukan kajian terhadap berbagai penelitian yang sebelumnya telah dilakukan untuk mencari keterkaitan antara penelititan tersebut dengan subjek penelitian ini. Berikut merupakan beberapa penelitian sebelumnya yang digunakan sebagai referensi dan acuan untuk menghindari terjadinya pengulangan hasil temuan pada penelitian ini.

Tabel 2.1 Kajian Penelitian yang Relevan

1.	**Judul Penelitian**	*Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional*
	Penulis, Tahun	(Darwis et al., 2021)
	Fokus dan tujuan penelitian	Penelitian ini bertujuan untuk mengetahui tingkat akurasi Algoritma *Naive Bayes* dalam analisis sentimen dengan topik BMKG. Jumlah *dataset* yang digunakan pada penelitian ini sebanyak 1179 data yang bersumber dari *Twitter.* Penelitian ini menggunakan Metode *TF-IDF* dalam tahap ekstraksi fitur dan Algoritma *Naive Bayes* dalam tahap klasifikasi. Hasil klasifikasi ke dalam

		bentuk sentimen positif, netral dan negatif dengan akurasi sebesar 69,97%.
	Keterkaitan penelitian	Penelitian ini menggunakan Metode *TF-IDF* pada tahap ekstraksi fitur dan Algoritma *Naïve Bayes* dalam tahap klasifikasi sehingga dapat dijadikan sebagai referensi mengenai pemanfaatan Metode *TF-IDF* dalam analisis sentimen.
2.	**Judul Penelitian**	*Analisa Sentimen Masyarakat Terhadap Kondisi New Normal Pasca Pembatasan Sosial Berskala Besar Akibat Covid 19 Menerapkan Metode Term Frequency-Relevan Frequency*
	Penulis, Tahun	(Harahap, 2022)
	Fokus dan tujuan penelitian	Penelitian ini bertujuan untuk mengetahui tingkat akurasi Metode *TF-RF* dalam analisis sentimen dengan topik *New Normal* pasca PSBB di Indonesia. Jumlah *dataset* yang digunakan pada penelitian ini sebanyak 120 data yang bersumber dari *Twitter*. Hasil klasifikasi ke dalam bentuk sentimen positif, netral dan negatif. Hasil penelitian menunjukan hasil yang cukup positif dengan presentasi positif sebesar 66,7%, netral sebesar 29,2%, dan negatif sebanyak 4,1%.
	Keterkaitan penelitian	Penelitian ini menggunakan Metode *TF-RF* pada tahap ekstraksi fitur sehingga dapat dijadikan sebagai referensi mengenai pemanfaatan Metode *TF-RF* dalam analisis sentimen.
	Judul Penelitian	*Perbandingan Metode Pembobotan TF-RF dan TF-IDF dikombinasikan dengan Weighted Tree Similarity untuk Sistem Rekomendasi Buku*

3.	**Penulis, Tahun**	(Sari et al., 2022)
	Fokus dan tujuan penelitian	Penelitian ini bertujuan untuk membandingkan Metode *TF-IDF* dan *TF-RF* dalam tahap ekstraksi fitur serta mengetahui tingkat akurasi dari Algoritma *Weighted Tree Similarity* dalam sistem rekomendasi buku. Jumlah *dataset* yang digunakan pada penelitian ini sebanyak 100 data yang bersumber dari data buku unit terpadu Universitas Lampung Mangkurat. Penelitian ini menggunakan Metode *TF-IDF* dan *TF-RF* dalam tahap ekstraksi fitur dan Algoritma *Weighted Tree Similarity* dalam tahap klasifikasi. Hasil klasifikasi dalam bentuk sistem rekomendasi berdasarkan nilai kemiripan *(similarity)* antara judul, pengarang, dan sinopsis dari data buku. Hasil akurasi yang diperoleh dengan menggunakan Metode *TF-RF* lebih tinggi jika dibandingkan dengan Metode *TF-IDF*, yaitu sebesar 98 % untuk *TF-RF* dan 96% untuk *TF-IDF*.
	Keterkaitan penelitian	Penelitian ini menggunakan Metode *TF-IDF* dan *TF-RF* pada tahap ekstraksi fitur sehingga dapat dijadikan sebagai referensi mengenai perbandingan performa antara Metode *TF-IDF* dan *TF-RF* dalam tahap ekstraksi fitur pada analisis sentimen..
	Judul Penelitian	*Fake News (Hoaxes) Detection on Twitter Social Media Content through Convolutional Neural Network (CNN) Method*
	Penulis, Tahun	(Tama & Sibaroni, 2023)

4.	**Fokus dan tujuan penelitian**	Penelitian ini bertujuan untuk mengetahui tingkat akurasi dari Algoritma *Convolutional Neural Network (CNN)* dalam deteksi berita palsu *(Hoax)* mengenai vaksin *Covid-19*. Jumlah *dataset* yang digunakan pada penelitian ini sebanyak 16.000 data yang bersumber dari *Twitter.* Penelitian ini menggunakan Metode *TF-IDF* dan *TF-RF* dalam tahap ekstraksi fitur dan Algoritma *Convolutional Neural Network (CNN)* dalam tahap klasifikasi. Hasil Klasifikasi ke dalam label *"hoax"* dan *"fact".* Berdasarkan hasil pengujian, diketahui bahwa pembobotan dengan menggunakan Metode *TF-RF* serta Algoritma *CNN* dalam tahap klasifikasi lebih baik dengan nilai akurasi sebesar 84,11% jika dibandingkan dengan Metode *TF-IDF* yaitu sebesar 80,29%
	Keterkaitan penelitian	Penelitian ini menggunakan Metode *TF-IDF* dan *TF-RF* pada tahap ekstraksi fitur sehingga dapat dijadikan sebagai referensi mengenai perbandingan performa antara Metode *TF-IDF* dan *TF-RF* dalam tahap ekstraksi fitur pada analisis sentimen.
5.	**Judul Penelitian**	*Analisis Pembobotan Kata pada Text Mining*
	Penulis, Tahun	(Deolika & Taufiq Luthfi, 2019).
	Fokus dan tujuan penelitian	Penelitian ini bertujuan untuk mengetahui tingkat akurasi dari Algoritma *Naïve Bayes* dalam ulasan terhadap Aplikasi LAPOR (Layanan Aspirasi dan Pengaduan *Online* Rakyat). Jumlah *dataset* yang digunakan pada penelitian ini sebanyak 100 data yang bersumber dari Aplikasi LAPOR

		(Layanan Aspirasi dan Pengaduan *Online* Rakyat). Penelitian ini menggunakan Metode *TF-IDF*, *TF-RF,* dan *WIDF* dalam tahap ekstraksi fitur dan Algoritma *Naive Bayes* dalam tahap klasifikasi. Hasil Klasifikasi ke dalam ke dalam 5 kategori yaitu kesehatan, pendidikan, PUPR, pertanian, dan Dukcapil. Berdasarkan hasil pengujian, diketahui bahwa pembobotan dengan menggunakan Metode *TF-RF* serta Algoritma *Naïve Bayes* dalam tahap klasifikasi lebih baik dari pembobotan *TF-IDF* dan *WIDF* dengan nilai *Accuracy* 98,67%, *Precision* 93,81%, dan *Recall* 96,67%.
	Keterkaitan penelitian	Penelitian ini menggunakan Metode *TF-IDF, TF-RF,* dan *WIDF* pada tahap ekstraksi fitur serta Algoritma *Naïve Bayes* pada tahap perancangan model klasifikasi sehingga dapat dijadikan sebagai referensi mengenai perbandingan performa antara Metode *TF-IDF* dan *TF-RF* dalam tahap ekstraksi fitur serta penggunaan Algoritma *Naïve Bayes* dalam analisis sentimen.

Berdasarkan hasil dari penelitian terdahulu yang telah disajikan pada tabel 2.1 tersebut, dapat diketahui bahwa penggunaan metode pembobotan kata pada tahap ekstraksi fitur dapat diaplikasikan ke dalam berbagai kasus dengan mengekstraksi dan memberikan nilai bobot pada setiap kata yang ada di dalam dokumen.

Sebagai pembeda dengan penelitian sebelumnya, pada penelitian ini peneliti akan membandingkan performa dan tingkat akurasi antara Metode Pembobotan Kata *TF-IDF*

(Term Frecuency – Inverse Document Frecuency) dan *TF-RF (Term Frequency - Relevance Frequency)* pada tahap ekstraksi fitur serta penggunaan Algoritma *Naïve Bayes* pada tahap perancangan model klasifikasi dalam analisis sentimen pengguna media sosial *X* mengenai fenomena kenaikan harga tiket konser musik pasca pandemi *Covid-19* di Indonesia. Dan untuk mengevaluasi hasil dari model analisis sentimen yang terdiri dari sentimen positif dan negatif, peneliti memanfaatkan *Confusion Matrix* pada tahap evaluasi model yang telah dirancang untuk mendapatkan nilai performa dari model yang telah dirancang.

2.2 Kajian Pustaka

2.2.1 Konser Musik

Kata konser berasal dari Bahasa Belanda dan Prancis yaitu *concerto* yang memiliki arti pertunjukan. Sedangkan dalam KBBI sendiri kata "konser" berarti pertunjukan musik di depan umum (BPPB Kemdikbudristek, 2016).

Pertunjukan musik merupakan salah satu bentuk komunikasi antara manusia dengan memanfaatkan media suara sebagai perantaranya (Hidayatullah, 2021). Pertunjukan musik sendiri telah ada sejak zaman dimana manusia mulai mengenal bentuk komunikasi sederhana. Pertunjukan musik

pada zaman sebelum era modernisasi seringkali dihubungkan dengan kegiatan keagamaan namun biasanya tetap bersifat santai agar dapat menarik minat para pendengarnya.

Dengan adanya kemajuan teknologi yang semakin pesat, basis penggemar yang telah memiliki berbagai akses, dan perkembangan ekonomi yang lebih baik telah menjadikan konser tidak hanya sebagai suatu pertunjukan yang menawarkan pengalaman menikmati musik secara langsung saja. Konser semakin berkembang sehingga dapat menjadi suatu pengalaman yang di dalamnya penuh dengan hal spontan dan tidak terduga (Hidayatullah, 2021). Saat ini menghadiri suatu konser musik merupakan suatu bentuk apresiasi dan penghormatan oleh para penikmat musik kepada musisi favorit atas karya-karya yang telah mereka ciptakan.

2.2.2 X

X merupakan *rebranding* dari salah satu media sosial yang sebelumnya bernama *Twitter* oleh pimpinan *X Corp* yaitu Elon Musk pada akhir Juli 2023. Sebelum rebranding menjadi X, *Twitter* merupakan suatu aplikasi berbasis layanan *microblogging* dimana para penggunanya dapat mengunggah pesan

menggunakan fitur *"tweet"* yang ada dibatasi hingga 240 karakter (Rosenberg et al., 2020). Selain itu, *Twitter* juga memungkinkan para penggunanya untuk dapat saling mengungkapkan pendapat dan perasaan mereka tentang berbagai isu dan fenomena yang sedang berlangsung di masyarakat dengan leluasa (Pravina et al., 2019).

Setelah dilakukan *rebranding* menjadi *X, Twitter* telah bertransformasi dari platform berbasis layanan *microblogging* menjadi *SuperApps* dimana penggunanya dapat memposting berbagai hal termasuk video dengan durasi yang lama, dan dalam beberapa bulan ke depan akan ditambahkan fitur berupa komunikasi secara komprehensif serta pengelolaan keuangan (Ivanova, 2023).

Namun tidak seperti media sosial lainnya yang mengharuskan penggunanya untuk menjalin hubungan pertemanan terlebih dahulu sebelum dapat berkomunikasi, *X* memungkinkan para penggunanya untuk untuk tetap terhubung meskipun mereka tidak berteman satu sama lain melalui fitur *"trending topic"* dimana di dalamnya memuat berbagai topik hangat yang sedang dibahas oleh seluruh pengguna *X* di seluruh dunia dalam kurun waktu tertentu.

2.2.3 Analisis Sentimen

Analisis sentimen adalah suatu tahapan yang digunakan untuk memahami informasi dalam bentuk tekstual, yang kemudian diekstraksi dan diolah secara otomatis untuk mendapatkan data opini yang terdapat dalam suatu dokumen teks (Rusydiana & Marlina, 2020). Dengan bantuan analisis sentimen dapat ditentukan kecenderungan suatu masyarakat dalam melihat suatu fenomena baik secara negatif maupun positif berdasarkan data yang telah diolah dalam sistem.

Secara umum analisis sentimen terdiri dalam lima tahapan, yaitu *Crawling Data, Pre-Processing, Feature Selection, Classification,* dan *Evaluation* (Salsabila, 2022). Analisis sentimen memiliki kemampuan untuk mengubah data yang tidak teratur menjadi data yang terstruktur. Keuntungan dari analisis sentimen yaitu dapat berfungsi sebagai salah satu sarana yang digunakan dalam menampung berbagai opini yang terdapat dalam masyarakat tanpa adanya rasa khawatir oleh suatu kebijakan ataupun peraturan yang merenggut kebebasan mereka dalam berpendapat. Analisis sentimen juga dapat digunakan untuk menganalisis peristiwa, pernyataan, dan komentar kontroversial. Hasil dari analisis sentimen

juga dapat memberikan wawasan serta dapat dijadikan sebagai bahan rujukan dan evaluasi bagi suatu perusahaan, tokoh masyarakat, hingga pemerintah dalam menentukan langkah selanjutnya.

2.2.4 Text-Preprocessing

Text-Preprocessing merupakan suatu tahap yang dilakukan untuk mengubah kumpulan data teks yang tidak teratur ke dalam format yang lebih mudah diolah dan dianalisis (Nuri, 2022). Sebelum melakukan pengolahan pada data berupa dokumen teks, baik untuk keperluan klasifikasi, mencari persamaan, maupun analisis sentimen maka data tersebut harus melalui tahapan *text-preprocessing* terlebih dahulu agar dapat diolah dengan lebih mudah dan rapi. *Text-prepocessing* sendiri merupakan tahapan yang sangat penting dalam teknik dan penerapan dari *text mining* karena semakin bersih *prepocessing* yang dilakukan, maka besar kemungkinan data yang dihasilkan akan semakin akurat.

Text-preprocessing sendiri terdiri dari beberapa tahapan, diantaranya yaitu (Rahman Isnain et al., 2021).

1. *Case Folding*

   Merupakan tahap mengubah kalimat yang ada di dalam data teks menjadi seragam.

2. *Cleaning*

   Merupakan tahap membersihkan data pada dokumen dan menyaring item yang tidak perlu seperti *html, emoticon, hashtag, mention* dan *url*.

3. *Filtering*

   Merupakan tahap penyaringan terhadap kata-kata tidak efektif yang ada di dalam dokumen.

4. *Tokenization*

   Merupakan tahap memilah dan memotong kata dalam kalimat dengan menggunakan pemisah seperti tanda koma (,), titik (.), mapupun tanda pemisah yang lain.

5. *Normalize*

   Merupakan tahap pengubahan kata-kata *slang*, singkatan dan bentuk kata tidak baku menjadi kata baku yang sesuai dengan ejaan dalam Kamus Besar Bahasa Indonesia (KBBI) (Nuri, 2022).

6. *Stemming*

   Merupakan tahap penyaringan kata yang terdapat kata sambung, kata ganti, dan kata depan yang akan diubah menjadi bentuk kata dasar

dengan menghilangkan awalan atau akhiran dari kata tersebut.

2.2.5 Term Weighting

Pembobotan Kata *(Term Weighting)* merupakan suatu metode yang digunakan untuk memberikan nilai bobot kepada setiap kata *(term)* yang terdapat dalam sebuah dokumen berdasarkan jumlah beserta tingkat kontribusinya untuk menentukan kelas atau kategorinya dalam dokumen (Deolika & Taufiq Luthfi, 2019). Metode *Term Weighting* dapat dibagi menjadi dua kategori berdasarkan kelas yang digunakan dalam dokumen, yaitu metode pembobotan kata tanpa pengawasan *(unsupervised)* dan metode pembobotan kata dalam pengawasan *(supervised)* (Carvalho & Guedes, 2020).

Metode pembobotan kata tanpa pengawasan *(unsupervised)* mengabaikan informasi dari suatu kelas untuk menghasilkan nilai bobot dan tidak dapat membedakan kelas yang sesuai untuk suatu *term* (Alshehri & Algarni, 2023). Dengan demikian, nilai bobot yang dihasilkan hanya dipengaruhi oleh frekuensi *term* yang terdapat pada dokumen sehingga metode ini tidak terlalu cocok apabila digunakan dalam tahap klasifikasi teks dan analisis sentimen. Sedangkan

pada metode pembobotan kata dalam pengawasan *(supervised),* menggunakan informasi dari suatu kelas untuk menghasilkan nilai bobot (Norindah Sari et al., 2023). Pada metode ini, informasi mengenai kategori yang terdapat di dalam data latih memiliki peranan yang penting terhadap tahap pelatihan data. Informasi tersebut dapat digunakan dengan berbagai metode untuk mengontrol tahap pembobotan kata dalam klasifikasi teks yang spesifik (Alshehri & Algarni, 2023).

Beberapa contoh dari metode pembobotan kata tanpa pengawasan *(unsupervised)* yang paling populer digunakan yaitu Metode *TF* dan *TF-IDF*. Sedangkan untuk metode pembobotan kata dalam pengawasan *(supervised)* terdapat beberapa metode yang cukup baru seperti Metode *Delta TF-IDF, TF-IDF-ICF* dan *TF-RF* (Norindah Sari et al., 2023).

2.2.6.1 TF-IDF

TF-IDF (Term Frequency - Inverse Document Frequency) adalah salah satu metode pembobotan kata tanpa pengawasan *(unsupervised term weighting)* yang sangat populer digunakan di dalam tahap ekstraksi fitur. Metode ini merupakan kombinasi antara Metode *TF* dan Metode *IDF.* Metode

TF-IDF digunakan dalam menentukan keterhubungan suatu kata *(term)* terhadap dokumen dengan memberikan bobot pada setiap kata. Pada Metode *TF*, terdapat anggapan bahwa kata dengan frekuensi kemunculan yang lebih tinggi akan memiliki nilai bobot yang lebih tinggi daripada kata dengan frekuensi kemunculan yang lebih rendah (Jiang et al., 2021).

$$W_{TF}(t) = \frac{\boldsymbol{jumlah\ kemunculan\ term\ (t)}}{\boldsymbol{jumlah\ kata\ dalam\ dokumen\ (d)}} \quad (2.1)$$

Persamaan 2.1 Metode *TF*

Persamaan 2.1 merupakan bentuk persamaan dasar dari Metode *TF (Term Frequency)* (Dwi Wanti, 2023). Dapat diketahui bahwa dalam Metode *TF,* nilai bobot *(W)* dari suatu kata *(t)* dapat diperoleh dengan melakukan pembagian antara jumlah kemunculan dari kata *(t)* dalam suatu dokumen dengan jumlah kata yang ada dalam dokumen *(d)*.

Metode *TF* memiliki kekurangan yaitu proses penghitungan yang dilakukan hanya bergantung dari jumlah kemunculan suatu kata yang terdapat di dalam dokumen. Jika suatu kata muncul pada seluruh dokumen yang ada di dalam korpus, maka akan dianggap sebagai kata umum *(common term)*

sehingga nilai bobot dari kata tersebut akan diabaikan (Dwi Wanti, 2023). Hal ini juga menyebabkan rendahnya kemampuan dari Metode *TF* untuk membedakan relevansi antar dokumen yang terdapat di dalam korpus.

Agar dapat mengatasi permasalahan tersebut, diperlukan Metode *IDF (Inverse Document Frecuency)* sebagai solusi untuk menghilangkan dominasi dari kata umum yang terdapat di dalam korpus (Dwi Wanti, 2023). Pada Metode *IDF,* terdapat asumsi bahwa suatu kata yang memiliki frekuensi kemunculan lebih sedikit di dalam dokumen akan dianggap lebih penting daripada frekuensi kemunculan kata yang lebih banyak (Jiang et al., 2021). Berikut merupakan bentuk persamaan dasar dari Metode *IDF* (Carvalho & Guedes, 2020).

$$W_{IDF}(t) = log\left(\frac{N}{DF_{(t)}}\right) \quad (2.2)$$

Persamaan 2.2 Metode *IDF*

Berdasarkan persamaan 2.2, dapat diketahui bahwa dalam Metode *IDF, N* adalah jumlah dari semua dokumen di dalam korpus, sedangkan *DF* adalah jumlah kemunculan kata *(t)* yang ada di dalam dokumen (Deolika & Taufiq Luthfi, 2019).

$$W_{(TF.IDF)} = TF_{(d,t)} \times IDF_{(t)} \quad (2.3)$$

Persamaan 2.3 *Metode TF-IDF*

Berdasarkan kedua gagasan tersebut, dapat disimpulkan bahwa Metode *TF-IDF* adalah bentuk perkalian dari Metode *TF* dan Metode *IDF*. Pada metode ini, nilai bobot akan semakin tinggi apabila dihasilkan oleh kata yang memiliki jumlah kemunculan yang lebih banyak dalam suatu dokumen, namun kata tersebut jarang muncul di dalam korpus. Namun metode ini memiliki kelemahan yaitu tahap pembobotan tidak dapat menghasilkan nilai bobot yang berbeda baik untuk kata negatif maupun positif sehingga tidak terlalu cocok apabila digunakan dalam klasifikasi teks (Norindah Sari et al., 2023).

2.2.6.2 TF-RF

TF-RF (Term Frequency - Relevance Frequency) adalah salah satu metode pembobotan kata dalam pengawasan *(supervised term weighting)* yang digunakan di dalam tahap ekstraksi fitur. Metode ini merupakan kombinasi dari Metode *TF* dengan Metode *RF* yang bertujuan agar dapat memperoleh tingkat performa yang lebih baik jika dibandingkan metode pembobotan kata sebelumnya (Ramadhan et

al., 2021). Sebagai contohnya yaitu pada Metode *TF-IDF*, nilai bobot akan ditentukan hanya berdasarkan frekuensi kemunculan dari kata yang terdapat di dalam suatu dokumen, namun kata tersebut jarang muncul dalam korpus. Sedangkan pada Metode *TF-RF*, nilai bobot dapat diperoleh dengan mempertimbangkan relevansi antar dokumen, yang dapat dilihat dari frekuensi kemunculan kata yang terdapat dalam kategori yang saling berhubungan (Al Ghofany et al., 2022). Selain itu, Metode *TF-RF* juga berfokus terhadap seluruh dokumen yang terdapat di dalam korpus baik di dalam dokumen mengandung kata tersebut ataupun tidak (Ramadhan et al., 2021).

Metode *TF-RF* dapat didefinisikan sebagai bentuk perkalian antara Metode *TF* dan Metode *RF* dengan mempertimbangkan relevansi data berdasarkan frekuensi munculnya kata *(term)* dalam kategori yang relevan (Joergensen Munthe et al., 2022). Berikut merupakan bentuk persamaan dari Metode *TF-RF* (Harahap, 2022).

$$W_{(TF.RF)} = TF_{(d,t)} \times log\left(2 + \frac{b}{max\ (1, c)}\right) \quad (2.4)$$

Persamaan 2.4 Bentuk Dasar Metode *TF-RF*

Berdasarkan persamaan 2.4, dapat diketahui bahwa dalam Metode *TF (Term Frequency),* nilai bobot *(W)* dari suatu kata *(t)* dapat diperoleh dengan melakukan pembagian antara jumlah kemunculan dari kata *(t)* dalam suatu dokumen dengan jumlah kata yang ada dalam dokumen *(d)*. Sedangkan pada Metode *RF (Relevance Frecuency),* relevansi dokumen dapat diperoleh berdasarkan frekuensi kemunculan kata di dalam kategori yang berkaitan dimana *b* adalah total dari data yang berisi kata *(t),* dan *c* adalah total dari data yang tidak berisi kata *(t).*

2.2.6 Klasifikasi

Klasifikasi merupakan sebuah tahap yang digunakan dalam menemukan suatu pola ataupun fungsi tertentu yang terdapat di dalam *dataset* yang berukuran relatif besar dan mengelompokkannya ke dalam beberapa kelas (Derajad Wijaya & Dwiasnati, 2020).

Dalam metode klasifikasi dilakukan perancangan model berdasarkan pemeriksaan karakteristik pada data latih yang telah ada sebelumnya, yang kemudian menggunakan model tersebut ke dalam data yang baru untuk dapat diklasifikasikan ke dalam beberapa kelas objek. Setiap

kemungkinan dari nilai yang dimiliki oleh atribut target menunjukkan kelas yang nantinya akan diprediksi berdasarkan nilai dari atribut prediktor (Derajad Wijaya & Dwiasnati, 2020).

Metode klasifikasi sendiri telah digunakan secara luas dalam mengelompokkan dan mengatur berbagai hal ke dalam suatu sistem tertentu yang telah dirancang dengan sedemikian rupa sehingga menjadi lebih mudah untuk dikenali, dipelajari, dan dipahami . Beberapa contoh metode klasifikasi yang sering digunakan diantaranya yaitu Algoritma *Naïve Bayes*, Algoritma *Support Vector Machine (SVM*), Algoritma *Decission Tree*, Algoritma *Fuzzy* dan Algoritma Jaringan Saraf Tiruan *(Artificial Neural Network)* (Wibawa et al., 2018).

2.2.7 Algoritma *Naïve Bayes*

Algoritma *Naïve Bayes* adalah salah satu metode klasifikasi yang dapat digunakan untuk membuat prediksi berbasis probabilistik sederhana berdasarkan Teorema *Bayes* (Yuniarti et al., 2020). Algoritma ini dapat mengasumsikan bahwa suatu atribut tertentu dalam sebuah kelas tidak secara langsung berkaitan dengan atribut yang lainnya meskipun masing-masing

atribut yang ada di dalam *class* tersebut saling ketergantungan satu sama lain (Uddin et al., 2019).

Algoritma *Naïve Bayes* sendiri merupakan metode klasifikasi yang cara kerja nya berdasarkan Teorema Bayes yang memiliki bentuk persamaan umum sebagai berikut (Wibawa et al., 2018).

$$P(A|B) = \frac{P(A)}{P(B)} \times P(B|A) \quad (2.5)$$

Persamaan 2.5 Bentuk Umum Teorema *Bayes*

Keterangan :

A	: Data dengan kelas yang belum diketahui
B	: Hipotesis pada data A yang merupakan suatu kelas spesifik
P(A|B)	: Nilai probabilitas pada Hipotesis A berdasarkan Kondisi B
P(A)	: Nilai probabilitas pada Hipotesis A
P(B)	: Nilai probabilitas B
P(B|A)	: Nilai probabilitas pada hipotesis B berdasarkan kondisi A

Dengan menggunakan bentuk persamaan tersebut, memungkinkan untuk dapat menilai data yang akan diklasifikasikan, yang selanjutnya dapat digunakan untuk mengolah data yang telah diperoleh.

Algoritma *Naïve Bayes* memiliki beberapa kelebihan, diantaranya yaitu relatif lebih mudah untuk dirancang dan dapat berfungsi dengan baik dalam klasifikasi *multiclass* (Arunadevi et al., 2018). Namun algoritma ini juga memiliki beberapa kelemahan,

diantaranya yaitu algoritma ini sangat peka terhadap jumlah fitur yang berlebihan sehingga menghasilkan nilai akurasi yang rendah, dan vektor fitur yang dihasilkan memiliki ukuran yang cukup besar sehingga diperlukan teknik atau metode tertentu untuk mengurangi ukuran vektor tersebut (Wibawa et al., 2018).

2.2.8 Confusion Matrix

Confusion Matrix merupakan suatu tahapan dalam konsep data mining dimana dilakukan perhitungan akurasi ke dalam bentuk tabel matriks yang mengklasifikasikan jumlah data uji baik yang benar maupun salah (Normawati & Prayogi, 2021). Berikut adalah bentuk tabel dari *Confusion Matrix* :

Tabel 2.2 Bentuk Tabel *Confusion Matrix*

		Nilai Aktual	
		Possitive	***Negative***
Nilai PrediKsi	***Possitive***	*True Possitive (TP)*	*False Possitive (FP)*
	Negative	*False Negative (FN)*	*True Negative (TN)*

Keterangan :

TP : jumlah data dari nilai aktual kelas positif dan nilai prediksi kelas positif

TN : jumlah data dari nilai aktual negatif dan nilai prediksi negatif

FP : jumlah data dari nilai aktual positif dan nilai prediksi negatif

FN : jumlah data dari nilai aktual negatif dan nilai prediksi positif

Setelah membuat Tabel *Confusion Matrix*, diperlukan perhitungan menggunakan Rumus *Confusion Matrix* dengan bentuk persamaan sebagai berikut (Rahman Isnain et al., 2021).

Tabel 2.3 Rumus *Confusion Matrix*

No	Jenis	Bentuk Persamaan
1.	*Accuracy*	$\frac{TP + TN}{TP + TN + FP + FN} \times 100$
2.	*Precision*	$\frac{TP}{TP + FP} \times 100$
3.	*Recall*	$\frac{TP}{TP + FN} \times 100$
4.	*F1-Score*	$2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$

Berdasarkan tabel diatas, *accuracy* (keakuratan) adalah jumlah rasio antara prediksi benar yang dibandingkan dengan keseluruhan data, *precision* (presisi) adalah jumlah rasio antara prediksi benar positif yang dibandingkan dengan keseluruhan hasil yang diprediksi positif, *recall* (sensitifitas) merupakan

jumlah rasio prediksi benar positif yang dibandingkan keseluruhan data yang benar positif, dan *F1-Score* merupakan hasil perbandingan rata-rata dari *precision* dan *recall* yang telah dibobotkan (Arthana, 2019).

2.2.9 Python

Python merupakan salah satu bahasa pemrograman tingkat tinggi yang sangat populer karena memiliki aturan sintaks yang mudah dipahami dan lebih mudah dibaca oleh pengguna jika dibandingkan dengan bahasa pemrograman yang lainnya (Raschka et al., 2020). Selain itu, *Python* mendukung multi paradigma pemrograman, diantaranya yaitu Pemrograman Berorientasi Objek (PBO), Pemrograman Imperatif, dan Pemrograman Fungsional (Syahrudin & Kurniawan, 2018).

Sebagai bahasa pemrograman yang dinamis, *Python* dilengkapi dengan kemampuan manajemen memori secara otomatis. Pada praktiknya, *python* digunakan sebagai bahasa *script* dan dapat digunakan dalam berbagai tujuan diantaranya yaitu pengembangan perangkat lunak serta dapat berjalan dalam berbagai platform sistem operasi seperti *Linux/Unix, Windows, Mac OS*, dan lain sebagainya dengan memanfaatkan beberapa *library* seperti

Pandas, Numpy, dan *Matplotlib* yang digunakan dalam membersihkan dan mengolah data sehingga dapat menghasilkan *output* sesuai dengan apa yang diinginkan (Nuri, 2022).

BAB III
METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan oleh peneliti dalam penelitian ini yaitu metodologi penelitian kuantitatif – eksperimen yang merupakan pendekatan dalam ilmiah menggunakan data kuantitatif (berupa angka atau statistik) untuk mengetahui pengaruh dari variabel tertentu dan mengukur dampaknya terhadap variabel yang lainnya (Anam et al., 2023). Penelitian ini dimulai dengan melakukan penghitungan serta pengolahan data ke dalam bentuk angka, data sampel yang telah diperoleh dengan tahap *crawling data* terhadap data *tweet* pengguna media sosial *X* mengenai fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia yang berbentuk kalimat akan melalui berbagai macam tahapan seperti persiapan data, *text-preprocessing*, dan ekstraksi fitur sehingga setiap kata yang terdapat di dalam dokumen akan mempunyai nilai. Penghitungan tersebut dilakukan dengan melakukan perbandingan antara Metode Pembobotan Kata *TF-IDF* dan *TF-RF* dalam model klasifikasi sentimen berbasis Algoritma *Naïve Bayes.*

Berdasarkan fungsinya, variabel penelitian yang digunakan oleh peneliti dalam penelitian ini dapat dibagi menjadi 3, diantaranya yaitu (Riyanto & Hatmawan, 2020):

a. Variabel bebas, merupakan variabel yang diketahui sebagai penyebab terjadinya perubahan pada variabel terikat. Penggunaan Metode Pembobotan Kata *TF-IDF* dan *TF-RF* pada tahap ekstraksi fitur merupakan variabel bebas yang akan mempengaruhi variabel terikat pada penelitian ini.
b. Variabel terikat merupakan variabel yang dipengaruhi oleh variabel bebas dan diukur selama eksperimen berlangsung. Hasil data analisis sentimen pengguna media sosial *X* merupakan variabel terikat yang akan dipengaruhi oleh variabel bebas pada penelitian ini.
c. Variabel antara, merupakan variabel yang secara tidak langsung mempengaruhi hubungan antara variabel bebas dan variabel terikat. Fenomena kenaikan harga tiket konser musik pasca pandemi *Covid-19* di Indonesia serta penggunaan Algoritma *Naïve Bayes* pada tahap perancangan model klasifikasi merupakan variabel antara yang akan menentukan pengaruh hubungan antara variabel bebas dan variabel terikat pada penelitian ini.

3.2 Metodologi Pengumpulan Data

3.2.1 Studi Literatur

Pada tahap studi literatur dilakukan dengan persiapan data sampel yang terbagi menjadi data primer dan data sekunder. Data primer diperoleh langsung dari media sosial *X* mengenai fenomena kenaikan harga tiket konser musik pasca pandemi *Covid-19* di Indonesia. Sedangkan data sekunder diperoleh berdasarkan referensi dari buku, situs, peramban dan penelitian terkait mengenai analisis sentimen, Metode *TF-IDF* dan *TF-RF,* serta Algoritma *Naïve Bayes.*

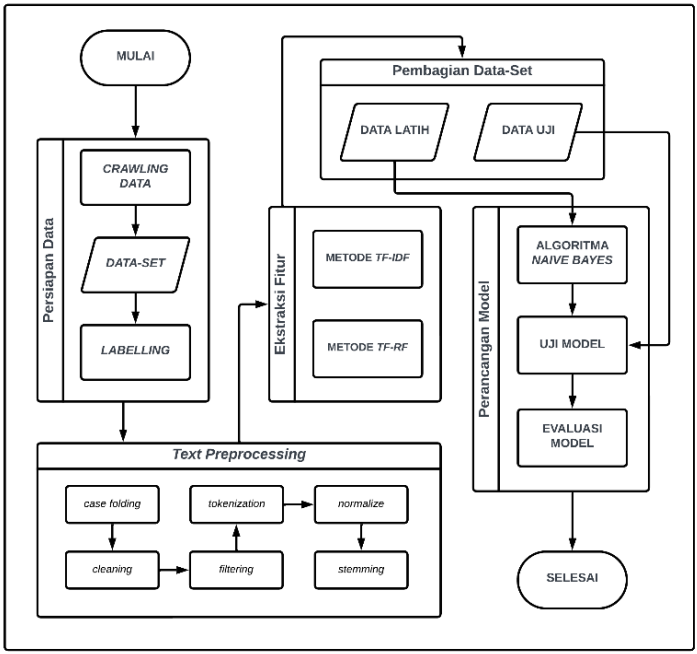
3.2.2 Studi Lapangan

Untuk memperkuat identifikasi masalah pada penelitian ini, peneliti melakukan studi lapangan berupa pengamatan secara langsung terhadap semua aktivitas di dalam media sosial dan peramban yang sedang *trending* mengenai fenomena kenaikan harga tiket konser musik pasca era pandemi *Covid-19* di Indonesia, khususnya pada media sosial *X*.

3.3 Alur Penelitian

Agar penelitian ini dapat dilakukan secara sistematis dan terstruktur, diperlukan perancangan alur kerja pada penelitian ini sehingga dapat memberikan gambaran

umum mengenai alur penelitian yang akan dilakukan oleh peneliti dari awal hingga akhir. Adapun gambar 3.1 merupakan gambaran dari alur penelitian ini (Assidyk et al., 2020).



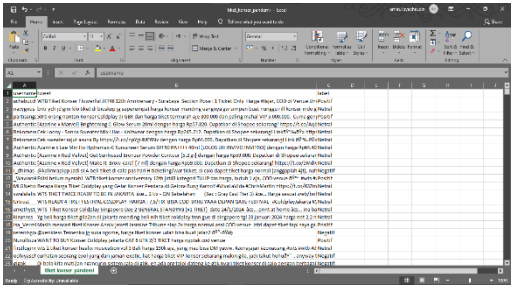
Gambar 3.1 *Flowchart* Alur Penelitian

3.3.1 Persiapan Data

Alur pada penelitian ini dimulai dengan identifikasi dan perumusan masalah melalui studi literatur dan studi lapangan. Setelah itu dilanjutkan dengan tahap persiapan data yang terdiri dari tahap *crawling data* dan *labelling*.

A. Crawling Data

Merupakan tahap pengambilan data dengan tujuan untuk mendapatkan informasi dari suatu basis data (Normawati & Prayogi, 2021). Setelah diperoleh identifikasi masalah yang sesuai dengan topik penelitian ini, peneliti melakukan pengumpulan data sampel berdasarkan kata kunci "harga tiket konser" dengan memanfaatkan *library Python* yang terhubung dengan media sosial *X* sejak tanggal 21 Juni 2023 menganut pada kebijakan yang telah diumumkan oleh Pemerintah Pusat yang secara resmi mencabut status pandemi *Covid-19* di Indonesia pada tanggal 21 Juni 2023 (Humas Kemensetneg, 2023). *Dataset* yang diperoleh hasil dari tahap *crawling data* nantinya akan disimpan dalam format *file CSV*. Gambar 3.2 merupakan contoh hasil dari pengumpulan data sampel dengan menggunakan *crawling data* yang disimpan dalam format *file CSV*.



Gambar 3.2 Contoh Hasil Tahap *Crawling Data*

B. Labelling

Setelah diperoleh data sampel yang disimpan dalam format *file CSV*, dilakukan tahap *labelling* dengan tabel 3.1 sebagai contoh hasilnya. Data yang telah dikumpulkan akan diberikan label sentimen positif dan negatif secara manual dengan bantuan pakar Bahasa Indonesia sebagai validator untuk menghindari terjadinya subjektivitas dan bias dalam penentuan label sentimen pada hasil data *tweet* yang telah diperoleh dari tahap sebelumnya (Zidan, 2022). Data hasil dari tahap *labelling* nantinya akan diolah dalam tahap *text-preprocessing*.

Tabel 3.1 Contoh Hasil Tahap *Labelling*

Data *Tweet*	Label Sentimen
lagian promotor skrng aja yg kek babi, nyari untung gede doang tp bad service sblm covid harga tiket konser msh normal begitu covid selesai anjir harga tiket naik 2x lipat itupun dpt kategori yg lebih jelek even di SG - MY yg kursnya lbh tinggi, tiketnya jauh lbh murah dr indo"	Negatif
Bismillah tawakal WTB // Want to Buy!! Tiket Coldplay Jakarta CAT 4,5,6 Harga Normal naik dikit gpp yg penting wajar. COD FULLPAYMENT VENUE PLISS t. wts wtb konser coldplay #ColdplayJakarta	Positif

3.3.2 Text-Preprocessing

Data yang telah diberikan label sentimen pada tahap *labelling* merupakan data kotor karena terdiri dari berbagai item yang tidak diperlukan dalam analisis sentimen sehingga perlu diolah agar menjadi lebih rapi dan bersih dalam tahap *Text-Preprocessing* yang terdiri dari 6 tahapan yaitu *case folding, cleaning, tokenization, normalize, stopwords removal,* dan *stemming.*

A. Case Folding

Data hasil tahap *labelling* yang masih terdiri dari berbagai kalimat dengan huruf kapital akan diubah menjadi huruf kecil *(lower case)* dengan tahap *case folding* agar seluruh kalimat yang terdapat di dalam dokumen menjadi seragam seperti yang ditunjukkan pada tabel 3.2.

Tabel 3.2 Contoh Hasil Tahap *Case Folding*

Sebelum	Sesudah
Tiket semahal ini jangan dinormalisasikan, jangan overproud juga karna ada artis besar. Nanti promotor lain jadi ikut ikutan bikin harga tiket jadi nggak ngotak next konser☺	tiket semahal ini jangan dinormalisasikan, jangan overproud juga karna ada artis besar. nanti promotor lain jadi ikut ikutan bikin harga tiket jadi nggak ngotak next konser☺

B. Cleaning

Data yang telah dilakukan tahap *case folding* belum dapat disebut data yang bersih karena mengandung berbagai *item* yang tidak diperlukan dalam model analisis sentimen seperti *html, emoticon, hashtag, mention, url*, teks duplikat, angka, tanda baca, spasi yang tidak perlu, dan karakter tunggal. Maka dari itu diperlukan tahap *cleaning* yang akan menyaring dan menghapus berbagai *item* yang tidak diperlukan dalam analisis sentimen dengan tabel 3.3 sebagai contoh penerapan dari tahap *cleaning.*

Tabel 3.3 Contoh Hasil Tahap *Cleaning*

Sebelum	Sesudah
tiket semahal ini jangan dinormalisasikan, jangan overproud juga karna ada artis besar. nanti promotor lain jadi ikut ikutan bikin harga tiket jadi nggak ngotak next konser☺	tiket semahal ini jangan dinormalisasikan jangan overproud juga karna ada artis besar nanti promotor lain jadi ikut ikutan bikin harga tiket jadi nggak ngotak next konser

C. Filtering

Setelah melalui tahap *cleaning*, dokumen yang mengandung kata-kata tidak efektif dan tidak memiliki makna akan dihapus dan disaring dengan menggunakan tahap *filtering* sehingga dapat mengurangi kata yang akan disimpan ke dalam korpus.

Adapun tabel 3.4 adalah contoh penerapan tahap *filtering*.

Tabel 3.4 Contoh Hasil Tahap *Filtering*

Sebelum	Sesudah
tiket semahal ini jangan dinormalisasikan jangan overproud juga karna ada artis besar nanti promotor lain jadi ikut ikutan bikin harga tiket jadi nggak ngotak next konser	tiket semahal jangan dinormalisasikan jangan overproud karna artis besar promotor jadi ikut ikutan bikin harga tiket jadi ngotak next konser

D. Tokenization

Data yang telah dilakukan tahap *filtering* yang terdiri dari berbagai kalimat akan diseleksi dan dipotong pada tahap *tokenization* menjadi kata yang terpisah dengan menggunakan pemisah seperti tanda koma (,), titik (.), mapupun tanda pemisah yang lain. Adapun tabel 3.5 adalah contoh penerapan tahap *tokenization*.

Tabel 3.5 Contoh Hasil Tahap *Tokenization*

Sebelum	Sesudah
tiket semahal ini jangan dinormalisasikan jangan overproud juga karna ada artis besar nanti promotor lain jadi ikut ikutan bikin harga tiket jadi nggak ngotak next konser	['tiket', 'semahal', 'jangan', 'dinormalisasikan', 'jangan', 'overproud', 'karna', 'artis', 'besar', 'promotor', 'jadi', 'ikut', 'ikutan', 'bikin', 'harga', 'tiket', 'jadi', 'ngotak', 'next', 'konser']

E. Normalize

Data hasil dari tahap *tokenization* yang mengandung kata-kata *slang*, singkatan dan bentuk kata tidak baku akan diubah dengan tahap *normalize* menjadi kata yang baku dan sesuai dengan ejaan dalam Kamus Besar Bahasa Indonesia (KBBI) (Nuri, 2022). Adapun tabel 3.6 adalah contoh penerapan tahap *normalize*.

Tabel 3.6 Contoh Hasil Tahap *Normalize*

Sebelum	Sesudah
['tiket', 'semahal', 'jangan', 'dinormalisasikan', 'jangan', 'overproud', 'karna', 'artis', 'besar', 'promotor', 'jadi', 'ikut', 'ikutan', 'bikin', 'harga', 'tiket', 'jadi', 'ngotak', 'next', 'konser']	['tiket', 'semahal', 'jangan', 'dinormalisasikan', 'jangan', 'overproud', 'karena', 'artis', 'besar', 'promotor', 'jadi', 'ikut', 'ikutan', 'bikin', 'harga', 'tiket', 'jadi', 'ngotak', 'next', 'konser']

F. Stemming

Data yang diperoleh dari hasil tahap *normalize* akan dilakukan tahap penyaringan terhadap kata yang mengandung kata sambung, kata ganti, dan kata depan yang akan diubah dengan tahap *stemming* menjadi bentuk kata dasar dengan menghilangkan awalan atau akhiran dari kata tersebut dengan memanfaatkan *library Python*. Tahap ini merupakan akhir dari tahapan *text-preprocessing* dengan tabel 3.7 sebagai contoh penerapan dari tahap *stemming*.

Tabel 3.7 Contoh Hasil Tahap *Stemming*

Sebelum	Sesudah
['tiket', 'semahal', 'jangan', 'dinormalisasikan', 'jangan', 'overproud', 'karena', 'artis', 'besar', 'promotor', 'jadi', 'ikut', 'ikutan', 'bikin', 'harga', 'tiket', 'jadi', 'ngotak', 'next', 'konser']	['tiket', 'mahal', 'jangan', 'normalisasi', 'jangan', 'overproud', 'karena', 'artis', 'besar', 'promotor', 'jadi', 'ikut', 'ikut', 'bikin', 'harga', 'tiket', 'jadi', 'ngotak', 'next', 'konser']

3.3.3 Ekstraksi Fitur

Setelah semua *DataFrame* melalui tahap *text-preprocessing*, dilanjutkan dengan tahap ekstraksi fitur yang berguna untuk memberikan nilai pada setiap kata dan memudahkan dalam tahap perancangan model klasifikasi. Pada tahap ektraksi fitur, terdapat dua tahap yang akan dilakukan oleh peneliti, yaitu tahap pembobotan kata *(term weighting)* dan pembuatan vektor kata *(word vector).*

Pada tahap pembobotan kata, data kalimat hasil dari *text-preprocessing* akan diberikan nilai bobot terhadap setiap kata yang terdapat dalam setiap kalimat dalam dokumen dengan menggunakan Metode *TF-IDF* dan *TF-RF* yang rumusnya telah dijelaskan pada bab sebelumnya. Setelah itu, data hasil dari pembobotan masing-masing metode akan melalui tahap pemberian vektor kata. Data tersebut akan digabungkan menjadi kumpulan *array* yang kemudian diubah ke dalam bentuk

matriks, dimana setiap baris yang terdapat di dalam matriks menunjukkan baris pada dokumen, dan setiap kolom yang terdapat di dalam matriks menunjukkan semua kata yang ada di dalam dokumen teks secara keseluruhan (Imron. Ali, 2019).

3.3.4 Pembagian *Dataset*

Pada tahap ini, *DataFrame* hasil dari pembobotan masing-masing metode akan dibagi menjadi data latih dan data uji yang akan digunakan dalam tahap perancangan model klasifikasi dengan perbandingan rasio 70:30, dengan 70% sebagai data latih dan 30% sebagai data uji (Salsabila, 2022).

3.3.5 Perancangan Model

A. Algoritma *Naïve Bayes*

Pada tahap ini, peneliti akan melakukan perancangan model klasifikasi berbasis Algoritma *Naïve Bayes* serta pelatihan dengan menggunakan *dataset* yang telah dibagi menjadi data latih dan data uji pada tahap sebelumnya. Nilai *input* yang akan digunakan dalam pelatihan model klasifikasi berasal dari data latih masing-masing metode pembobotan berupa data vektor kata, sehingga dalam penelitian ini akan digunakan dua kombinasi dari Metode *TF-IDF*

dan Algoritma *Naïve Bayes*, serta Metode *TF-RF* dan Algoritma *Naïve Bayes* (Deolika & Taufiq Luthfi, 2019).

B. **Uji Model**

Setelah pelatihan data selesai, dilakukan tahap uji model untuk mengetahui performa dari model klasifikasi yang telah dirancang. Pengambilanc data dilakukan secara acak dengan memanfaatkan *library* dari *Python*. Setelah uji model selesai, dapat diketahui nilai akurasi yang menunjukkan keakuratan dari metode yang digunakan.

C. **Evaluasi Model**

Merupakan tahap terakhir dari penelitian ini dimana peneliti akan melakukan evaluasi pada model analisis sentimen yang telah dirancang dengan memanfaatkan *Confusion Matrix* untuk mendapatkan nilai akurasi, akurasi, presisi, *recall,* dan *f1-score*.

3.4 Perangkat Penelitian

Berikut merupakan spesifikasi baik perangkat keras *(hardware)* maupun perangkat lunak *(software)* yang digunakan oleh peneliti pada penelitian ini.

Tabel 3.8 Spesifikasi Perangkat yang Digunakan

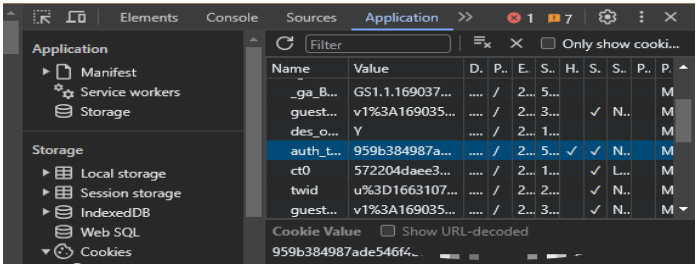
Hardware	Laptop LENOVO ideapad 330	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
		8 GB RAM
		256 GB SSD
		Monitor 14 inch
Software	Sistem Operasi	*Windows 10 Pro*
	Tools	• *Google Colaboratory* • *Google Spreadsheet* • *Microsoft Office Word 2019*

BAB IV
HASIL DAN PEMBAHASAN

4.1 Persiapan Data

4.1.1 Crawling Data

Pengumpulan data sampel dilakukan dengan menggunakan *tools* berupa *Tweet-Harvest* yang terhubung dengan media sosial *X. Tweet-Harvest* merupakan sebuah *command-line tools* berbasis *npm* yang menggunakan Pustaka Otomatisasi Terbuka *Playwright* dalam pengumpulan data sampel berupa data *tweet* dari pengguna media sosial *X* berdasarkan kata kunci dan rentang waktu tertentu yang kemudian akan disimpan ke dalam dokumen dengan format *file CSV* (socket.dev, 2023). Agar dapat menggunakan *Tweet-Harvest* dengan optimal, dibutuhkan *Authorization Token* yang dapat diperoleh dengan mengekstraksi *auth_token cookie* pada halaman utama media sosial *X* di peramban.



Gambar 4.1 *auth_token cookie* media sosial *X*

Sebelum menjalankan *Tweet-Harvest,* langkah pertama yang harus dilakukan yaitu menginstal *Node.js* pada *compiler Python.* Hal ini diperlukan karena *Tweet-Harvest* membutuhkan *Node.js* untuk berjalan secara optimal. Pada penelitian ini peneliti menggunakan *Google Colaboratory* sebagai *compiler Python* yang terhubung dengan *Google Drive* sebagai media penyimpanan data. *Google Colaboratory* merupakan *compiler Python* berbasis *cloud* buatan Google yang dapat digunakan hanya dengan membuka halaman *https://colab.google/* pada peramban tanpa perlu melakukan instalasi terlebih dahulu serta dapat diakes ke dalam berbagai perangkat. Kode program 4.1 merupakan cara instalasi *Node.js* pada *compiler Google Colaboratory*.

```
# Import Python Package yang dibutuhkan
!pip install pandas

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

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

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

!node -v
```

Kode Program 4.1 Instalasi *Node.js*

Setelah menginstal *Node.js* dengan mengetikan kode perintah seperti pada kode program 4.1, maka dapat dilakukan tahap *crawling data* pada *compiler* dengan menggunakan *tools Tweet-Harvest*. Kode program 4.2 merupakan kode yang berisi perintah untuk menjalankan tahap *crawling data* dengan menggunakan *tools Tweet-Harvest*.

```
# Auth Token Pada Media Sosial X
x_auth_token = '959b384987a[redacted]'

# Crawl Data dan Penyimpanan dalam Format File CSV
filename = 'tiket_konser_pandemi.csv'
search_keyword = 'harga tiket konser lang:id since:2023-06-
21'
limit = 1500

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

Kode Program 4.2 Tahap *Crawling Data*

Pada kode program 4.2, dilakukan tahap *crawling data* untuk mengumpulkan data sampel berupa data *tweet* dari pengguna media sosial *X* berbahasa Indonesia berdasarkan kata kunci "harga tiket konser" dengan limit 1500 *tweet* serta rentang waktu yaitu sejak tanggal 21 Juni 2023 menganut pada kebijakan yang telah diumumkan oleh Pemerintah Pusat yang secara resmi mencabut status pandemi *Covid-19* di Indonesia pada tanggal 21 Juni 2023 (Humas Kemensetneg, 2023). Diperoleh hasil pengumpulan data sampel sebanyak 1350 data *tweet* yang nantinya akan disimpan dalam *DataFrame* dengan format *file CSV* seperti pada gambar 4.2.

	created_at	id_str	full_text	quote_count	reply_count	retweet_count	favorite_count	lang
0	Thu Nov 30 16:59:17 +0000 2023	1730270286623469807	stop normalize harga tiket konser kpop gak ngotak	279	50	2680	6186	in
1	Fri Dec 01 03:17:40 +0000 2023	1730425906261643702	masa harga tiket konser di negara berkembang l...	105	16	657	1734	in
2	Fri Dec 01 00:18:51 +0000 2023	1730380904823144609	FAK MEN INI BOCORANNYA SAMA SEATPLANNYA KELUAR...	413	59	372	1293	in
3	Fri Dec 01 07:26:07 +0000 2023	1730488429660733925	Tembak aja itu promotor yang suka mahalin harg...	0	0	4	1	in
4	Thu Nov 30 23:57:32 +0000 2023	1730375540006379700	Kalo ada konser yang ga sold karena harga tike...	480	77	4276	10235	in
...	...	...	...	...	...	...	...	...
1291	Wed Jun 21 02:59:44	1671352172863459328	Bukan cuman tiket konser aja yang	0	0	0	0	in

Gambar 4.2 Hasil Tahap *Crawling Data*

4.1.2 Labelling

Setelah diperoleh data sampel yang disimpan dalam *DataFrame* dengan format *file CSV*, akan dilakukan tahap *labelling. DataFrame* yang telah dikumpulkan akan diberikan label sentimen positif dan negatif secara manual dengan bantuan pakar Bahasa Indonesia sebagai validator untuk menghindari terjadinya subjektivitas dan bias dalam penentuan label sentimen pada hasil data *tweet* yang telah diperoleh dari tahap sebelumnya sehingga dapat diperoleh hasil pemberian label sentimen seperti pada gambar 4.3 yang akan disimpan ke dalam kolom *'label'.*

NO	USERNAME	TWEET	LABEL
1	achabuccha	WTB Tiket Konser Flowerful JKT48 12th Anniversary - Surabaya Section Rose : 1 Ticket Only Harga Wajar, COD di Venue Unesa Surabaya #JKT48Anniversary #JKTAnniversary12Ticket	Positif
2	nacygnus	bntr ych jd gini klo tiket di bioskop jg seperempat harga konser mending uangnya gw simpen buat nunggu rill konser indo ga sieh, adh bimbang😔	Negatif
3	i_dhimas	@kalimrapipp Jadi si A beli tiket di calo pas hari H ticketing/war ticket, si calo dapet tiket harga normal (anggaplah 4jt), nah di wts/jual seharga 5jt, terus perjanjiannya dp dulu dan pelunasan di venue hari H konser. Transaksi tuh deal dan payment. Eh, terus si calo ini ngejual lagi...	Negatif
4	_WawanH_	Edisi belum nyerah! WTB tiket konser anniversary 12th jkt48 kategori TULIP dm harga, butuh 1 aja, COD venue 🙏 #wts #wtb #jkt48	Positif
5	swalalalisaaa	WTS TIKET TWICE READY TO BE IN JAKARTA ✅ 2 tix - QN Sebelahan (Sec : Gray East Tier 3) ✅ Harga sesuai web/boleh nego ✅ DP sisa COD venue diurus sampai tiket fisik NO SCAM, TIKET PRIBADI! DM FOR MORE tags wts wtb twice konser in jakarta #TWICE #TWICE_5TH_WORLD_TOUR https://t.co/2FeJDcrDIC	Positif
6	Urtrust___	WTS READY 4 TIKET FESTIVAL COLDPLAY HARGA : 7.5/TIX BISA COD SKRG YAAA DEPAN GATE FESTIVAL #ColdplayJakarta #ColdplayInJakarta #wtscoldplay #wtbcoldplay #coldplay	Positif
7	amethyst_i8	WTS Tiket Konser Coldplay Singapore Day 2 GENERAL STANDING (x1 TIKET) date 24/1/2024 ✅ print at home ✅ ina bank Price : sesuai harga web Alasan dijual : kelebihan check out #coldplayinsingapore #ColdplayInJakarta #ColdplayinJakarta2023	Positif

Gambar 4.3 Hasil Tahap *Labelling*

Tabel 4.1 Contoh Hasil Tahap *Labelling* pada *DataFrame*

Data *Tweet*	Label Sentimen
Harga tiket Coldplay memang mahal, tapi tetap tidak menghalangi penggemar untuk menikmati konser mereka. Bagi para haters, tetaplah berada di sini karena kami akan tetap mendukung dan melindungi Coldplay dari segala kritikan yang tidak adil.	Positif
Bisa gak sih kita boikot promotor konser tukang rampok? Jangan dibeli gitu. Biar nggak kebiasaan jual tiket harga nggak waras gitu.	Negatif

Berdasarkan tabel 4.1, dapat diketahui bahwa pemberian label sentimen pada masing-masing data *tweet* dilakukan berdasarkan reaksi dari para pengguna media sosial *X* terhadap fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia. Pada penelitian ini, peneliti bersama pakar Bahasa Indonesia memberikan label sentimen "Positif" pada data *tweet* yang dapat diartikan sebagai reaksi pro atau setuju dari para pengguna media sosial *X* mengenai adanya fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia. Sedangkan peneliti memberikan label sentimen "Negatif" pada data *tweet* yang dapat diartikan sebagai reaksi kontra atau penolakan dari para pengguna media sosial *X* adanya fenomena tersebut.

Untuk mengetahui hasil distribusi dari masing-masing label sentimen di dalam *DataFrame*, peneliti menggunakan beberapa *library Python* yaitu *pandas* untuk memanipulasi serta analisis data, *seaborn* dan *matpolib* untuk memvisualisasikan hasil dari distribusi label sentimen ke dalam bentuk grafik agar lebih mudah dimengerti.

```
#import library yang dibutuhkan
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns

print('Distribusi Label Sentimen pada
DataFrame :')
data.label.value_counts()
```

Kode Program 4.3 Menampilkan Distribusi Label Sentimen pada *DataFrame*

```
Distribusi Label Sentimen pada Dataframe :
label
Negatif    864
Positif    486
Name: count, dtype: int64
```

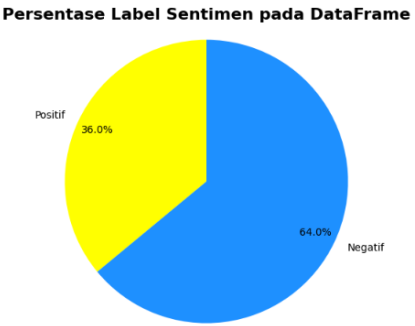
Gambar 4.4 *Output* Distribusi Label Sentimen

Dengan mengetikkan perintah pada *compiler* seperti yang terdapat dalam kode program 4.3, dapat diperoleh *output* berupa gambar 4.4 yaitu hasil dari distribusi label sentimen yang terdapat di dalam *DataFrame*. Berdasarkan gambar 4.4, dapat diketahui bahwa hasil distribusi dari masing-masing label sentimen yang terdapat pada *DataFrame* yaitu label

sentimen “Negatif” sebanyak 864 *tweet*, serta label sentimen “Positif” sebanyak 486 *tweet*. Lalu untuk mengetahui persentase dari distribusi label sentimen pada *DataFrame*, peneliti menggunakan kode perintah seperti pada kode program 4.4 agar dapat menampilkan hasil hasil perbandingan persentase label sentimen dalam bentuk grafik *pie* dengan gambar 4.5 sebagai *output* dari kode perintah tersebut.

```
# membuat grafik pie untuk menampilkan informasi label sentimen
labels = ['Positif', 'Negatif']
sizes = [len(data[data['label'] == 'Positif']),
 len(data[data['label'] == 'Negatif'])]
colors = ['#FFFF00', '#1E90FF']

plt.pie(sizes, colors = colors, labels = labels, autopct='%1.1f%%', startangle=90, pctdistance=0.85)
plt.title('Persentase Label Sentimen pada DataFrame', fontsize=16, fontweight='bold')
plt.axis('equal')
plt.tight_layout()
```

Kode Program 4.4 Menampilkan Persentase Distribusi Label Sentimen



Gambar 4.5 Persentase Label Sentimen pada *DataFrame*

Berdasarkan gambar 4.5, dapat diketahui bahwa perbandingan persentase dari masing-masing label sentimen yaitu label sentimen "Positif" sebanyak 36% dan label sentimen "Negatif" sebanyak 64% dari keseluruhan *DataFrame* yang terdapat di dalam korpus.

4.2 Text-Preprocessing

Data *tweet* yang telah diberikan label sentimen pada tahap *labelling* masih merupakan data kotor karena terdiri dari berbagai *item* yang tidak diperlukan dalam perancangan model sehingga diperlukan pengolahan pada *DataFrame* agar menjadi lebih rapi dan bersih yang akan dilakukan dalam tahap *text-preprocessing* yang terdiri dari 6 tahapan yaitu *case folding, cleaning, tokenization, normalize, stopwords removal,* dan *stemming.*

```
# import library yang diperlukan
import pandas as pd
import numpy as np
!pip install Sastrawi
!pip install swifter

# membaca dan menampilkan file (Koneksi
dengan GDrive)
from google.colab import drive
drive.mount('/content/drive')
```

Kode Program 4.5 *Library* yang digunakan pada Tahap *Text-Preprocessing*

Pada kode program 4.5, peneliti memanfaatkan *library Pandas* sebagai *library Python* yang berfungsi untuk memanipulasi dan analisis data, *library NumPy* untuk keperluan operasi matematika dalam *compiler*. Selain itu

peneliti juga melakukan instalasi untuk beberapa *library* tambahan seperti *library Satrawi* yang merupakan *NLP (Natural Language Processing)* khusus untuk kata berbahasa Indonesia dan *library Swifter* untuk mempercepat proses yang terdapat pada *compiler*. Untuk menyimpan data hasil dari tahap *text-preprocessing, peneliti* melakukan koneksi antara *compiler* dengan *Google Drive.*

4.2.1 Case Folding

Pada tahap *case folding*, data hasil tahap *labelling* yang masih terdiri dari berbagai kalimat dengan huruf kapital akan diubah menjadi huruf kecil *(lower case)* dengan mengetikkan kode perintah seperti pada kode program 4.6 dalam *compiler* sehingga dapat diperoleh *output* berupa gambar 4.6 yaitu teks yang berubah menjadi huruf kecil jika dibandingkan dengan tahap sebelumnya. *DataFrame* hasil dari tahap *case folding* selanjutnya akan disimpan ke dalam kolom *'case_folding'* pada tabel.

```
# membuat kolom baru
data['case_folding'] = ''

#Filtering - Lower Text
data['case_folding'] =
data['tweet'].str.lower()

print ('Hasil dari Case Folding : \n')
print (data['case_folding'].head())
```

Kode Program 4.6 Tahap *Case Folding*

```
Hasil dari Case Folding :

0    wtb tiket konser flowerful jkt48 12th annivers...
1    bntr ych jd gini klo tiket di bioskop jg seper...
2    @kalimrapipp jadi si a beli tiket di calo pas ...
3    edisi belum nyerah!  wtb tiket konser annivers...
4    wts tiket twice ready to be in jakarta ✅ 2 ti...
Name: case_folding, dtype: object
```

Gambar 4.6 *Output* dari Tahap *Case Folding*

4.2.2 Cleaning

Pada tahap *cleaning*, *DataFrame* hasil dari tahap *case folding* akan dilakukan proses pembersihan dari berbagai *item* yang tidak diperlukan dalam perancangan model analisis sentimen. Agar dapat melakukan tahap cleaning secara optimal, peneliti menggunakan perintah untuk memanggil modul bawaan dari *Python* yaitu *string, re,* dan *nltk* seperti yang ditampilkan pada kode program 4.7.

```
import string
import re #regex library
import nltk
nltk.download('punkt')
```

Kode Program 4.7 Modul *Python* yang digunakan dalam Tahap *Cleaning*

Sebelum melakukan tahap *cleaning*, peneliti mengetikkan kode perintah seperti pada kode program 4.8 untuk mengetahui jumlah data yang terdapat pada *DataFrame*. Hal ini dilakukan oleh peneliti agar dapat mengamati apakah ada perubahan jumlah data pada

DataFrame baik sebelum maupun sesudah tahap *cleaning* dilakukan.

```
print("Jumlah Data sebelum Dibersihkan :", len(data), "tweet")
```

```
Jumlah Data sebelum Dibersihkan : 1350 tweet
```

Kode Program 4.8 Jumlah *DataFrame* sebelum Tahap *Cleaning*

Berdasarkan kode program 4.8, dapat diketahui bahwa sebelum tahap *cleaning* dilakukan, jumlah data yang terdapat di dalam *DataFrame* hasil dari tahap *case folding* yaitu sebanyak 1350 *tweet*.

```
import string
import re #regex library
import nltk
nltk.download('punkt')



# -------------------- CLEANING -------------------

# membuat kolom baru
data['cleaning'] = ''

def remove_tweet_special(text):
    # menghapus tab, baris baru, dan backslash
    text = text.replace('\\t'," ").replace('\\n'," 
").replace('\\u'," ").replace('\\',"")
    # menghapus non ASCII (emoticon, chinese word, 
.etc)
    text = text.encode('ascii', 
'replace').decode('ascii')
    # menghapus mention, link, hashtag
    text = ' '.join(re.sub("([@#][A-Za-z0-
9]+)|(\w+:\/\/\S+)"," ", text).split())
    # menghapus URL tidak lengkap
    return text.replace("http://", " 
").replace("https://", " ")
data['cleaning'] = 
data['case_folding'].apply(remove_tweet_special)

# menghapus angka
def remove_number(text):
    return  re.sub(r"\d+", "", text)
data['cleaning'] = 
data['cleaning'].apply(remove_number)

# menghapus tanda baca
def remove_punctuation(text):
```

```
    return
text.translate(str.maketrans("","",string.punctuati
on))
data['cleaning'] =
data['cleaning'].apply(remove_punctuation)

# menghapus spasi di awal dan di akhir
def remove_whitespace_LT(text):
    return text.strip()

data['cleaning'] =
data['cleaning'].apply(remove_whitespace_LT)

# menghapus beberapa spasi menjadi 1 spasi
def remove_whitespace_multiple(text):
    return re.sub('\s+',' ',text)

data['cleaning'] =
data['cleaning'].apply(remove_whitespace_multiple)

# menghapus karakter tunggal
def remove_singl_char(text):
    return re.sub(r"\b[a-zA-Z]\b", "", text)

data['cleaning'] =
data['cleaning'].apply(remove_singl_char)

# menghapus teks duplikat
data = data.drop_duplicates()
data = data.reset_index(drop=True)

print('Hasil dari Cleaning : \n')
print(data['cleaning'].head(), '\n')
```

Kode Program 4.9 Tahap *Cleaning*

Berdasarkan kode program 4.9, terdapat beberapa proses yang dilakukan oleh peneliti dalam tahap *cleaning* agar menghasilkan *DataFrame* yang bersih sehingga dapat digunakan dalam perancangan model analisis sentimen, diantaranya yaitu tahap penghapusan teks duplikat, item yang tidak diperlukan, angka, tanda baca, spasi, serta karakter tunggal sehingga dapat diperoleh hasil seperti pada gambar 4.7 yang merupakan

output dari tahap cleaning berupa teks bersih yang disimpan ke dalam kolom *'cleaning'* pada tabel.

```
Hasil dari Cleaning :

0    wtb tiket konser flowerful jkt th anniversary ...
1    bntr ych jd gini klo tiket di bioskop jg seper...
2    jadi si  beli tiket di calo pas hari  ticketin...
3    edisi belum nyerah wtb tiket konser anniversar...
4    wts tiket twice ready to be in jakarta tix qn ...
Name: cleaning, dtype: object
```

Gambar 4.7 *Output* dari Tahap *Cleaning*

Setelah melakukan tahap *cleaning*, peneliti mengetikkan kode perintah seperti pada kode program 4.10 untuk mengetahui jumlah data yang terdapat pada *DataFrame*. Hal ini dilakukan oleh peneliti agar dapat mengamati kembali apakah ada perubahan jumlah data pada *DataFrame* baik sebelum maupun sesudah tahap *cleaning* dilakukan.

```
print("Jumlah Data setelah Dibersihkan :", len(data), "tweet")
```

```
Jumlah Data setelah Dibersihkan : 1333 tweet
```

Kode Program 4.10 Jumlah *DataFrame* setelah Tahap *Cleaning*

Berdasarkan kode program 4.10, dapat diketahui bahwa setelah tahap *cleaning* dilakukan, jumlah data yang terdapat di dalam *DataFrame* hasil dari tahap *case folding* yaitu sebanyak 1333 *tweet*. Terdapat selisih data sebanyak 17 *tweet* jika dibandingkan dengan jumlah *DataFrame* sebelum dilakukan tahap *cleaning*. Hal ini dapat terjadi dikarenakan pada tahap *cleaning* terdapat

tahap penghapusan teks duplikat yang mendeteksi sebanyak 17 teks duplikat pada *DataFrame*.

4.2.3 Filtering

Setelah melalui tahap *cleaning*, *DataFrame* yang mengandung kata-kata tidak efektif dan tidak memiliki makna akan dihapus dan disaring dengan menggunakan tahap *filtering* berupa *stopwords removal* sehingga dapat mengurangi kata yang akan disimpan ke dalam korpus. Untuk dapat menjalankan tahap *filtering*, peneliti memanfaatkan fungsi *'stopword'* beserta *library Sastrawi* dengan mengetikkan kode perintah seperti yang ditunjukkan oleh kode program 4.11 pada *compiler*.

```
import Sastrawi
from Sastrawi.StopWordRemover.StopWordRemoverFactory
import StopWordRemoverFactory, StopWordRemover,
ArrayDictionary
more_stop_word = []

stop_words = StopWordRemoverFactory().get_stop_words()
new_array = ArrayDictionary(stop_words)
stop_words_remover_new = StopWordRemover(new_array)

# membuat kolom baru
data['filtering'] = ''

def stopword (str_text) :
  str_text = stop_words_remover_new.remove(str_text)
  return str_text

# menambahkan data hasil stopwords removal ke dalam
kolom baru di file csv
data['filtering'] = data['cleaning'].apply(stopword)

print('Hasil dari Filtering (Stopwords Removal) : \n')
print(data['filtering'].head(10))
```

Kode Program 4.11 Tahap *Filtering*

Berdasarkan kode program 4.11, peneliti melakukan beberapa tahapan dalam tahap *filtering*. Yang pertama yaitu mengimpor *library* beserta *package* yang diperlukan untuk menjalankan tahap filtering. selanjutnya yaitu tahap pembuatan *Stopwords Dictionary* yang berfungsi sebagai tempat untuk menampung kata-kata berbahasa Indonesia yang terdapat di dalam *library Sastrawi*. Setelah itu dilanjutkan dengan pembuatan fungsi yang berisi program untuk menjalankan tahap *filtering* dengan menggunakan kata-kata yang telah tertampung di dalam *Stopwords Dictionary*. Kemudian dilakukan tahap *filtering* dengan adanya pemanggilan fungsi ke dalam *DataFrame* hasil dari tahap *cleaning* sehingga dihasilkan *DataFrame* yang bersih dari kata-kata yang tidak efektif. Setelah dilakukan tahap *stopwords removal*, *DataFrame* yang telah diperoleh akan disimpan ke dalam kolom *'filtering'* pada tabel dengan gambar 4.8 sebagai *output* dari tahap *filtering*.

```
Hasil dari Filtering (Stopwords Removal) :

0    wtb tiket konser flowerful jkt th anniversary ...
1    bntr ych jd gini klo tiket bioskop jg seperemp...
2    jadi si  beli tiket calo pas hari  ticketingwa...
3    edisi nyerah wtb tiket konser anniversary th j...
4    wts tiket twice ready to be in jakarta tix qn ...
5    wts ready tiket festival coldplay harga tix co...
6    wts tiket konser coldplay singapore day genera...
7    yg beli harga tiket gilaan jakarta mending bel...
8    mencari tiket konser anniv jeketi jasmine trib...
9    temenku jg suka ngome harga tiket konser udah ...
Name: filtering, dtype: object
```

Gambar 4.8 *Output* dari Tahap *Filtering*

4.2.4 Tokenization

Pada tahap *tokenization*, *DataFrame* yang telah dibersihkan pada tahap *filtering* akan diseleksi dan dipotong menjadi kata yang terpisah dari yang sebelumnya masih dalam bentuk kalimat utuh dengan menggunakan tanda pemisah seperti tanda koma (,), titik (.), mapupun tanda pemisah yang lain. Kode program 4.12 merupakan kode perintah yang digunakan oleh peneliti untuk menjalankan tahap *tokenization* pada *compiler*.

```
# import fungsi word_tokenize  dari NLTK
from nltk.tokenize import word_tokenize

# ---------------- TOKENIZING ----------------

# membuat kolom baru
data['tokenize'] = ''

# NLTK word tokenize
def word_tokenize_wrapper(text):
    return word_tokenize(text)

# menambahkan data hasil tokenize ke dalam
kolom baru di file csv
data['tokenize'] =
data['filtering'].apply(word_tokenize_wrapper)

print('Hasil dari Tokenizing : \n')
print(data['tokenize'].head(10))
```

Kode Program 4.12 Tahap *Tokenization*

Berdasarkan kode program 4.12, peneliti memanggil *package word_tokenize* yang merupakan salah satu *package* yang terdapat di dalam *library NLTK.* Selanjutnya peneliti membuat sebuah fungsi yang akan digunakan sebagai sebagai tempat untuk menampung

perintah dalam menjalankan tahap *tokenization.* Setelah itu dilanjutkan dengan tahap *tokenization* dengan pemanggilan fungsi pada *DataFrame* hasil dari tahap *cleaning* yang kemudian akan disimpan ke dalam kolom *'tokenize'* pada tabel dengan gambar 4.9 sebagai *output* dari tahap *tokenization*.

```
Hasil dari Tokenizing :

0    [wtb, tiket, konser, flowerful, jkt, th, anniv...
1    [bntr, ych, jd, gini, klo, tiket, bioskop, jg,...
2    [jadi, si, beli, tiket, calo, pas, hari, ticke...
3    [edisi, nyerah, wtb, tiket, konser, anniversar...
4    [wts, tiket, twice, ready, to, be, in, jakarta...
5    [wts, ready, tiket, festival, coldplay, harga,...
6    [wts, tiket, konser, coldplay, singapore, day,...
7    [yg, beli, harga, tiket, gilaan, jakarta, mend...
8    [mencari, tiket, konser, anniv, jeketi, jasmin...
9    [temenku, jg, suka, ngome, harga, tiket, konse...
Name: tokenize, dtype: object
```

Gambar 4.9 *Output* dari Tahap *Tokenization*

4.2.5 Normalize

Setelah melalui tahap *filtering*, *DataFrame* yang dihasilkan dari tahap *filtering* belum dapat dikatakan sebagai data yang bersih dikarenakan di dalam *DataFrame* tersebut masih mengandung kata-kata slang, singkatan dan bentuk kata tidak baku. Oleh karena itu diperlukan tahap *normalize* untuk menyaring dan mengubah kumpulan kata tersebut menjadi kata yang baku dan sesuai dengan ejaan dalam Kamus Besar Bahasa Indonesia (KBBI) . Untuk dapat menjalankan tahap *normalize*, peneliti mengetikkan kode perintah

seperti yang ditunjukkan oleh kode perintah 4.13 pada *compiler*.

```
# membaca format file csv dari GDrive
normalized_word =
pd.read_csv("/content/drive/MyDrive/Skripsi_Muhammad_
Amirul_S/text_preprocessing/colloquial-indonesian-
lexicon.csv")

# ---------------------- NORMALIZE -------------------

# membuat kolom baru
data['normalize'] = ''

normalized_word_dict = {}

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

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

# menambahkan data hasil normalize ke dalam kolom
baru di file csv
data['normalize'] =
data['tokenize'].apply(normalized_term)

print('Hasil dari Normalize : \n')
print(data['normalize'].head(10))
```

Kode Program 4.13 Tahap *Normalize*

Berdasarkan kode program 4.13, dapat diketahui bahwa peneliti melakukan beberapa tahapan untuk melakukan tahap *normalize*. Yang pertama yaitu menyiapkan *file colloquial-indonesian-lexicon.csv* yang berisi kumpulan kata baku yang telah disesuaikan dengan ejaan Kamus Besar Bahasa Indonesia (KBBI) sebagai sumber data yang akan digunakan dalam pembuatan *Normalize Dictionary*. Setelah itu dilanjutkan dengan pembuatan fungsi yang berisi program untuk

menjalankan tahap *normalize* serta mempercepat proses pengecekan ada atau tidaknya data token hasil dari tahap *tokenization* pada *dictionary.* Kemudian dilakukan tahap *normalize* dengan adanya pemanggilan fungsi ke dalam *DataFrame* hasil dari tahap *tokenization*. *DataFrame* yang telah dihasilkan akan disimpan ke dalam kolom *'normalize'* pada tabel dengan gambar 4.10 sebagai *output* dari tahap *normalize*.

```
Hasil dari Normalize :

0    [wtb, tiket, konser, flowerful, jakarta, tahun...
1    [bentar, ych, jadi, begini, kalo, tiket, biosk...
2    [jadi, sih, beli, tiket, calo, pas, hari, tick...
3    [edisi, nyerah, wtb, tiket, konser, anniversar...
4    [wts, tiket, twice, ready, tapi, be, ini, jaka...
5    [wts, ready, tiket, festival, coldplay, harga,...
6    [wts, tiket, konser, coldplay, singapore, day,...
7    [yang, beli, harga, tiket, gilaan, jakarta, me...
8    [mencari, tiket, konser, anniv, jeketi, jasmin...
9    [temenku, juga, suka, ngome, harga, tiket, kon...
Name: normalize, dtype: object
```

Gambar 4.10 *Output* dari Tahap *Normalize*

4.2.6 Stemming

Sebelum *DataFrame* dapat digunakan dalam tahap selanjutnya, dilakukan tahap *stemming* yang merupakan akhir dari tahap *text-preprocessing*. *DataFrame* yang diperoleh dari tahap *normalize* akan dilakukan tahap penyaringan terhadap kata yang mengandung kata sambung, kata ganti, dan kata depan yang akan diubah dengan tahap *stemming* menjadi bentuk kata dasar dengan menghilangkan awalan atau akhiran dari kata tersebut. Agar tahap stemming dapat

dilakukan secara cepat dan efisien, peneliti memanfaatkan *library Sastrawi* untuk menghasilkan data berupa kata dasar dalam Bahasa Indonesia dan *library Swifter* untuk membantu mempercepat tahap *stemming* pada korpus dengan mengetikkan kode perintah seperti yang ditunjukkan oleh kode program 4.14 pada *compiler*.

```
# import Sastrawi package

from Sastrawi.Stemmer.StemmerFactory import
StemmerFactory
import swifter

# membuat kolom baru
data['clean_tweet'] = ''

# membuat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# proses stemmed
def stemmed_wrapper(term):
    return stemmer.stem(term)

term_dict = {}

for document in data['normalize']:
    for term in document:
        if term not in term_dict:
            term_dict[term] = ' '

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

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

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

# mengaplikasikan stemmed term ke dalam dataframe
(mengembalikan kata ke dalam bentuk asli)
def get_stemmed_term(document):
    return [term_dict[term] for term in document]

data['clean_tweet'] =
data['normalize'].swifter.apply(get_stemmed_term)
```

```
print('Hasil dari Stemming : \n')
print(data['clean_tweet'])
```

Kode Program 4.14 Tahap *Stemming*

Berdasarkan kode program 4.14, dapat diketahui bahwa peneliti melakukan beberapa tahapan untuk menjalankan tahap *stemming.* Yang pertama yaitu pengimporan *package* yang diperlukan dalam tahap *stemming*. Sebelum tahap *stemming* dapat dijalankan, dilakukan pembuatan *stemmer, Stemmer Dictionary*, dan fungsi yang di dalamnya berisi kode perintah untuk menjalankan tahap *stemming*. Selanjutnya dilakukan tahap *stemming* dengan mengaplikasikan fungsi *swifter()* pada setiap *DataFrame* hasil dari tahap *normalize* yang kemudian disimpan dalam kolom *'clean_tweet'* pada tabel dengan gambar 4.11 sebagai *output* dari tahap *stemming*.

```
Hasil dari Stemming :

0       [wtb, tiket, konser, flowerful, jakarta, tahun...
1       [bentar, ych, jadi, begini, kalo, tiket, biosk...
2       [jadi, sih, beli, tiket, calo, pas, hari, tick...
3       [edisi, nyerah, wtb, tiket, konser, anniversar...
4       [wts, tiket, twice, ready, tapi, be, ini, jaka...
Name: clean_tweet, Length: 1333, dtype: object
```

Gambar 4.11 *Output* dari Tahap *Stemming*

Setelah melalui seluruh tahapan yang terdapat di dalam tahap *text-preprocessing, DataFrame* yang sebelumnya merupakan data kotor dikarenakan mempunyai berbagai macam *item* yang tidak perlu serta

memiliki bentuk kata yang tidak efektif telah diubah menjadi data yang bersih sehingga dapat digunakan dalam tahap selanjutnya. *DataFrame* yang telah diperoleh dari tahap text-preprocessing kemudian akan disimpan ke dalam *format file CSV* agar dapat digunakan pada tahap selanjutnya. Berikut gambar 4.12 merupakan hasil dari tahap *text-preprocessing* yang telah dilakukan oleh peneliti.

	tweet	label	case_folding	cleaning	filtering	tokenize	normalize	clean_tweet
0	WTB Tiket Konser Flowerful JKT48 12th Annivers...	Positif	wtb tiket konser flowerful jkt48 12th annivers...	wtb tiket konser flowerful jkt th anniversary ...	wtb tiket konser flowerful jkt th anniversary ...	[wtb, tiket, konser, flowerful, jkt, th, anniv...	[wtb, tiket, konser, flowerful, jakarta, tahun...	[wtb, tiket, konser, flowerful, jakarta, tahun...
1	bntr ych jd gini klo tiket di bioskop jg seper...	Negatif	bntr ych jd gini klo tiket di bioskop jg seper...	bntr ych jd gini klo tiket di bioskop jg seper...	bntr ych jd gini klo tiket bioskop jg seperemp...	[bntr, ych, jd, gini, klo, tiket, bioskop, jg, ...	[bentar, ych, jadi, begini, kalo, tiket, biosk...	[bentar, ych, jadi, begini, kalo, tiket, biosk...
2	@kalimrapipp Jadi si A beli tiket di calo pas ...	Negatif	@kalimrapipp jadi si a beli tiket di calo pas ...	jadi si beli tiket di calo pas hari tickellin...	jadi si beli tiket calo pas hari tickellingva...	[jadi, si, beli, tiket, calo, pas, hari, tickel...	[jadi, sih, beli, tiket, calo, pas, hari, tick...	[jadi, sih, beli, tiket, calo, pas, hari, tick...
3	Edisi belum nyerah! WTB tiket konser annivers...	Positif	edisi belum nyerah! wtb tiket konser annivers...	edisi belum nyerah wtb tiket konser anniversar...	edisi nyerah wtb tiket konser anniversary th j...	[edisi, nyerah, wtb, tiket, konser, anniversar...	[edisi, nyerah, wtb, tiket, konser, anniversar...	[edisi, nyerah, wtb, tiket, konser, anniversar...
4	WTS TIKET TWICE READY TO BE IN JAKARTA ✅ 2 ti...	Positif	wts tiket twice ready to be in jakarta ✅ 2 ti...	wts tiket twice ready to be in jakarta tix on ...	wts tiket twice ready to be in jakarta tix on ...	[wts, tiket, twice, ready, to, be, in, jakarta...	[wts, tiket, twice, ready, tapi, be, ini, jaka...	[wts, tiket, twice, ready, tapi, be, ini, jaka...
...	...	...	...	...	...	...	...	...
1328	Para jastip tiket konser bisa gak ya numpang j...	Positif	para jastip tiket konser bisa gak ya numpang j...	para jastip tiket konser bisa gak ya numpang j...	jastip tiket konser gak numpang jualan tiket n...	[jastip, tiket, konser, gak, numpang, jualan, ...	[jastip, tiket, konser, enggak, numpang, jualа...	[jastip, tiket, konser, enggak, numpang, jual'...
1329	Rest jajan pc, boleh konser lagi kalo minus ke...	Positif	rest jajan pc, boleh konser lagi kalo minus ke...	rest jajan pc boleh konser lagi kalo minus keu...	rest jajan pc konser kalo minus keuangan septo...	[rest, jajan, pc, konser, kalo, minus, keuanga...	[rest, jajan, pc, konser, kalo, minus, keuanga...	[rest, jajan, pc, konser, kalo, minus, uang, s...
1330	WTT or WTS tiket konser Niall Horan The Show:...	Positif	wtt or wts tiket konser niall horan the show:...	wtt or wts tiket konser niall horan the show t...	wtt or wts tiket konser niall horan the show t...	[wtt, or, wts, tiket, konser, niall, horan, th...	[wtt, orang, wts, tiket, konser, niall, horan,...	[wtt, orang, wts, tiket, konser, niall, horan,...
1331	@kalihansol perbandingan harga tiket konser kpo...	Negatif	@kalihansol perbandingan harga tiket konser kpo...	perbandingan harga tiket konser kpop by promot...	perbandingan harga tiket konser kpop by promot...	[perbandingan, harga, tiket, konser, kpop, by,...	[perbandingan, harga, tiket, konser, kpop, by,...	[banding, harga, tiket, konser, kpop, by, prom...
1332	aga kaget denger harga lapangan sekali maen bo...	Negatif	aga kaget denger harga lapangan sekali maen bo...	aga kaget denger harga lapangan sekali maen bo...	aga kaget denger harga lapangan sekali maen bo...	[aga, kaget, denger, harga, lapangan, sekali, ...	[aga, kaget, dengar, harga, lapangan, sekali, ...	[aga, kaget, dengar, harga, lapang, sekali, ma...

1333 rows × 8 columns

Gambar 4.12 Hasil dari Tahap *Text-Preprocessing*

4.3 Ekstraksi Fitur

Setelah dilakukan tahap *text-preprocessing*, dilakukan tahap ekstraksi fitur dengan memberikan nilai bobot pada setiap kata yang terdapat di dalam dokumen. Untuk dapat menjalankan tahap ekstraksi fitur dengan optimal, peneliti memanfaatkan *library Python* yaitu *NumPy* dan *Pandas* beserta *Google Drive* sebagai media penyimpanan *file* hasil dari tahap ekstraksi fitur.

```
import pandas as pd
import numpy as np

# membaca dan menampilkan file (Koneksi dengan
GDrive)
from google.colab import drive
drive.mount('/content/drive')
```

Kode Program 4.15 *Library* yang digunakan pada Tahap Ekstraksi Fitur

Sebelum dilakukan tahap ekstraksi fitur, *DataFrame* hasil dari tahap *text-preprocessing* yang bertipe data *string* akan diubah menjadi bertipe data *list* untuk memudahkan proses manipulasi data pada tahap ekstraksi fitur dengan mengetikkan kode perintah seperti yang ditunjukkan oleh kode program 4.16 pada *compiler* dan gambar 4.13 sebagai *output* dari kode perintah tersebut.

```
# mengubah format string ke list
import ast

def convert_text_list(texts):
    texts = ast.literal_eval(texts)
    return [text for text in texts]

data["tweet_list"] =
data["tweet"].apply(convert_text_list)

print(data["tweet_list"])
print("\ntype : ", type(data["tweet_list"]))
```

Kode Program 4.16 Pengubahan Tipe Data *String* ke *List*

```
0       [wtb, tiket, konser, flowerful, jakarta, tahun...
1       [bentar, ych, jadi, begini, kalo, tiket, biosk...
2       [jadi, sih, beli, tiket, calo, pas, hari, tick...
3       [edisi, nyerah, wtb, tiket, konser, anniversar...
4       [wts, tiket, twice, ready, tapi, be, ini, jaka...
                              ...
1328    [jastip, tiket, konser, enggak, numpang, jual,...
1329    [rest, jajan, pc, konser, kalo, minus, uang, s...
1330    [wtt, orang, wts, tiket, konser, niall, horan,...
1331    [banding, harga, tiket, konser, kpop, by, prom...
1332    [aga, kaget, dengar, harga, lapang, sekali, ma...
Name: tweet_list, Length: 1333, dtype: object
```

Gambar 4.13 *Output* Pengubahan Tipe Data *String* ke *List*

Setelah dilakukan pengubahan tipe data dari *string* ke *list*, dilanjutkan dengan pemberian nilai bobot pada setiap kata yang terdapat di dalam dokumen dengan menggunakan Metode *TF-IDF* dan Metode *TF-RF*. Sebagai contoh, peneliti menggunakan beberapa dokumen yang akan dijadikan sebagai sampel dalam penghitungan nilai bobot secara manual.

Tabel 4.2 Contoh Dokumen

Dokumen	*Tweet*
D1	Mana sekarang ada yg matokin harga tiket konser mahal banget. 7,2jt buat section vip dapet apa? Dapet nomor wa artisnya ga? Wkwkwk
D2	@sykesxsand Nah iya, harga tiket konser disini jg lebih mahal daripada di LN ☺
D3	WTB tiket konser coldplay Seating untuk 2 tiket harga normal plis 🙏

Berdasarkan tabel 4.2, terdapat 3 dokumen yang dijadikan sebagai sampel pada penelitian ini. Setelah dilakukan tahap *text-preprocessing*, diperoleh korpus atau kumpulan data bersih sebagai berikut.

D1 : ['patok', 'harga', 'tiket', 'konser', 'mahal', 'banget', 'juta', 'section', 'dapat', 'nomor', 'artis']

D2 : ['iya', 'harga', 'tiket', 'konser', 'juga', 'mahal']

D3 : ['tiket', 'konser', 'coldplay', 'harga', 'normal']

4.3.1 Metode *TF-IDF*

Untuk menghitung nilai bobot dengan menggunakan Metode *TF-IDF,* terdapat beberapa tahapan yang harus dilakukan. Tahap pertama yaitu penghitungan nilai *TF (Term Frequency*) terlebih dahulu dengan menggunakan rumus yang terdapat pada persamaan 2.1. Adapun contoh hasil penghitungan nilai bobot *TF* pada suatu kata dapat dilihat pada tabel 4.3.

Tabel 4.3 Contoh Penghitungan Nilai *TF*

Dokumen	Jumlah Kata	Kata	Frekuensi	*TF*
D1	11	mahal	1	0,09090909
D2	6		1	0,16666667
D3	5		0	0

Berdasarkan tabel 4.3, dapat diketahui bahwa nilai bobot *TF* dapat diperoleh dengan melakukan pembagian antara jumlah *term* terpilih dengan jumlah *term* yang ada dalam suatu dokumen. Sebagai contoh, pada D1 terdiri dari 11 kata yang di dalamnya terdapat kata "mahal" muncul sebanyak 1 kali, sehingga Nilai *TF* yang dihasilkan untuk kata "mahal" sebesar 0,09090909.

IDF yang dapat diperoleh dengan menggunakan rumus yang terdapat pada persamaan 2.2. Dalam Metode *IDF,* suatu *term* yang memiliki frekuensi kemunculan lebih sedikit di dalam dokumen akan dianggap lebih

penting daripada frekuensi kemunculan term yang lebih banyak. Sebagai contoh, berikut merupakan contoh hasil penghitungan nilai bobot *IDF* untuk kata "mahal" .

$$W_{(IDF)}t = log\left(\frac{N}{DF_{(t)}}\right)$$
$$= log\left(\frac{3}{2}\right)$$
$$= 0{,}17609126$$

Setelah diperoleh nilai *IDF*, dapat dilanjutkan dengan penghitungan nilai bobot *TF-IDF.* Dalam Metode *TF-IDF,* nilai bobot yang tinggi akan dihasilkan oleh kata yang memiliki jumlah kemunculan yang lebih banyak dalam suatu dokumen, namun kata tersebut jarang muncul dalam korpus. Nilai bobot *TF-IDF* dapat dihasilkan dengan mengalikan antara nilai bobot TF dengan nilai bobot *IDF.* Sebagai contoh, berikut merupakan contoh hasil penghitungan nilai bobot *TF-IDF* untuk kata "mahal" pada D1.

$$W_{(TF.IDF)} = TF_{(d,t)} \times IDF_{(t)}$$
$$= 0{,}09090909 \times 0{,}17609126$$
$$= 0{,}01600829$$

Adapun hasil dari penghitungan nilai bobot *TF-IDF* terhadap seluruh kata yang terdapat di dalam korpus dapat dilihat pada tabel 4.4.

Tabel 4.4 Contoh Hasil Penghitungan Metode *TF-IDF*

Q	*TF*			*IDF*	*W = TF * IDF*		
	D1	D2	D3		D1	D2	D3
patok	0,0909	0	0	0,4771	0,1342	0	0
harga	0,0909	0,1667	0,2	0	0	0	0
tiket	0,0909	0,1667	0,2	0	0	0	0
konser	0,0909	0,1667	0,2	0	0	0	0
mahal	0,0909	0,1667	0	0,1761	0,0160	0,0293	0
banget	0,0909	0	0	0,4771	0,1342	0	0
juta	0,0909	0	0	0,4771	0,1342	0	0
section	0,0909	0	0	0,4771	0,1342	0	0
dapat	0,0909	0	0	0,4771	0,1342	0	0
nomor	0,0909	0	0	0,4771	0,1342	0	0
artis	0,0909	0	0	0,4771	0,1342	0	0
iya	0	0,1667	0	0,4771	0	0,0795	0
juga	0	0,1667	0	0,4771	0	0,0795	0
coldplay	0	0	0,2	0,4771	0	0	0,0954
normal	0	0	0,2	0,4771	0	0	0,0954

Sebagai implementasi dari contoh penghitungan tersebut, peneliti mengetikkan kode perintah seperti yang ditunjukkan oleh kode program 4.17 pada *compiler*.

```
# Menghitung TF
def calc_TF(document):

    # Menghitung jumlah kemunculan kata di di dalam
dokumen
    TF_dict = {}
    for term in document:
        if term in TF_dict:
            TF_dict[term] += 1
        else:
            TF_dict[term] = 1
    # Menghitung TF pada setiap kata
    for term in TF_dict:
        TF_dict[term] = TF_dict[term] /
len(document)
    return TF_dict

# Menyimpan TF Dictionary ke dalam kolom tabel
data["TF_dict"] = data['tweet_list'].apply(calc_TF)
```

```
print('Hasil Penghitungan Nilai TF : \n')
print(data["TF_dict"].head(10))
```

Kode Program 4.17 Menghitung Nilai *TF*

Berdasarkan kode program 4.17, dapat diketahui bahwa untuk menghitung nilai *TF,* peneliti menggunakan sebuah fungsi yang didalamnya terdapat *dictionary* yang menampung setiap kata yang berada di dalam dokumen beserta logika yang digunakan untuk menghitung nilai *TF*. Selanjutnya dilakukan tahap penghitungan dengan mengaplikasikan fungsi pada setiap *term* yang tertampung di dalam *TF Dictionary* yang kemudian akan disimpan dalam kolom *'TF_dict'* pada tabel dengan gambar 4.14 sebagai *output* dari tahap penghitungan nilai *TF*.

```
Hasil Penghitungan Nilai TF :

0    {'wtb': 0.05555555555555555, 'tiket': 0.055555...
1    {'bentar': 0.041666666666666664, 'ych': 0.0416...
2    {'jadi': 0.02631578947368421, 'sih': 0.0789473...
3    {'edisi': 0.0625, 'nyerah': 0.0625, 'wtb': 0.0...
4    {'wts': 0.04878048780487805, 'tiket': 0.073170...
5    {'wts': 0.07692307692307693, 'ready': 0.076923...
6    {'wts': 0.041666666666666664, 'tiket': 0.08333...
7    {'yang': 0.041666666666666664, 'beli': 0.08333...
8    {'cari': 0.045454545454545456, 'tiket': 0.0909...
9    {'temenku': 0.1, 'juga': 0.1, 'suka': 0.1, 'ng...
Name: TF_dict, dtype: object
```

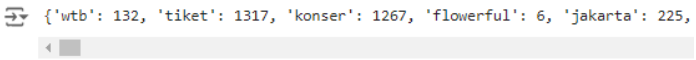
Gambar 4.14 *Output* Hasil Penghitungan Nilai *TF*

Setelah ditemukan nilai *TF* yang tertampung dalam folder *'TF_dict'*, dilanjutkan dengan penghitungan nilai *DF* yang merupakan jumlah kemunculan setiap *term* yang terdapat di dalam korpus. Adapun kode perintah yang dijalankan dapat dilihat pada kode program 4.18.

```
#menghitung  DF
def calc_DF(tfDict):
    count_DF = {}
    # Menjalankan TF Dictionary tiap dokumen
dan menambahkannya ke dalam DF Dictionary
    for document in tfDict:
        for term in document:
            if term in count_DF:
                count_DF[term] += 1
            else:
                count_DF[term] = 1
    return count_DF

# Menyimpan DF Dictionary ke dalam kolom
tabel
DF = calc_DF(data["TF_dict"])
print(DF)
```

Kode Program 4.18 Menghitung Nilai *DF*

Berdasarkan kode program 4.18, dapat diketahui bahwa untuk menentukan nilai *DF*, dilakukan pembuatan fungsi yang didalamnya terdapat perintah untuk menghitung kemunculan setiap *term* yang terdapat di dalam *TF Dictionary* dan menampung nilai *DF* ke dalam *DF Dictionary*. Adapun *output* yang diperoleh dari hasil penghitungan nilai *DF* dapat dilihat pada gambar 4.15.



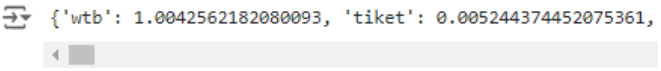
Gambar 4.15 *Output* Hasil Penghitungan Nilai *DF*

Setelah dilakukan penghitungan nilai *DF*, dilanjutkan dengan penghitungan nilai *IDF*. Adapun kode perintah yang digunakan oleh peneliti untuk menghitung nilai *IDF* dapat dilihat pada kode program 4.19.

```
n_document = len(data)
# Menghitung IDF
def calc_IDF(__n_document, __DF):
    IDF_Dict = {}
    for term in __DF:
        IDF_Dict[term] = np.log10 (__n_document /
(__DF[term]))
    return IDF_Dict

# Menjalankan fungsi IDF
IDF = calc_IDF(n_document, DF)
print(IDF)
```

Kode Program 4.19 Menghitung Nilai *IDF*

Berdasarkan kode program 4.19, dapat diketahui bahwa untuk menghitung nilai *IDF*, peneliti menggunakan sebuah fungsi yang di dalamnya terdapat perintah untuk menghitung nilai *IDF* pada setiap *term* yang terdapat di dalam korpus dan menampung nilai *IDF* yang telah diperoleh ke dalam *IDF Dictionary.* Adapun *output* yang diperoleh dari hasil penghitungan nilai *IDF* dapat dilihat pada gambar 4.16.



Gambar 4.16 *Output* Hasil Penghitungan Nilai *IDF*

Setelah diperoleh nilai *IDF,* dilanjutkan dengan melakukan penghitungan nilai bobot dengan menggunakan metode *TF-IDF*. Nilai bobot *TF-IDF* dapat diperoleh dengan mengalikan hasil dari nilai *TF* dengan hasil dari nilai *IDF* pada setiap *term* yang ada di dalam korpus.

```
# Menghitung TF-IDF
def calc_TF_IDF(TF):
    TF_IDF_Dict = {}
    # Perkalian antara TF dan IDF pada tiap kata
    for key in TF:
        TF_IDF_Dict[key] = TF[key] * IDF[key]
    return TF_IDF_Dict

# Menyimpan Hasil TF-IDF Dictionary ke dalam kolom
tabel
data["TF_IDF_dict"] =
data["TF_dict"].apply(calc_TF_IDF)
data["TF_IDF_dict"].head()
```

Kode Program 4.20 Menghitung Nilai Bobot *TF-IDF*

```
0    {'wtb': 0.05579201212266718, 'tiket': 0.000291...
1    {'bentar': 0.0977782874595923, 'ych': 0.130201...
2    {'jadi': 0.024890873740018152, 'sih': 0.079544...
3    {'edisi': 0.17648750960936738, 'nyerah': 0.195...
4    {'wts': 0.03865325314625629, 'tiket': 0.000383...
Name: TF_IDF_dict, dtype: object
```

Gambar 4.17 *Output* Hasil Penghitungan Nilai *TF-IDF*

Berdasarkan kode program 4.20, peneliti menggunakan sebuah fungsi yang di dalamnya terdapat perintah untuk menghitung nilai bobot *TF-IDF* pada setiap *term* yang terdapat di dalam korpus dan menyimpan hasil yang telah diperoleh berupa *TF-IDF Dictionary* ke dalam kolom *'TF_IDF_dict'* pada tabel dengan gambar 4.17 sebagai *output*.

4.3.2 Metode *TF-RF*

Untuk menghitung nilai bobot dengan menggunakan Metode *TF-RF,* terdapat beberapa tahapan yang harus dilakukan. Tahap pertama yaitu menghitung nilai *TF (Term Frequency*) terlebih dahulu dengan contoh penghitungan yang telah ditunjukan oleh tabel 4.3. Setelah didapatkan nilai *TF*, dapat dilanjutkan dengan

menghitung nilai *RF (Relevance Frecuency)* yang diperoleh dengan menggunakan rumus yang terdapat pada persamaan 2.4. Dalam Metode *RF*, relevansi dokumen dapat diketahui dari frekuensi kemunculan *term* di dalam kategori yang berkaitan. Sebagai contoh, berikut merupakan contoh hasil penghitungan nilai bobot *RF* untuk kata "mahal" .

$$W_{(RF)} = log\left(2 + \frac{b}{max\,(1,c)}\right)$$

$$= log\left(2 + \frac{2}{max\,(1,1)}\right)$$

$$= log\left(2 + \frac{2}{2}\right)$$

$$= log\,(2 + 1)$$

$$= log\,(3)$$

$$= 0{,}47712125$$

Setelah diperoleh nilai *RF*, dapat dilanjutkan dengan penghitungan nilai bobot *TF-RF.* Dalam Metode *TF-RF,* nilai bobot dapat ditentukan dengan mempertimbangkan relevansi antar dokumen yang dapat dilihat dari frekuensi kemunculan *term* yang terdapat dalam kategori yang saling berkaitan. Nilai bobot *TF-RF* dapat dihasilkan dengan mengalikan antara nilai bobot *TF* dengan nilai bobot *RF.* Sebagai contoh,

berikut merupakan contoh hasil penghitungan nilai bobot *TF-RF* untuk kata "mahal" pada D1.

$$W_{(TF.RF)} = TF_{(d,t)} \times RF$$
$$= 0{,}09090909 \times 0{,}47712125$$
$$= 0{,}04337465$$

Adapun hasil dari penghitungan nilai bobot *TF-RF* terhadap seluruh kata yang terdapat di dalam korpus dapat dilihat pada tabel 4.5.

Tabel 4.5 Contoh Hasil Penghitungan Metode *TF-RF*

Q	*TF*			*RF*	*W = TF * RF*		
	D1	**D2**	**D3**		**D1**	**D2**	**D3**
patok	0,0909	0	0	0,6532	0,0593	0	0
harga	0,0909	0,1667	0,2	0,4771	0,0434	0,0795	0,0954
tiket	0,0909	0,1667	0,2	0,4771	0,0434	0,0795	0,0954
konser	0,0909	0,1667	0,2	0,4771	0,0434	0,0795	0,0954
mahal	0,0909	0,1667	0	0,5440	0,0160	0,0906	0
banget	0,0909	0	0	0,6532	0,0593	0	0
juta	0,0909	0	0	0,6532	0,0593	0	0
section	0,0909	0	0	0,6532	0,0593	0	0
dapat	0,0909	0	0	0,6532	0,0593	0	0
nomor	0,0909	0	0	0,6532	0,0593	0	0
artis	0,0909	0	0	0,6532	0,0593	0	0
iya	0	0,1667	0	0,6532	0	0,1088	0
juga	0	0,1667	0	0,6532	0	0,1088	0
coldplay	0	0	0,2	0,6532	0	0	0,1306
normal	0	0	0,2	0,6532	0	0	0,1306

Sebagai implementasi dari contoh penghitungan tersebut, peneliti menggunakan beberapa tahapan untuk dapat melakukan pembobotan kata dengan Metode *TF-*

RF. Yang pertama yaitu tahap penghitungan nilai *TF* yang telah disajikan pada kode program 4.17 sehingga dapat dihasilkan *TF Dictionary*. Tahap selanjutnya yaitu menghitung nilai *RF*. Adapun kode perintah yang digunakan oleh peneliti untuk menghitung nilai *RF* dapat dilihat pada kode program 4.21.

```
def calc_RF(tfDict):
    count_RF = {}
    # Menjalankan TF Dictionary tiap dokumen dan
menambahkannya ke dalam RF Dictionary
    for document in tfDict:
        for term in document:
            if term in count_RF:
                count_RF[term] += 1
            else:
                count_RF[term] = 1
    return count_RF

RF = calc_RF(data["TF_dict"])
print(RF)
```

Kode Program 4.21 Pembuatan *RF Dictionary*

Berdasarkan kode program 4.21, terdapat dua langkah yang dilakukan oleh peneliti dalam melakukan penghitungan nilai *RF*. Langkah yang pertama dalam penghitungan nilai *RF* yaitu pembuatan fungsi yang di dalamnya terdapat perintah untuk membuat *dictionary* sementara sebagai tempat penghitungan jumlah kemunculan dari kata tertentu dalam kategori yang berkaitan dengan gambar 4.18 sebagai *output* dari kode program 4.21.

```
{'wtb': 132, 'tiket': 1317, 'konser': 1267, 'flowerful': 6, 'jakarta': 225,
```

Gambar 4.18 *Output* Pembuatan *RF Dictionary*

Setelah didapatkan *dictionary* sementara, dilanjutkan dengan penghitungan nilai *RF* dengan mengetikkan perintah seperti pada kode program 4.22.

```
n_document = len(data)
# Menghitung RF
def calc_RF(__n_document, __RF):
    RF_Dict = {}
    for term in __RF:
        RF_Dict[term] = np.log10 (2 +
((__RF[term]) / max(1, (__n_document -
(__RF[term])))))
    return RF_Dict

# Menjalankan fungsi RF
_RF = calc_RF(n_document, RF)
print(_RF)
```

Kode Program 4.22 Pengitungan Nilai *RF*

Berdasarkan kode program 4.22, peneliti menggunakan sebuah fungsi yang di dalamnya terdapat logika yang digunakan untuk menjalankan tahap penghitungan nilai *RF* pada setiap *term* yang terdapat di dalam dokumen yang kemudian menyimpannya ke dalam *RF Dictionary.* Adapun *output* dari pembuatan *RF Dictionary* dapat dilihat pada gambar 4.19.

```
{'wtb': 0.3242636031445165, 'tiket': 1.9258919670159795,
```

Gambar 4.19 *Output* Pengitungan Nilai *RF*

Setelah diperoleh nilai *RF,* dilanjutkan dengan melakukan penghitungan nilai bobot dengan menggunakan metode *TF-RF*. Nilai bobot *TF-RF* dapat diperoleh dengan mengalikan hasil dari nilai *TF* dengan

hasil dari nilai *RF* pada setiap *term* yang ada di dalam korpus. Adapun kode perintah yang digunakan oleh peneliti untuk menghitung nilai bobot dengan menggunakan Metode *TF-RF* dapat dilihat pada kode program 4.23.

```
# Menghitung TF-RF
def calc_TF_RF(TF):
    TF_RF_Dict = {}
    # Perkalian antara TF dan RF pada tiap kata
    for key in TF:
        TF_RF_Dict[key] = TF[key] * _RF[key]
    return TF_RF_Dict

# Menyimpan Hasil TF-RF Dictionary ke dalam kolom tabel
data["TF_RF_dict"] = data["TF_dict"].apply(calc_TF_RF)
data["TF_RF_dict"].head()
```

Kode Program 4.23 Penghitungan Nilai Bobot *TF-RF*

Berdasarkan kode program 4.23, peneliti menggunakan sebuah fungsi yang di dalamnya terdapat perintah untuk menghitung nilai bobot *TF-RF* pada setiap *term* yang terdapat di dalam korpus dan menyimpan hasil yang telah diperoleh berupa *TF-RF Dictionary* ke dalam kolom *'TF_RF_dict'* pada tabel dengan gambar 4.20 sebagai *output*.

```
0    {'wtb': 0.018014644619139805, 'tiket': 0.10699...
1    {'bentar': 0.01258377974027631, 'ych': 0.01254...
2    {'jadi': 0.00862948718017657, 'sih': 0.0255847...
3    {'edisi': 0.01883476031397427, 'nyerah': 0.018...
4    {'wts': 0.016629341839106023, 'tiket': 0.14091...
Name: TF_RF_dict, dtype: object
```

Gambar 4.20 *Output* Hasil Penghitungan Nilai *TF-RF*

Setelah *DataFrame* melalui tahap pembobotan kata, dilanjutkan dengan tahap pembentukan vektor kata yang

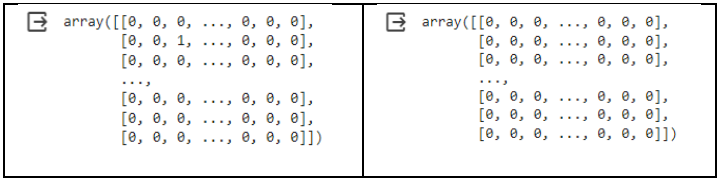
merupakan tahap penggabungan pada setiap *DataFrame* hasil dari tahap pembobotan kata menjadi kumpulan *array* yang kemudian diubah ke dalam bentuk matriks, dimana setiap baris yang terdapat di dalam matriks menunjukkan baris pada dokumen, dan setiap kolom yang terdapat di dalam matriks menunjukkan semua kata yang ada di dalam dokumen teks secara keseluruhan. Untuk dapat menjalankan tahap pembentukan vektor kata, peneliti memanfaatkan modul *CountVectorizer* yang berasal dari *library scikit-learn*. Adapun kode perintah yang digunakan oleh peneliti untuk menjalankan tahap pembentukan vektor kata dapat dilihat pada kode program 4.24 dan 4.25 dengan gambar 4.21 sebagai *output*.

```
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer()
TF_IDF_Vec = cv.fit_transform(data['TF_IDF_dict'])
TF_IDF_Vec = TF_IDF_Vec.toarray()
TF_IDF_Vec
```

Kode Program 4.24 Pembentukan Vektor Kata *TF-IDF*

```
TF_RF_Vec = cv.fit_transform(data['TF_RF_dict'])
TF_RF_Vec = TF_RF_Vec.toarray()
TF_RF_Vec
```

Kode Program 4.25 Pembentukan Vektor Kata *TF-RF*



Gambar 4.21 *Output* Vektor Kata *TF-IDF* dan *TF-RF*

4.4 Pembagian *Dataset*

Setelah *DataFrame* melalui tahap ekstraksi fitur, hasil dari pembobotan masing-masing metode akan dibagi menjadi data latih dan data uji yang akan digunakan dalam model klasifikasi dengan perbandingan rasio 70:30, dengan 70% sebagai data latih dan 30% sebagai data uji.

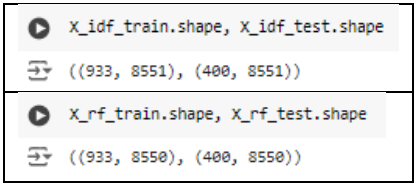
Untuk menjalankan tahap pembagian *dataset*, peneliti memanfaatkan modul *train_test_split* yang berasal dari *library scikit-learn*. Adapun kode perintah yang digunakan oleh peneliti untuk menjalankan tahap pembagian *dataset* pada masing-masing metode pembobotan kata dapat dilihat pada kode program 4.26.

```
from     sklearn.model_selection
import train_test_split

X_idf = TF_IDF_Vec
y_idf = data['label']
X_idf_train, X_idf_test,
y_idf_train, y_idf_test =
train_test_split(X_idf, y_idf,
test_size=0.30,random_state=35)
```

```
from     sklearn.model_selection
import train_test_split

X_rf = TF_RF_Vec
y_rf = data['label']
X_rf_train, X_rf_test,
y_rf_train, y_rf_test =
train_test_split(X_rf, y_rf,
test_size=0.30,random_state=35)
```

Kode Program 4.26 Pembagian *dataset TF-IDF* dan *TF-RF*

Berdasarkan kode program 4.27, dapat diperoleh data latih sebanyak 933 *tweet* dan data uji sebanyak 400 *tweet* untuk masing-masing metode pembobotan kata seperti yang disajikan pada gambar 4.22.



Gambar 4.22 Jumlah Data Latih dan Data Uji

4.5 Perancangan Model

Pada tahap perancangan model, *DataFrame* yang telah dibagi menjadi data latih dan data uji akan digunakan sebagai *input* dalam tahap pelatihan model, uji coba model, serta evaluasi model analisis sentimen dengan memanfaatkan Algoritma *Naïve Bayes*. Untuk menyiapkan *DataFrame* hasil dari tahap ekstraksi fitur agar dapat digunakan pada tahap perancangan model, peneliti memanfaatkan beberapa *library Python*, diantaranya yaitu *library NumPy* untuk keperluan operasi matematika dalam *compiler*, *library Pandas* sebagai *library Python* yang berfungsi untuk memanipulasi dan analisis data. Selain itu, peneliti juga memanfaatkan *Google Drive* untuk menghubungkan *file* yang berisi *DataFrame* dengan *compiler*.

```
import pandas as pd
import numpy as np

# membaca dan menampilkan file (Koneksi dengan
GDrive)
from google.colab import drive
drive.mount('/content/drive')
```

Kode Program 4.27 *Library* yang digunakan pada Tahap Perancangan Model

Adapun kode perintah yang digunakan oleh peneliti untuk menyiapkan *DataFrame* agar dapat digunakan dalam tahap pelatihan model dapat dilihat pada kode program 4.28 dengan gambar 4.23 Sebagai *output*.

```
data=pd.read_csv
("/content/drive/MyDrive/Skripsi_Muhamm
ad_Amirul_S/ekstraksi_fitur/Ekstraksi_F
itur.csv", usecols =["label", "tweet",
'TF_IDF_dict", "TF_RF_dict"])

data.head(10)
```

Kode Program 4.28 Menyiapkan *DataFrame*

	label	tweet	TF_IDF_dict	TF_RF_dict
0	Positif	['wtb', 'tiket', 'konser', 'flowerful', 'jakar...	{'wtb': 0.05579201212266718, 'tiket': 0.000291...	{'wtb': 0.018014644619139805, 'tiket': 0.10699...
1	Negatif	['bentar', 'ych', 'jadi', 'begini', 'kalo', 't...	{'bentar': 0.0977782874595923, 'ych': 0.130201...	{'bentar': 0.012583779740276 31, 'ych': 0.01254...
2	Negatif	['jadi', 'sih', 'beli', 'tiket', 'calo', 'pas'...	{'jadi': 0.024890873740018152, 'sih': 0.079544...	{'jadi': 0.008629487180176 57, 'sih': 0.0255847...
3	Positif	['edisi', 'nyerah', 'wtb', 'tiket', 'konser', ...	{'edisi': 0.17648750960936738, 'nyerah': 0.195...	{'edisi': 0.018834760313974 27, 'nyerah': 0.018...
4	Positif	['wts', 'tiket', 'twice', 'ready', 'tapi', 'be...	{'wts': 0.03865325314625629, 'tiket': 0.000383...	{'wts': 0.016629341839106023, 'tiket': 0.14091...
5	Positif	['wts', 'ready', 'tiket', 'festival', 'coldpla...	{'wts': 0.060953206884481075, 'ready': 0.12787...	{'wts': 0.02622319290012873, 'ready': 0.023525...
6	Positif	['wts', 'tiket', 'konser', 'coldplay', 'singap...	{'wts': 0.03301632039576058, 'tiket': 0.000437...	{'wts': 0.014204229487569728, 'tiket': 0.16049...
7	Negatif	['yang', 'beli', 'harga', 'tiket', 'gila', 'ja...	{'yang': 0.025428433198065 55, 'beli': 0.071082...	{'yang': 0.015268010045385184, 'beli': 0.02792...
8	Positif	['cari', 'tiket', 'konser', 'anniv', 'jeket', ...	{'cari': 0.07129671130211691, 'tiket': 0.00047...	{'cari': 0.013955262382076264, 'tiket': 0.1750...
9	Negatif	['temenku', 'juga', 'suka', 'ngome', 'harga', ...	{'temenku': 0.23466788990302156, 'juga': 0.131...	{'temenku': 0.030201071376663148, 'juga': 0.03...

Gambar 4.23 *DataFrame* yang akan digunakan pada Tahap Perancangan Model

4.4.1 Algoritma *Naïve Bayes*

Setelah *DataFrame* melalui tahap pembagian *dataset*, data latih yang berisi hasil bobot kata dari masing-masing metode pembobotan kata akan digunakan sebagai *input* dalam tahap pelatihan model klasifikasi dengan memanfaatkan Algoritma *Naïve Bayes*. Untuk dapat menjalankan tahap pelatihan model dengan maksimal, peneliti memanfaatkan *library scikit-learn* untuk menjalankan proses klasifikasi model. Adapun kode perintah yang digunakan oleh peneliti

untuk menjalankan tahap pelatihan model dapat dilihat pada kode program 4.29.

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
CNB = MultinomialNB()
CNB.fit(X_idf_train, y_idf_train)
```

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix
CNB = MultinomialNB()
CNB.fit(X_rf_train, y_rf_train)
```

Kode Program 4.29 Tahap Pelatihan Model

Berdasarkan kode program 4.29, peneliti menggunakan beberapa modul yang diperlukan pada tahap pelatihan, pengujian, serta evaluasi model. Diantaranya yaitu modul *MultinomialNB* untuk menjalankan Algoritma *Multinomial Naïve Bayes* yang merupakan salah satu metode klasifikasi yang menggunakan nilai probabilitas suatu kelas dalam suatu dokumen. Selain itu, peneliti juga menggunakan modul *classification_report* untuk menampilkan hasil dari klasifikasi sentimen, serta modul *confusion_matrix* untuk menampilan hasil dari *Confussion Matrix*. Untuk menjalankan tahap pelatihan model, peneliti menggunakan *method MultinomialNB()* yang berasal dari modul *MultinomialNB* terhadap data latih dari masing-masing metode pembobotan kata sebesar 70% dari keseluruhan *DataFrame*.

4.4.2 Uji Model

Setelah melalui tahap pelatihan model, data uji yang telah diperoleh dari masing-masing pembobotan kata akan diujikan untuk mengetahui performa dari model klasifikasi yang telah dirancang. Adapun kode perintah yang digunakan oleh peneliti untuk menjalankan tahap uji model dapat dilihat pada kode program 4.30.

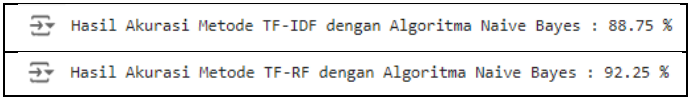
```
from sklearn import metrics

predicted = CNB.predict(X_idf_test)
accuracy_score = metrics.accuracy_score(predicted,
y_idf_test)

print('Hasil Akurasi Metode TF-IDF dengan
Algoritma Naive Bayes
:',str('{:04.2f}'.format(accuracy_score*100))+'
%')
```

```
from sklearn import metrics

predicted = CNB.predict(X_rf_test)
accuracy_score = metrics.accuracy_score(predicted,
y_rf_test)

print('Hasil Akurasi Metode TF-RF dengan Algoritma
Naive Bayes
:',str('{:04.2f}'.format(accuracy_score*100))+'
%')
```

Kode Program 4.30 Kode Perintah Tahap Uji Model

Berdasarkan kode program 4.30, peneliti menggunakan modul *metrics* yang diperlukan untuk menghitung nilai akurasi yang diperoleh dari tahap uji model klasifikasi. Untuk menjalankan tahap uji model, peneliti menggunakan *method predict* untuk memprediksi nilai *output* yang dihasilkan oleh data uji.

Setelah itu dilanjutkan dengan menghitung nilai akurasi dari model klasifikasi yang telah dirancang. Adapun *output* yang dihasilkan dari tahap uji model dapat dilihat pada gambar 4.24.



Gambar 4.24 *Output* Tahap Uji Model

Berdasarkan gambar 4.24, dapat diketahui bahwa nilai akurasi yang diperoleh data uji hasil dari pembobotan kata dengan menggunakan Metode *TF-RF* lebih tinggi dibandingan dengan Metode *TF-IDF,* yaitu **92.25%** untuk Metode *TF-RF* dan **88.75%** untuk Metode *TF-IDF*. Namun untuk mengetahui performa keseluruhan dari model klasifikasi yang telah dirancang, diperlukan tahap evaluasi model.

4.4.3 Evaluasi Model

Setelah diperoleh nilai akurasi dari model klasifikasi pada tahap uji model, dilanjutkan dengan mengevaluasi model klasifikasi dengan *Confusion Matrix* untuk mengetahui performa keseluruhan dari model yang telah dirancang berupa nilai akurasi, presisi, *recall,* dan *f1-score*. Berdasarkan hasil dari klasifikasi yang telah dilakukan pada tahap uji model, dapat dibentuk sebuah tabel *Confusion Matrix* dari

masing-masing metode yang ditunjukkan oleh tabel 4.6 dan 4.7.

Tabel 4.6 Hasil *Confusion Matrix TF-IDF*

		Nilai Aktual	
		Positif	***Negatif***
Nilai PrediKsi	***Positif***	117	22
	Negatif	23	238

Tabel 4.7 Hasil *Confusion Matrix TF-RF*

		Nilai Aktual	
		Positif	***Negatif***
Nilai PrediKsi	***Positif***	126	10
	Negatif	21	243

Setelah tabel *Confusion Matrix* terbentuk, peneliti melakukan penghitungan performa dari model analisis sentimen kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia secara manual untuk masing-masing metode pembobotan kata yang terdiri atas beberapa tahap. Yang pertama yaitu tahap penghitungan nilai akurasi yang merupakan sebuah metrik evaluasi yang menunjukkan seberapa baik model klasifikasi membuat prediksi yang benar dari semua prediksi yang telah dilakukan. Berikut merupakan hasil penghitungan nilai akurasi dari

model klasifikasi yang telah dirancang ditunjukkan oleh tabel 4.8.

Tabel 4.8 Hasil Penghitungan Nilai Akurasi

Akurasi	
TF-IDF	***TF-RF***
$\frac{TP + TN}{TP + FP + FN + TN} \times 100$	$\frac{TP + TN}{TP + FP + FN + TN} \times 100$
$\frac{117 + 238}{117 + 22 + 23 + 238} \times 100$	$\frac{126 + 243}{126 + 10 + 21 + 243} \times 100$
$\frac{355}{400} \times 100$	$\frac{369}{400} \times 100$
88,75%	**92,25%**

Berdasarkan hasil nilai akurasi yang disajikan pada tabel 4.8, dapat diketahui bahwa tingkat kemampuan model analisis sentimen dalam mengklasifikasikan sentimen mengenai fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia dengan benar menggunakan data uji hasil dari pembobotan Metode *TF-RF* memiliki persentase yang lebih tinggi jika dibandingan dengan Metode *TF-IDF,* yaitu **92.25%** untuk Metode *TF-RF* dan **88.75%** untuk Metode *TF-IDF.*

Setelah didapatkan nilai akurasi pada model klasifikasi, langkah selanjutnya yaitu tahap penghitungan nilai presisi untuk masing-masing kelas sentimen. Berikut merupakan hasil penghitungan nilai

presisi dari model klasifikasi yang telah dirancang ditunjukkan oleh tabel 4.9 dan 4.10.

Tabel 4.9 Hasil Penghitungan Nilai Presisi Kelas Positif

Presisi (Kelas Positif)	
TF-IDF	***TF-RF***
$\frac{TP}{TP + FP} \times 100$	$\frac{TP}{TP + FP} \times 100$
$\frac{117}{117 + 22} \times 100$	$\frac{126}{126 + 10} \times 100$
$\frac{117}{139} \times 100$	$\frac{126}{136} \times 100$
84,17%	**92,64%**

Tabel 4.10 Hasil Penghitungan Nilai Presisi Kelas Negatif

Presisi (Kelas Negatif)	
TF-IDF	***TF-RF***
$\frac{TN}{TN + FN} \times 100$	$\frac{TN}{TN + FN} \times 100$
$\frac{238}{238 + 23} \times 100$	$\frac{243}{243 + 21} \times 100$
$\frac{238}{261} \times 100$	$\frac{243}{264} \times 100$
91,18%	**92,04%**

Berdasarkan hasil nilai presisi yang disajikan pada tabel 4.9 dan 4.10, dapat diketahui bahwa tingkat kemampuan model analisis sentimen dalam menggambarkan keakuratan antara data yang diminta

dengan hasil prediksi menggunakan data uji yang berasal dari Metode *TF-RF* memiliki persentase yang lebih tinggi jika dibandingkan dengan Metode *TF-IDF* dengan perbandingan untuk kelas sentimen positif sebesar **92,64%** pada Metode *TF-RF* berbanding **84,17%** pada Metode *TF-IDF*, dan untuk kelas sentimen negatif sebesar **92,04%** pada Metode *TF-RF* berbanding **91,18%** pada Metode *TF-IDF*.

Setelah didapatkan nilai presisi pada model klasifikasi, langkah selanjutnya yaitu tahap penghitungan nilai *recall* untuk masing-masing kelas sentimen. Berikut merupakan hasil penghitungan nilai *recall* dari model klasifikasi yang telah dirancang ditunjukkan oleh tabel 4.11 dan 4.12.

Tabel 4.11 Hasil Penghitungan Nilai *Recall* Kelas Positif

***Recall* (Kelas Positif)**	
TF-IDF	***TF-RF***
$\frac{TP}{TP + FN} \times 100$	$\frac{TP}{TP + FN} \times 100$
$\frac{117}{117 + 23} \times 100$	$\frac{126}{126 + 21} \times 100$
$\frac{117}{140} \times 100$	$\frac{126}{147} \times 100$
83,57%	**85,57%**

Tabel 4.12 Hasil Penghitungan Nilai *Recall* Kelas Negatif

***Recall* (Kelas Negatif)**	
TF-IDF	***TF-RF***
$\frac{TN}{TN+FP} \times 100$	$\frac{TN}{TN+FP} \times 100$
$\frac{238}{238+22} \times 100$	$\frac{243}{243+10} \times 100$
$\frac{238}{260} \times 100$	$\frac{243}{253} \times 100$
91,53%	**96,04%**

Berdasarkan hasil nilai *recall* yang disajikan pada tabel 4.11 dan 4.12, dapat diketahui bahwa tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi dengan menggunakan data uji yang berasal dari Metode *TF-RF* memiliki persentase yang lebih tinggi jika dibandingkan dengan Metode *TF-IDF* dengan perbandingan untuk kelas sentimen positif sebesar **85,57%** pada Metode *TF-RF* berbanding **83,57%** pada Metode *TF-IDF*, dan untuk kelas sentimen negatif sebesar **96,04%** pada Metode *TF-RF* berbanding **91,53%** pada Metode *TF-IDF*.

Setelah didapatkan nilai *recall* pada model klasifikasi, tahap evaluasi model dapat diakhiri dengan penghitungan nilai *f1-score* untuk mengukur kemampuan model analisis sentimen yang telah

dirancang dalam menyeimbangkan antara nilai presisi dan nilai *recall* pada masing-masing kelas sentimen. Berikut merupakan hasil penghitungan nilai *f1-score* dari model klasifikasi yang telah dirancang ditunjukkan oleh tabel 4.13 dan 4.14.

Tabel 4.13 Hasil Penghitungan Nilai *F1-Score* Kelas Positif

***F1-Score* (Kelas Positif)**	
TF-IDF	***TF-RF***
$2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$	$2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$
$2 \times \frac{(83{,}57 \times 84{,}17)}{(83{,}57 + 84{,}17)}$	$2 \times \frac{(85{,}57 \times 92{,}64)}{(85{,}57 + 92{,}64)}$
$2 \times \frac{7.034{,}0869}{167{,}74}$	$2 \times \frac{7.927{,}2048}{178{,}21}$
83,86%	**88,96%**

Tabel 4.14 Hasil Penghitungan Nilai *F1-Score* Kelas Negatif

***F1-Score* (Kelas Negatif)**	
TF-IDF	***TF-RF***
$2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$	$2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$
$2 \times \frac{(91{,}53 \times 91{,}18)}{(91{,}53 + 91{,}18)}$	$2 \times \frac{(96{,}04 \times 92{,}04)}{(96{,}04 + 92{,}04)}$
$2 \times \frac{8.345{,}7054}{182{,}71}$	$2 \times \frac{8.839{,}5216}{188{,}08}$
91,35%	**93,99%**

Selain melakukan penghitungan nilai presisi dan *recall*, diperlukan penghitungan nilai *f1-score* untuk mengukur kemampuan model analisis sentimen yang telah dirancang dalam menyeimbangkan antara nilai presisi dan nilai *recall*. Berdasarkan hasil nilai *recall* pada tabel 4.13 dan 4.14, dapat diketahui bahwa nilai *f1-score* yang berasal dari data uji menggunakan Metode *TF-RF* memiliki persentase yang lebih tinggi jika dibandingkan dengan Metode *TF-IDF* dengan perbandingan untuk kelas sentimen positif sebesar **88,96%** pada Metode *TF-RF* berbanding **83,86%** pada Metode *TF-IDF*, dan untuk kelas sentimen negatif sebesar **93,99%** pada Metode *TF-RF* berbanding **91,35%** pada Metode *TF-IDF*

Adapun kode perintah yang digunakan oleh peneliti sebagai bentuk implementasi tahap evaluasi terhadap model klasifikasi yang telah dirancang dari masing-masing metode pembobtan kata dapat dilihat pada kode program 4.31 dengan gambar 4.25 sebagai *output*.

```
print('-----------------------------------------------
-')
print('Confusion Matrix :')
print(pd.DataFrame(confusion_matrix(y_idf_test,
predicted)))
print('-----------------------------------------------
-')
print('Hasil Klasifikasi :')
print(classification_report(y_idf_test, predicted))
```

```
print('-----------------------------------------------
-')
print('Confusion Matrix :')
print(pd.DataFrame(confusion_matrix(y_rf_test,
predicted)))
print('-----------------------------------------------
-')
print('Hasil Klasifikasi :')
print(classification_report(y_rf_test, predicted))
```

Kode Program 4.31 Tahap Evaluasi Model

```
-------------------------------------------------
Confusion Matrix :
     0    1
0  238   22
1   23  117
-------------------------------------------------
Hasil Klasifikasi (TF-IDF) :
              precision    recall  f1-score   support

     Negatif       0.91      0.92      0.91       260
     Positif       0.84      0.84      0.84       140

    accuracy                           0.89       400
   macro avg       0.88      0.88      0.88       400
weighted avg       0.89      0.89      0.89       400
```

```
-------------------------------------------------
Confusion Matrix :
     0    1
0  243   10
1   21  126
-------------------------------------------------
Hasil Klasifikasi (TF-RF) :
              precision    recall  f1-score   support

     Negatif       0.92      0.96      0.94       253
     Positif       0.93      0.86      0.89       147

    accuracy                           0.92       400
   macro avg       0.92      0.91      0.92       400
weighted avg       0.92      0.92      0.92       400
```

Gambar 4.25 *Output* Tahap Evaluasi Model

Berdasarkan kode perintah yang dijalankan pada kode program 4.32 untuk menjalankan tahap evaluasi model, diperoleh hasil *output* berupa gambar 4.25. Adapun tabel 4.15 merupakan hasil evaluasi model analisis sentimen dengan menggunakan data uji yang berasal dari masing-masing metode pembobotan kata.

Tabel 4.15 Hasil Tahap Evaluasi Model

	TF-IDF			***TF-RF***		
Kelas Sentimen	**Presisi**	***Recall***	***F1-Score***	**Presisi**	***Recall***	***F1-Score***
Positif	84,17%	83,57%	83,86%	92,64%	85,57%	88,96%
Negatif	91,18%	91,53%	92,24%	92,04%	96,04%	93,99%

Berdasarkan hasil nilai *presisi, recall*, dan *f1-score* yang telah disajikan pada tabel 4.15, dapat diketahui bahwa masing-masing model analisis sentimen memiliki performa yang sangat baik. Hal ini dikarenakan baik nilai presisi maupun *recall* yang dimiliki oleh masing-masing model mempunyai persentase yang tinggi dan mendekati nilai ideal dari suatu model yaitu 1 (100%) (Surampudi, 2022). Secara keseluruhan, hasil evaluasi model analisis sentimen dengan menggunakan data uji yang berasal dari hasil pembobotan kata Metode *TF-RF* terbukti memiliki performa yang lebih baik jika dibandingkan dengan

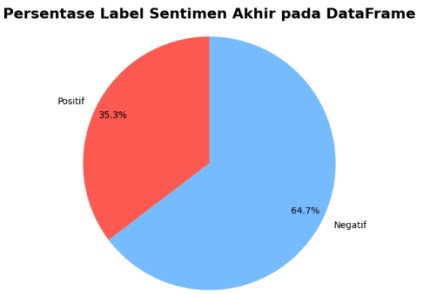
data uji yang berasal dari hasil pembobotan kata Metode *TF-IDF*.

4.6 Visualisasi Model

Setelah dilakukan tahap evaluasi model, penelitian ini dapat diakhiri dengan memvisualisasikan hasil dari model analisis sentimen kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia ke dalam bentuk grafik dan *WordCloud* sehingga menjadi lebih mudah untuk dipahami. Untuk memastikan kembali hasil dari distribusi label sentimen akhir pada *DataFrame*, peneliti menggunakan perintah untuk menampilkannya ke dalam bentuk teks dan grafik *pie* yang ditunjukkan oleh gambar 4.26 dan gambar 4.27.

```
Distribusi Label Sentimen Akhir :
label
Negatif    862
Positif    471
Name: count, dtype: int64
```

Gambar 4.26 Hasil Distribusi Label Sentimen Akhir



Gambar 4.27 Persentase Label Sentimen Akhir

Berdasarkan gambar 4.26, dapat diketahui bahwa distribusi label untuk sentimen "Negatif" sebanyak 862 *tweet*, sedangkan untuk label sentimen "Positif" sebanyak 471 *tweet*. Sehingga ketika dijumlahkan terdapat sebanyak 1333 *DataFrame*. Selain itu, berdasarkan gambar 4.27 dapat diketahui bahwa perbandingan persentase dari masing-masing label sentimen akhir yaitu label sentimen "Positif" sebanyak 35,3% dan label sentimen "Negatif" sebanyak 64,7% dari keseluruhan *DataFrame* yang terdapat di dalam korpus. Oleh karena itu berdasarkan gambar 4.25 dan gambar 4.26 dapat diketahui bahwa pola sentimen yang tercermin dalam data *tweet* yang diunggah di media sosial *X* memiliki kecenderungan yang negatif dan kontra terkait adanya fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia.

Selain menggunakan grafik untuk mengetahui hasil distribusi label sentimen akhir pada *DataFrame*, peneliti juga menggunakan *WordCloud* untuk mengetahui kata-kata yang sering digunakan oleh pengguna media sosial *X* saat mengekspresikan opini mereka mengenai fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia yang ditunjukkan oleh gambar 4.28.

Gambar 4.28 *WordCloud Tweet* Kenaikan Harga Tiket Konser

Berdasarkan gambar 4.28, dapat diketahui bahwa diantara berbagai kata yang muncul dalam *WordCloud*, terdapat beberapa kata yang dicetak lebih besar seperti "harga", "tiket", dan "konser" merupakan kata yang sering digunakan oleh pengguna media sosial X saat mengekspresikan opini mereka mengenai fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia. Selain itu, peneliti juga menggunakan *WordCloud* untuk mengetahui kata-kata yang sering digunakan oleh pengguna media sosial X pada masing-masing label sentimen yang ditunjukkan oleh gambar 4.29.

Gambar 4.29 *WordCloud Tweet* Positif dan Negatif

Berdasarkan gambar 4.28, dapat diketahui bahwa pada *WordCloud Tweet* dengan label sentimen "Positif" menunjukkan sikap antusiasme para pengguna media sosial X untuk tetap membeli tiket konser meskipun terjadi kenaikan harga yang sangat signifikan. Selain itu, dapat diketahui juga bahwa pada *WordCloud Tweet* dengan label sentimen "Negatif" menunjukkan sikap tidak setuju para pengguna media sosial X terhadap adanya kenaikan harga tiket konser yang signifikan.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, diperoleh beberapa hal yang dapat disimpulkan, antara lain:

1. Dari penelitian ini dapat diketahui bahwa pola sentimen yang tercermin dalam data *tweet* yang diunggah di media sosial *X* memiliki kecenderungan negatif dan kontra terkait adanya fenomena kenaikan harga tiket konser pasca pandemi *Covid-19* di Indonesia. Hal ini dapat dibuktikan dengan lebih banyaknya jumlah data *tweet* yang memiliki label sentimen negatif daripada label sentimen positif, yaitu sebanyak 862 data *tweet* pada label sentimen negatif dengan persentase sebesar 64,7% berbanding 471 data *tweet* pada label sentimen positif dengan persentase sebesar 35,3% dari 1333 *DataFrame* yang telah melalui tahap *text-preprocessing*. Selain itu hasil dari tahap evaluasi model juga menunjukkan hasil yang serupa. Pada data uji yang diperoleh dari hasil pembobotan kata dengan memanfaatkan Metode *TF-IDF* dan Metode *TF-RF* menghasilkan nilai presisi, *recall,* serta *f1-score*

yang lebih tinggi untuk kelas sentimen negatif jika dibandingkan dengan kelas sentimen positif.

2. Pada penelitian ini, hasil evaluasi dari model analisis sentimen yang telah dirancang menunjukan bahwa masing-masing model memiliki performa yang sangat baik. Berdasarkan hasil penelitian yang telah dilakukan oleh peneliti, Metode *TF-RF* terbukti memiliki performa yang lebih baik jika dibandingkan dengan Metode *TF-IDF* pada tahap ekstraksi fitur. Hal ini dikarenakan pada tahap perancangan model analisis sentimen berbasis Algoritma *Naïve Bayes*, *dataset* yang diperoleh dari hasil pembobotan kata dengan menggunakan Metode *TF-RF* dapat menghasilkan nilai akurasi yang lebih tinggi jika dibandingkan dengan Metode *TF-IDF*, yaitu sebesar **92,25%** pada Metode *TF-RF* berbanding dengan **88.75%** pada Metode T*F-IDF.* Selain itu, berdasarkan hasil nilai *presisi, recall*, dan *f1-score* yang telah diperoleh pada tahap evaluasi model dengan menggunakan *Confusion Matrix*, data uji yang berasal dari hasil pembobotan kata Metode *TF-RF* secara keseluruhan terbukti memiliki persentase yang lebih tinggi jika dibandingkan dengan data uji yang berasal dari hasil pembobotan kata Metode *TF-IDF*.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, peneliti memiliki beberapa saran dan masukan yang dapat dijadikan sebagai bahan pertimbangan pada penelitian selanjutnyam diantaranya sebagai berikut :

1. *Dataset* yang digunakan oleh peneliti pada penelitian ini bersumber dari media sosial *X* yang telah banyak digunakan oleh para peneliti sebelumnya. Alangkah baiknya untuk penelitian selanjutnya dapat menggunakan *dataset* yang berasal dari media sosial ataupun aplikasi yang lainnya.
2. Peneliti berharap untuk penelitian selanjutnya dapat menggunakan perbandingan performa antara metode pembobotan kata *supervised* dan *unsupervised* lainnya sehingga hasil uji yang telah diperoleh dapat dibandingkan dengan penelitian ini untuk menemukan metode pembobotan kata yang optimal baik dalam analisis sentimen maupun kasus yang lainnya.

DAFTAR PUSTAKA

Al Ghofany, M. S., Dwiyansaputra, R., Bimantoro, F., & Khairunnas. (2022). *Indonesian SMS Spam Detection Using TF-RF Feature Weighting Method and Support Vector Machine Classifier. Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science)*, 117–129. https://doi.org/10.2991/978-94-6463-084-8_12

Alshehri, A., & Algarni, A. (2023). TF-TDA: *A Novel Supervised Term Weighting Scheme for Sentiment Analysis. Electronics (Switzerland)*, *12*(7). https://doi.org/10.3390/electronics12071632

Anam, S., Nashihin, H., Taufik, A., Mubarok, Sitompul, H. S., Manik, Y. M., Suparto, Arsid, I., Jumini, S., Nurhab, M. I., Solehudin, W, N. E., & Luturmas, Y. (2023). *Metode Penelitian (Kualitatif, Kuantitatif, Eksperimen, dan R&D)* (S. Anam, Ed.; 1st ed.). PT GLOBAL EKSEKUTIF TEKNOLOGI.

Arthana, R. (2019). *Mengenal Accuracy, Precision, Recall dan Specificity serta yang diprioritaskan dalam Machine Learning*. Medium. https://rey1024.medium.com/mengenal-accuracy-precission-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8

Arunadevi, J., Ramya, S., & Raja, M. R. (2018). *A Study of Classification Algorithms using Rapidminer. International Journal of Pure and Applied Mathematics*, *119*(12). https://www.ijpam.eu/

Assidyk, A. N., Setiawan, E. B., & Kurniawan, I. (2020). *Analisis Perbandingan Pembobotan TF-IDF dan TF-RF pada Trending Topic di Twitter dengan Menggunakan*

Klasifikasi K-Nearest Neighbor. *E-Proceeding of Engineering*, *7*(2).

Batubara, N. F. (2023). *Harga Tiket Selangit, Konser Musik Perparah Tekanan Inflasi*. Tirto.Id. https://tirto.id/harga-tiket-selangit-konser-musik-perparah-tekanan-inflasi-gMHa

BPPB Kemdikbudristek. (2016). *Konser*. KBBI (Kamus Besar Bahasa Indonesia). https://www.kbbi.web.id/konser

Carvalho, F., & Guedes, G. P. (2020). *TF-IDFC-RF: A Novel Supervised Term Weighting Scheme*. http://arxiv.org/abs/2003.07193

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

Deolika, A., & Taufiq Luthfi, E. (2019). Analisis Pembobotan Kata pada Text Mining. *Jurnal Teknologi Informasi*, *3*(2).

Derajad Wijaya, H., & Dwiasnati, S. (2020). *Implementasi Data Mining dengan Algoritma Naïve Bayes pada Penjualan Obat. JURNAL INFORMATIKA*, *7*(1). http://ejournal.bsi.ac.id/ejurnal/index.php/ji

Dwi Wanti, A. (2023). *Analisis Sentimen menggunakan Kartu Prakerja menggunakan Metode Naive Bayes*. UIN Maulana Malik Ibrahim Malang.

Harahap, I. (2022). *Analisa Sentimen Masyarakat Terhadap Kondisi New Normal Pasca Pembatasan Sosial Berskala Besar Akibat Covid 19 Menerapkan Metode Term Frequency-Relevan Frequency. Informasi Dan Teknologi Ilmiah (INTI)*, *9*(2).

Hidayatullah, R. (2021). *Komunikasi Musikal dalam Konser "Musik Untuk Republik." Tonika: Jurnal Penelitian Dan Pengkajian Seni*, *4*(2), 145–160. https://doi.org/10.37368/tonika.v4i2.254

Hikmawan, S., Pardamean, A., & Nur Khasanah, S. (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

Humas Kemensetneg. (2023). *Pemerintah Putuskan Indonesia Masuki Masa Endemi*. KEMENTERIAN SEKRETARIAT NEGARA REPUBLIK INDONESIA. https://setneg.go.id/baca/index/pemerintah_putuskan_indonesia_masuki_masa_endemi#:~:text=Presiden%20Joko%20Widodo%20mengumumkan%20bahwa,dari%20masa%20pandemi%20menjadi%20endemi.

Imron. Ali. (2019). *Analisis Sentimen terhadap Tempat Wisata di Kabupaten Rembang menggunakan Metode Naive Bayes Classifier*. Universitas Islam Indonesia.

Ivanova, I. (2023). *Twitter is now X. Here's What That Means*. CBS NEWS. https://www.cbsnews.com/news/twitter-rebrand-x-name-change-elon-musk-what-it-means/

Jiang, Z., Gao, B., He, Y., Han, Y., Doyle, P., & Zhu, Q. (2021). *Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports*. *Mathematical Problems in Engineering*, *2021*. https://doi.org/10.1155/2021/6619088

Joergensen Munthe, C. E., Astuti Hasibuan, N., & Hutabarat, H. (2022). *Penerapan Algoritma Text Mining Dan TF-RF Dalam Menentukan Promo Produk Pada Marketplace. RESOLUSI : Rekayasa Teknik Informatika Dan Informasi*, *2*(3), 110–115. https://djournals.com/resolusi

Norindah Sari, S., Reza Faisal, M., Kartini, D., Budiman, I., & Hamonangan Saragih, T. (2023). *Perbandingan Ekstraksi Fitur dengan Pembobotan Supervised dan Unsupervised pada Algoritma Random Forest untuk Pemantauan Laporan Penderita COVID-19 di Twitter*. *Jurnal Komputasi*, *11*(1).

Normawati, D., & Prayogi, S. A. (2021). *Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. Jurnal Sains Komputer & Informatika (J-SAKTI*, *5*(2), 697–711.

Nuri, A. (2022). *Implementasi Naive Bayes dan Support Vector Machine dengan Lexicon Based untuk Analisis Sentimen pada Twitter* [UIN Syarif Hidayatullah]. https://repository.uinjkt.ac.id/

OBERLO. (2023). *Number of Twitter Users by Country*. OBERLO. https://www.oberlo.com/statistics/number-of-twitter-users-by-country#:~:text=According%20to%20a%20recent%20survey,with%2067.9%20million%20Twitter%20users.

Pravina, A. M., Cholissodin, I., & Adikara, P. P. (2019). *Analisis Sentimen Tentang Opini Maskapai Penerbangan pada Dokumen Twitter Menggunakan Algoritme Support Vector Machine (SVM). Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(3), 2789–2797. http://j-ptiik.ub.ac.id

Rahman Isnain, A., Indra Sakti, A., Alita, D., & Satya Marga, N. (2021). *Sentimen Analisis Publik terhadap Kebijakan Lockdown Pemerintah Jakarta menggunakan Algoritma SVM*. *JDMSI*, *2*(1), 31–37. https://t.co/NfhnfMjtXw

Ramadhan, R., Sari, Y. A., & Adikara, P. P. (2021). *Perbandingan Pembobotan Term Frequency-Inverse Document Frequency dan Term Frequency-Relevance Frequency terhadap Fitur N-Gram pada Analisis Sentimen*. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (JPTIIK)*, *5*(11), 5075–5079. http://j-ptiik.ub.ac.id

Raschka, S., Patterson, J., & Nolet, C. (2020). *Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence*. *Information (Switzerland)*, *11*(4). https://doi.org/10.3390/info11040193

Riyanto, S., & Hatmawan, A. A. (2020). *Metode Riset Penelitian Kuantitatif Penelitian di Bidang Manajemen, Teknik, Pendidikan, dan Eksperimen* (1st ed.). DEEPUBLISH.

Rosenberg, H., Syed, S., & Rezaie, S. (2020). *The Twitter pandemic: The critical role of Twitter in the dissemination of medical information and misinformation during the COVID-19 pandemic.* In *Canadian Journal of Emergency Medicine* (Vol. 22, Issue 4, pp. 418–421). Cambridge University Press. https://doi.org/10.1017/cem.2020.361

Rusydiana, A. S., & Marlina, L. (2020). *Analisis Sentimen terkait Sertifikasi Halal. Journal of Economics and Business Aseanomics*, *5*(1), 69–85. http://academicjournal.yarsi.ac.id/jeba

Salsabila, N. A. (2022). *Analisis Sentimen pada Media Sosial Twitter terhadap Tokoh Gus Dur menggunakan Metode Naive Bayes dan Support Vector Machine (SVM)* [UIN Syarif Hidayatullah]. https://repository.uinjkt.ac.id/

Sandy, K. F. (2023, May 6). *ARMY Protes Kenaikan Harga Tiket Konser, dari Rp3 Juta jadi Rp11 Juta*. Idxchannel. https://www.idxchannel.com/ecotainment/army-protes-kenaikan-harga-tiket-konser-dari-rp3-juta-jadi-rp11-juta

Sari, Y., Rizky Baskara, A., Prakoso, P. B., & Royani, N. (2022). *Perbandingan Metode Pembobotan TF-RF dan TF-IDF dikombinasikan dengan Weighted Tree Similarity untuk Sistem Rekomendasi Buku*. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, *9*(6). https://doi.org/10.25126/jtiik.202295709

socket.dev. (2023, November 26). *tweet-harvest*. Socket. https://socket.dev/npm/package/tweet-harvest

Surampudi, S. (2022). *Oracle® Machine Learning for SQL*. Oracle.

Syahrudin, A. N., & Kurniawan, T. (2018). *Input dan Output pada Bahasa Pemrograman Python (Studi Kasus : STMIK Sumedang)*. *Jurnal Dasar Pemrograman Python STMIK*.

Tama, F. R., & Sibaroni, Y. (2023). *Fake News (Hoaxes) Detection on Twitter Social Media Content through Convolutional Neural Network (CNN) Method*. *JINAV: Journal of Information and Visualization*, *4*(1), 70–78. https://doi.org/10.35877/454ri.jinav1525

Tim DetikHot. (2023). *Di Luar Nalar, Calo Jual Lagi Tiket Coldplay Seharga Rp 60 Juta!* . Detik.Com. https://www.detik.com/sumut/berita/d-6725243/di-luar-nalar-calo-jual-lagi-tiket-coldplay-seharga-rp-60-juta

Tim Kreatif CNBC Indonesia. (2022). *Fenomena Konser Post-Pandemi, Haus Hiburan atau Cuma FOMO?* CNBC Indonesia.

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). *Comparing different supervised machine learning algorithms for disease prediction*. *BMC Medical Informatics and Decision Making*, *19*(1). https://doi.org/10.1186/s12911-019-1004-8

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).

Yuniarti, W. D., Faiz, A. N., & Setiawan, B. (2020). *Identifikasi Potensi Keberhasilan Studi Menggunakan Naïve Bayes Classifier*. *Walisongo Journal of Information Technology*, *2*(1), 1. https://doi.org/10.21580/wjit.2020.2.1.5204

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

LAMPIRAN

Lampiran 1 Lembar Persetujuan Seminar Proposal

HALAMAN PERSETUJUAN

Proposal Skripsi ini telah disetujui oleh Pembimbing untuk dilaksanakan.

Disetujui pada

Hari : Rabu

Tanggal : 29 November 2023

Pembimbing I,	Pembimbing II,
Dr. Khothibul Umam, S.T., M.Kom.	Adzhal Arwani Mahfudh, M.Kom.
NIP. 197908272011011000	NIP. 199107032019031000

Mengetahui,

Ketua Jurusan Teknologi Informasi

Nurcahyo Hendro Wibowo S.T, M.Kom.

NIP. 19731222 200604 1 001

ii

Lampiran 2 Lembar Pengesahan Proposal

PENGESAHAN

Naskah proposal berikut ini:

Judul : Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* di Indonesia berbasis Algoritma *Naïve Bayes*

Penulis : **Muhammad Amirul Syachrudin**
NIM : 1908096040
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang komprehensif oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam Teknologi Informasi.

Semarang, 20 Desember 2023

DEWAN PENGUJI

Penguji I,

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

Penguji II,

Dr. Khothibul Umam., S.T., M.Kom.
NIP. 197908272011011007

Penguji III,

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

Penguji IV,

Mokhamad Iklil M., M.Kom.
NIP. 198808072019031010

iii

Lampiran 3 Surat Ijin Penelitian dari Fakultas

KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI
Alamat: Jl.Prof. Dr. Hamka Km. 1 Semarang 50185
E-mail: fst@walisongo.ac.id. Web : Http://fst.walisongo.ac.id

Nomor : B.1639/Un.10.8/K/SP.01.08/05/2024 2 Mei 2024
Lamp : Proposal Skripsi
Hal : Permohonan Izin Riset

Kepada Yth.
Kepala Sekolah SD Islam Al Madina
di tempat

Assalamu'alaikum Wr. Wb.

Diberitahukan dengan hormat dalam rangka penulisan skripsi, bersama ini kami sampaikan bahwa mahasiswa di bawah ini :

Nama : Muhammad Amirul Syachrudin
NIM : 1908096040
Fakultas/Jurusan : Sains dan Teknologi / Teknologi Informasi
Judul Penelitian : Perbandingan Metode Pembobotan TF-IDF dengan TF-RF dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi Covid-19 di Indonesia berbasis Algoritma Naive Bayes
Dosen Pembimbing : 1. Khotibul Umam., S.T., M.Kom
2. Adzhal Arwani Mahfudh, M.Kom

Mahasiswa tersebut membutuhkan data-data dengan tema/judul skripsi yang sedang disusun, oleh karena itu kami mohon mahasiswa tersebut Meminta ijin melaksanakan Riset di Sekolah yang Bapak/ibu pimpin ,yang akan dilaksanakan Jumat, 3 Mei 2024.

Demikian atas perhatian dan kerjasamanya disampaikan terima kasih.

Wassalamu'alaikum Wr. Wb.

a.n. Dekan
Wakil. TU

Muh. Kharis, SH, M.H
NIP. 19691017 199403 1 002

Tembusan Yth.
1. Dekan Fakultas Sains dan Teknologi UIN Walisongo (sebagai laporan)
2. Arsip

Lampiran 4 Surat Kesediaan Validator

SURAT KESEDIAAN VALIDATOR

Saya yang bertanda tangan di bawah ini :

Nama : Nurul Widiawatik, S.Pd

Jabatan : Wali dan Guru Kelas 6B SD Islam Al Madina Semarang

Dengan ini saya menyatakan bersedia menjadi validator Bahasa Indonesia untuk menunjang data penelitian skripsi yang berjudul "Perbandingan Metode Pembobotan *TF-IDF* dengan *TF-RF* dalam Analisis Sentimen Kenaikan Harga Tiket Konser Pasca Pandemi *Covid-19* Di Indonesia berbasis Algoritma *Naïve Bayes*" yang dilakukan oleh :

Nama : Muhammad Amirul Syachrudin

NIM : 1908096040

Prodi : Teknologi Informasi

Demikian surat pernyataan ini dibuat dengan sesungguhnya dan dapat dipergunakan sebagaimana mestinya.

Semarang, 6 Mei 2024

Nurul Widiawatik, S.Pd

Lampiran 5 *dataset* yang Digunakan pada Penelitian

NO	*USERNAME*	*TWEET*	LABEL SENTIMEN
1	achabuccha	WTB Tiket Konser Flowerful JKT48 12th Anniversary - Surabaya Section Rose : 1 Ticket Only Harga Wajar, COD di Venue Unesa Surabaya #JKT48Anniversary #JKTAnniversary12Ticket	Positif
2	whiteveitchii	stress bgt, di LN harga tiket konser meski mahal masih sepersekian UMR sana, di indo udah lebih dr UMR 😭😭	Negatif
3	bloomyskies	akal akalan siapa harga tiket konser melejit naik menyaingi harga properti	Negatif
4	oishiisoda	harga tiket konser naik tapi gajiku kok gak ikutan naik???!!????	Negatif
5	firstloginnnn	wts 2 tiket konser healix musication vol 3 Bali harga 190k aja, yang mau bisa DM yaww. #zonajajan #zonauang #wts #wtb #denpasar https://t.co/x1vkAuhnnR	Positif
6	hanniepeachh	bjir harga tiket konser sekarang kenapa semakin kesana kemari 🤦🤦	Negatif
7	purrgattorio	@helowellington COK MAHAL BGT BABI COMPARED TO HARGA TIKET KONSER KEMARIN ASLI	Negatif
8	mengskuyy	Duh ngeri ya harga tiket konser udah nyentuh angka 5 yuta	Negatif

9	adhierio	Bismillah tawakal WTB // Want to Buy!! Tiket Coldplay Jakarta CAT 4,5,6 Harga Normal naik dikit gpp yg penting wajar. COD FULLPAYMENT VENUE PLISS t. wts wtb konser coldplay #ColdplayJakarta	Positif
10	ulfarikaaa	wkwkwkwk mahal banget bjir, baru kali ini gue liat harga tiket konser sampe 5-7jt 😃	Negatif

Untuk *dataset* lengkap dapat diakses di *https://s.id/dataset_tiket*

Lampiran 6 Daftar Riwayat Hidup

RIWAYAT HIDUP

A. Identitas Diri

Nama Lengkap	: Muhammad Amirul Syachrudin
Tempat Tanggal Lahir	: Semarang, 15 April 2000
Alamat Rumah	: Jl. Menoreh Utara XII/9, RT 02 / RW 01, Kel. Sampangan, Kec. Gajahmungkur, Semarang.
HP	: 089669941150
Email	: *amirulsyachrudin@gmail.com*

B. Riwayat Pendidikan

1. SDI Al Madina Semarang
2. SMP N 13 Semarang
3. SMA N 5 Semarang
4. UIN Walisongo Semarang